

High Tech Marketing/Business Model Boot Camp (Part 5): Blue Plate Special, a la Carte, or All You Can Eat

by Nilofer Merchant, CEO, Rubicon Consulting

Do you remember a time when most meals were the sit-down, full-service, dessert-included kind? Even if all you wanted was a cup of soup or a simple salad, you were offered the blue plate special with everything at one price. Then the culinary folks came up with small plates, a la carte items, tastings, pairing menus, buffets, and the like. Whew! Choices—who knew!

So is it any surprise that high-tech companies have stopped serving everything one way with a side of structured licensing? Where once companies had to select /install/customize/upgrade, now we're allowed to use smaller-scale online services that do one thing really well, without integration and without customization.

It's a different era of software delivery models. Proof: A few weeks ago IBM started charging for software based on "processor value units." A recent article by Stephen Shankland of CNET announced the change on 7/25/06, calling it "one of vocabulary," but it's becoming clear that the utility pricing model I've written about for the last two years is about to become commonplace. In the same article, Jeff Tieszen, an IBM spokesperson, said that Big Blue is moving toward a pay-per-use (read: utility pricing) model.

What does this mean? Anything? Nothing? Probably a little of both.

To take a look at it, let's look at the new name of this delivery model: SaaS (software as a service) in depth.

SaaS: What Is It?

I know a lot about SaaS. So do you, but you may not know it by that name. It is typically...

- Software that's viral (anyone can download it and try it out)
- Generates value for an end user (so they'll have an incentive to install it)
- Doesn't require any data entry or training (so users can work with it instantly)
- Generates immediate value ("value first, pay later")

Anything strike you when you see that list? Do you think of salesforce.com, Skype, IM, QuickBooks Online, Six Apart's Movable Type, Google Desktop? These are all SaaS solutions. They range from simple to sophisticated, but they all have in common scalable offerings, hosted within firewalls, and they're taking over traditional software release cycles.

SaaS is a delivery model rather than a specific market or type of company. It could be described a software-on-demand model similar to the time-sharing systems that many of us grew up using. The hosting and licensing fees are not distinguished.

Why Does It Matter?

SaaS uses a utility pricing model rather than term licensing. Utility pricing offers benefits to both the creators and consumers/customers, as customers pay for usage rather than access. This means that new customers are not required to make a huge financial commitment up front while offering lots of upside to publishers from heavy users. If you are a consumer, you no longer need to buy a \$299 package of Microsoft Office, you can (likely soon) use Google Apps instead.

In the same way as full-service comprehensive meals went by the wayside in America, technology companies can't hope to retain customers just based on a huge, upfront financial commitment. With SaaS and the related business models, that gigantic check so beloved by your CFO just disappeared.

Key Changes in Sales Model Driven by SaaS

The selling and marketing model changes profoundly if you're trying to get an online service deployed virally in companies via end users, as opposed to selling down through IT managers. Most of today's enterprise software companies don't have a clue how to sell this way, and it makes them extremely uncomfortable. Selling to the end-user requires a different voice, approach, sales model, licensing vehicles, pricing structures, and service follow-through than selling to IT buyers.

It's easy for companies to underestimate the depth of the changes required to make SaaS work. In addition to a very different sales and marketing model, SaaS requires a completely different engineering structure and processes. Even the existing code base must be rewritten to support delivering it as a service.

Because these changes are so wrenching, it's very tempting for companies to make more superficial alterations and convince themselves that what they've done is good enough for now. They'll add a web interface to their product and offer a hosted price option (carefully tailored not to cannibalize the current product line), and then they'll wonder why the new initiative fails.

Substance Trumps Superficial

A colleague on the Rubicon team points out that the SaaS situation is reminiscent of the Melanesian cargo cults that proliferated after World War II on some Pacific islands. After the war ended and the soldiers went away, islanders tried to make the cargo come back by imitating the superficial features of the soldiers—they painted uniforms on themselves, marched in groups, and built things like air strips.

Companies trying to implement SaaS by copying its superficial features are basically technology cargo cults. Unless they make the fundamental changes that SaaS requires, they're going to get

about as rich as those island natives, marching around in the hot sun and wondering when the airplanes will arrive.

What's Not to Like?

The case against SaaS cites a couple of major barriers to its broad deployment. One is data security. Large, traditional firms are very concerned about securing their data against both loss and theft. The idea of allowing that data to be hosted someplace and accessed via the Internet is daunting, if not downright terrifying.

The second barrier is scale. Many software services are being designed as quick-turn projects, without a lot of thought to long-term maintenance and how they'll scale to handle large numbers of customers and more complex feature sets. This problem has sometimes affected even Google; its blogging service, Blogger, has been plagued by periodic outages and data loss. Consumers can live with that (sort of), but it would be utterly unacceptable to a large firm.

The fact that a service interruption at Six Apart or Salesforce.com hits the Wall Street Journal headlines says something about the impact of those "interruptions."

The SaaS skeptics conclude that it may get some traction at the edges of business but it will never be ready for the most demanding, mission-critical corporate tasks, especially in the enterprise.

An Inevitable Change

Certainly, in the next 12 months, the challenges in the SaaS world will prevent it from taking over the entire software industry. But major transitions never happen that fast anyway. We think it's a good time to go back and re-read Clayton Christensen. In *The Innovator's Dilemma* he did a great job of describing the gradual process by which steel mini-mills bit off pieces of the steel industry, starting with the lowest-value products that traditional steel companies didn't care about deeply.

We see the same sort of process happening in SaaS. The easy targets and the smaller customers are being nipped off first. But, over time, SaaS implementations are becoming more and more sophisticated. We also see SaaS companies starting to create versions of their services that are designed to scale, and companies can host those versions within their firewalls, reducing security concerns.

The best services will replicate a process similar to what Research in Motion did with Blackberry email. The first Blackberry systems came with a small desktop application that took care of passing messages to the mobile device. Over time, as the number of Blackberry users in a firm grew, RIM sold servers to the IT department that took over from the desktop component.

That process—get the users hooked, then sell to IT after you have established an installed base—lends itself perfectly to small guerrilla organizations like the SaaS companies. They can grow under the radar for a long time. By the time they're competing for the biggest corporate accounts,

the rest of the market will have already switched—and it'll be too late for the incumbent software companies to do anything about it.

Be sure that's not you.

August, 2006