

Software Gets A Service Call

New Rules For The Software Business

By Kyle C. Murphy

November 27, 2007

There is something abuzz in the software industry and this time it's not some new hotshot startup or marketing spin from the big boys—it's coming from customers and developers alike and it's called Software as a Service, SaaS for short. An IDC report projects that 10 percent of the market for enterprise software will migrate to pure SaaS by 2009. This is much more than a technology-driven revolution; it is a completely new game that will require software companies to reinvent themselves. The concept behind SaaS is not that new. During the dot.com boom ASPs (application service providers) were heralded as the next coming, but fell well short of expectations. Like many technology shifts in early stages, both the technology and the customers were not quite ready. This time around customers are driving the revolution and Web 2.0 technology is making it possible to bring enterprise level applications to the web. Hang on to your seat, it's going to be one heck of a ride.

The SaaS revolution is indeed under way, with vendors and customers actively talking up the new model, but there's a problem and it has nothing to do with customer adoption or technology issues. The problem is the software industry itself, and in particular, the executives and management of software companies. A quick examination of the current SaaS market reveals a cross-section of software industry veterans starting and running the new SaaS divisions and firms. These management teams take the "Software" part of SaaS very seriously, but seem to relegate the "Service" part to a secondary status, as just a function of being a web based application. This philosophical gap will ensure plenty of SaaS firms and initiatives fail, regardless of how cutting-edge the software itself may be. SaaS is a completely new business model, not just a new way of delivering software.

For the SaaS business model to really work and truly deliver on its revolutionary promise, SaaS companies will have to stop operating like software companies and start acting like service companies. That means everyone from strategy and marketing to operations and finance will have to learn new rules. Software companies are going to have to treat customers as partners and learn how to provide the right level of customer service without surcharge. This means no more charging for premium support—there are other ways to manage support load than charging for it. Imagine your mechanic or your doctor charging one rate to do their job, but if you really want it done right (e.g. premium support) you have to

pay more...how absurd! Ask yourself as a customer—when was the last time a software company treated you the way you wanted be treated and you felt you got your dollars' worth? Times are changing for the better of the customer and for the software companies if they learn the new rules. Otherwise, they may go the way of other technology companies that didn't change with customer demand...remember DEC, Atari, Osborne, and Data General?

Suggested below is a new set of rules to provide a much-needed makeover for software companies. These rules are culled from 15-plus years experience in service-based businesses and extensive management experience with software companies. These rules are written with business software providers in mind, but the rules translate very well to consumer software companies as well. For long-time software veterans these might sound like heresy, but just take a deep breath and forget everything you know.

Software as a Service—Contrary to what many software veterans think, SaaS is not just about hosted software. Putting a piece of software in a datacenter and changing your pricing model does not make you a SaaS company. First, you're not a software company anymore; you're a service company. This requires a monumental change in thought process and operations. It also comes with a high level of responsibility to your customers—long-term. Think of it as a marriage that requires work from both parties. The days of dumping a piece of software, working or not, at a customer's location, taking your money and running are gone. Charging more to fix your software down the road is definitely long gone. Your solution needs to work every day from the beginning and evolve as your customers' business does. Both parties will feel pain if everything isn't working and bitterness will develop if both parties don't give equally. This is bad because either your customer is unhappy and will not renew their contract or you're unhappy and don't renew the contract—either way you lose revenue. Remember, SaaS is all about recurring revenue, so long-term satisfaction equals long-term revenue. Secondly, your solution needs to provide more value than outsourced hosted software. This means adding value by proactively integrating with complementary service providers, providing non-software outsourced services (in-house or strategic partner), plus a whole lot more, depending on your industry. Just repeat after me, "We're a service company."

Forget the whales, aim for minnows—You have to love startup software company business plans that indicate they only need two large customers to hit their numbers. The probability of success with that strategy is very low, thus the large amount of software company failures. The real market power for SaaS companies is with small and medium sized businesses. The

SaaS model makes it feasible for companies to get off the ground with smaller customers for many reasons, including simplified support, lower implementation costs and reliably recurring revenue. The number of available customers as well as total revenue is much greater in the SMB space. Additionally, selling to SMBs is easier, with shorter sales cycles and more direct access to decision makers. One other fringe benefit is you won't have to compete with the 800lb gorilla firms as much; they give a lot of lip service trying to serve the SMB market, but ultimately they can't price right for the market and don't have the right structure to service them. While everybody competes for the whales' attention, aim for the minnows—it's almost like shooting fish in a barrel.

