

Grocers and Chefs: Software Service Models

I'm talking with more and more with companies considering a shift from traditional licensing models to hosted software-as-a-service (SaaS). It's important to recognize the radical changes such a move may force within your entire company. This month's column serves up a metaphor for the mental and organizational adjustments needed to move from a "product" model to a service business.

First, You're a Grocer

Most serious enterprise software has traditionally been sold under a licensing model: your sales team patiently harasses customers until they agree to take your software. You then send a CD along with an invoice, completing the transaction. The customer is responsible for implementation, integration, network security, and extracting some value out of your expertly crafted bits.

To kick off our metaphor, **licensed software vendors are grocers**. Grocers sell poultry and onions and carrots, but it's not their job to decide if the home-made chicken soup needs more salt.

Likewise, as a licensed software vendor, you've left it up to IT departments at your enterprise customers to choose the right applications, combine them correctly, decide when solutions are ready, and ultimately serve end users with what they need. You deliver an assortment of software bits, get a signature on the purchase order, and tip your hat on the way out. *Off to find another enterprise that needs your product set!*



If, for instance, your company sells resumé-scanning software to corporate HR departments, you expect your customers to buy server hardware, assign user passwords, take calls from confused end users, monitor disk space, and handle complaints from HR managers who don't like your application.

Given the buzz around software-as-a-service, though, your company's new strategic direction is to transform itself into a hosted service: HR managers everywhere will point *their* browsers to *your* server and run your resumé-scanning application over the Internet. You'll charge each a monthly user fee.



Now You're a Chef!

In a restaurant, the chef must cook and deliver complete meals for each guest. He's responsible for the entire dining experience: no excuses about ingredients, faulty ovens, or short-staffed kitchens. Food is available on demand – any time the door is open, customers can come in for hot meal.

Congratulations! By hosting an application centrally, you've put on a chef's apron. Your company now **directly** serves each individual HR manager, and is responsible for their success. You've taken on the entire range of operational and support functions

needed to serve up working applications. Presentation counts: every login is an opportunity to impress users – or fail to deliver.

So What Will You Need?

You'll have at least four new sets of challenges/opportunities in the shift from licensed software to software-as-a-service:

- **Shared Infrastructure.** Also called multi-tenancy, this means that your application must serve many users transparently, giving each a personal experience and protecting one client's data from the next. Converting an existing free-standing application to multi-tenancy can be a huge challenge, so don't attempt this without a SaaS-experienced software architect. There are also infrastructure players with multi-tenant-enabled platforms -- see for instance [AppExchange](#), [Cordys](#) and [Ekartha](#).

In food terms, this is the transition from throwing a dinner party for twenty [*one menu, one serving time, invited guests*] to a ten-table restaurant [*choice of food, staggered arrival times, strangers to please*]. Corporate IT has traditionally served up the "application of the day" while SaaS providers have to satisfy broader tastes.

- **Incremental Sales Cycles.** Enterprise buyers of service-model software tend to start with a few users, and slowly expand their subscriptions. This is one of the big selling points of SaaS: **customers can start small, without making huge financial commitments.** Early subscribers get to sample the service before recommending it to their co-workers... which translates directly into slower revenue ramps. (See my [March](#) byte.)

Licensed software companies, though, are used to big ticket sales with up-front revenue, long before the customer gets any use out of the product. Their marketing strategy and sales commissions don't translate well in the new on-demand world, and need to be overhauled. Think of customers who can buy (and taste!) the appetizers before committing to a full meal. Every online interaction is your chance for good reviews – or unappetizing comments to prospects.

- **Itemized Billing.** Corporate IT may allocate application costs to departments, but it rarely needs to track and account for individual user activity. IT budgets are allocated for bulk user advantage.



Service-based models must (by definition) be able to track each user's activity, then bill and invoice correctly. Many on-demand services fall back on monthly "all you can eat" subscriptions because they can't manage any usage-based accounting. They are missing the chance to fine-tune their value propositions and revenue models, unknowingly leaving money on the table. Think hard about your service pricing strategy and build the audit trail to support it. **"Separate checks, please!"**

- **Real Usage Information.** Enterprise product managers have a devil of a time finding out what features their end users actually use. We field endless surveys, host round tables, and beg to look over customers' shoulders. Results are incomplete, biased, anecdotal.

In this new world, though, service-model product folks suddenly have a new weapon – the activity logs in their hosted systems. SaaS vendors can directly ask and answer interesting questions about user preferences by looking at their customers' online transactions. B2C players like Google and Amazon have always known this, but enterprise folks are still catching up. (75% of service PMs in my [latest survey](#) neglected to do this.) Make sure you're using the new information that's served up for you.

And so on. It's clear that moving to a service model is much more than just hosting a copy of your existing software. **You must be willing to supplement your organization with fresh skills and new kinds of talent.** Not all software companies have the self-awareness to recognize what's missing, or the daring to redesign their organizations as service vendors.

It's the same for your friend whose *crème brûlée* is to die for. Not obvious that she should rent a building, hire a staff, and start a restaurant.

Sound Bytes

Discussions about shifting to a service model are inevitable. Now's the time to consider what new ingredients your company will need to serve up tasty on-demand offerings, and how service-ready your team is. Open up your requirements cookbook, dust off a set of performance measuring spoons, and start an on-demand application shopping list.

"Product Bytes" is an occasional newsletter about product strategies for C-level executives and product/service champions. To subscribe, send an email to subscribe@mironov.com. Back issues are archived at www.mironov.com.

All contents © 2007 by Rich Mironov. Product Bytes (TM) 2002. Contact rich@mironov.com or 650.315.7394.