Queer eye for the software guy—It's time to get pretty. The days of clunky looking user interfaces designed by engineers and programmers are gone. Both business and consumer users have come to expect very sophisticated but easy to learn and easy to use interfaces. A great looking and easy to use interface not only will win you new customers and provide for shorter sales cycles (ever try to sell a product that's hard to use or ugly?) but will keep support calls down. While form still must follow function and flash without substance will still fail, function without form will also fail. Winning SaaS companies will deliver great user experiences, attractively packaged, easy to use, and robustly engineered. So get a makeover before you go out on the town, it will improve your odds.

Get touchy-feely—Get your mind out of the gutter, this is about demos. From the \$4.95 per month consumer app to enterprise solutions, potential customers want and should be able to play around with a fully functional demo of your system. If it's too difficult to use without extensive training...see the KISS rule at the end of the list. If you're afraid of your competitors seeing your super-duper whiz-bang feature that gives you a competitive advantage, sorry, you're too late. If they haven't seen your product already, they will somehow sometime soon, so plan on it. The days of proprietary technology providing a defendable competitive advantage is fading fast. It's now about providing the best value-added service to maintain your competitive advantage. So come on, take the blackberry off your belt and let loose.

Port this—If one thing has investors and VCs more nervous than anything in the SaaS space it's the "porting" of older client-server systems. And rightly so, since many of the recent SaaS failures have occurred before the company ever got to the starting line, due to failed engineering execution. If your existing technology is over five years old, start over. Yes, I know you have millions of dollars already invested in your product, but the value is not in the

code itself. The value is in your knowledge of the customers' business processes and the way you improve them and in the power of your brand. New, more powerful programming tools and the need to build a new, pure, web-based user interface provide a unique opportunity to optimize everything about your system. It is less expensive than ever to get programming done almost anywhere on the globe, with Eastern Europe and Asia as primary options. Go for the upgrade on your system and programming platform—it won't hurt as much as you think, plus fixing the old system is ultimately a lost cause.

One fish, two fish, red fish, blue fish—Dr. Seuss wrote this book in 1960 and it is still a great read, but pricing by one seat, two seats is so very 1990. For instance, when my company signed up for Salesforce.com (the granddaddy of SaaS companies) last year, we found the pricing almost appalling. Not only did it feel too expensive for us as a startup, but also it was priced by the number of seats, disguised as logins, on a monthly basis. To add insult to injury, they require you to pay each year's fees up front (or quarterly, but at a higher rate). NetSuite (the new SaaS golden boy) charges the same way. Sounds a lot like traditional software to me. I don't know about you, but I know a CEO logging in a couple times a month to look at sales figures is a whole lot different than a sales manager using it daily. The consequence for these software companies is customers gaming the system, such as using one login for all non-sales related managers. Ultimately this means less revenue and more detrimentally, being less integrated and more dispensable in their customers operations. A much more logical and customer friendly way is to charge based on some metric that reflects how much a customer uses the system and how they perceive value. Examples include numbers of transactions, value derived per transaction, or measures of user activity. Of course, a minimum fee level is acceptable to maintain business viability. It's completely logical to assume that revenue and retention would increase and turnover would be reduced if SaaS companies priced based on usage and value received. It's time for software companies to grow up and get a seat at the big persons table by learning to treat customers like adults through responsible pricing.

Proprietary my arse—If you have a proprietary piece of hardware or software language, throw it away. Nothing scares a customer more than thought of your company going belly up and not being able to quickly recover data or operate because of a piece of proprietary technology. Before you get your suspenders all wound up, think of all the once-successful companies who had thrived off of proprietary technology only to crumble—firms such as Wang, NeXT, and many more. If you think your piece of proprietary software provides you a competitive advantage, think again—it no longer applies in today's age of cheap offshore

programming, standardized coding platforms and open source. Everything can be reverse engineered; the only way to maintain a competitive advantage is through continued innovation and by providing superior customer service (See a theme developing here?). Even former proprietary stalwarts Cray and Sun have become supporters of standards and open technology. While you're at it, make sure it is easy for other SaaS firms to integrate with your system and preemptively build a bi-directional application interface. The SaaS world will work better and be more profitable if we all get along and make it easy to work with each other. So get off your proprietary high horse and join the game before it passes you by.

Empire building—Microsoft never built computer hardware to run DOS or Windows, so why build your own datacenter? Yes, the thought of having a brand new datacenter running on solar energy next to Google and Yahoo! in Oregon gets your ego pumped, but leave the datacenter building and operations to the datacenter center people. The most popular non-search website in the world—Myspace—runs on around 150 servers in a joint data center facility. In our company, the application runs in a datacenter and network that has over \$500 million invested in it and more people handling network operations and security/intrusion than we have in our entire company. Leveraging the data center's expertise and track record provides a solid marketing message as well for start-ups. Imagine the response you would get from your investors if you said we needed to raise an additional \$100 million or so to build a datacenter. Not only will you save loads of money, but lots of headaches as well, by leaving the datacenter to the datacenter people. Put down your hammer, coil up the Cat 5 wire and focus on your core competency—software as a service.

Innovation over litigation—It still seems amazing to hear about a software company suing another for some minor IP infringement. Not to say that you shouldn't protect your IP, but pick your battles. Spending thousands, if not millions, of dollars on lawyers and wasting enormous management time is not going put your company in a better position in five years. That time and money would have much been much better spent on things that your customers care about—like better customer service and new functionality. What about infringing on someone else's IP? Well there's always more than one way to skin a cat. For instance, when Research in Motion (maker of the ubiquitous Blackberry) was sued by a patent squatter, they let the lawyers go to war. When the judge placed a date for an injunction, it took less than 30 days for RIM to say they had a technical work around, though in the end they decided to settle. Either way, it still cost them many millions of dollars in legal fees and, more detrimentally, it made many of their customers very nervous. It's a safe bet that they could have spent a few million dollars as soon as the suit was filed to re-engineer

some of their technology/processes and avoided most if not all of the legal costs. Given the current state of technology, if your business strategy and model rely exclusively on one or more patents, you are probably in deep trouble. Like service companies, your brand reputation will be more valuable than any technology you have, so protect that. Save the planet—there are enough lawyers in the world without encouraging them with more lawsuits.

Moral support—Technical support shouldn't cost extra; in the service industry, service is service, good or bad, and SaaS companies are service companies. Price your product correctly in the first place to allow for the "right" level and type of support required to maintain high customer satisfaction with your product. While you're at it, put all your training online and make it free to use. It is well documented that web-based training is highly effective and the ongoing cost is negligible. If you charge for training, customers will use the least amount of it possible, thus leading to more support calls. A well developed and easy to use web-based training program along with a well integrated online help system is crucial to controlling costs in a SaaS business model. Also, leverage your customers' knowledge for support and product planning by letting them speak to each other in online user groups. Yes, they might say bad things about you, but better to hear and respond to complaints now than ignore the fact customers are unhappy and lose them later on. Learn from the support load and user comments. If support levels are high or there are a lot of negative user comments, then either your product isn't working right, you need to improve the usability of it, or improvement is needed in your training/online help program. Blaming customers for your product's shortcomings will not fly in the SaaS world—it's like blaming your CPA for taxes. This may be the most contentious of these rules, but it's time the software industry gets a conscience and treats customers as more than a paycheck.

Go lean—Between the addiction of receiving large amounts of venture capital and poor management, software companies are notoriously overstaffed. Contrarily, service companies don't hire for a position until they can see a 90% or more long-term utilization rate for that employee. Overstaffing is a sign of poor management. No matter how much funding you receive, restrain the desire to hire a person for every task. Having too much staff can lead to poor employee morale and subsequently poor customer service because employees will lack purpose if not tasked adequately and correctly. Whether it be outsourcing, off shoring, freelancing, or strategic partnering, there are other ways to get jobs done. Running lean requires advanced planning skills complimented by superior people management talent to leverage the most out of your staff, but it's worth it. The reward for growing lean is better valuations (less dilution); less time spent raising capital, and fewer growing pains. Put down

the venture capital donut and run your company lean, like you were spending your own money.

And finally...

Keep It Simple, Stupid (KISS principle)—Ok, so this isn't new, but executives and software businesses seem to keep forgetting it. Stop making everything so complex - it doesn't impress your customers or improve your bottom line. Customers don't like overly complex pricing, contracts, training, interfaces, or relationships...just as you probably don't like them. Senior level executives and other final decision makers are generally not technical gurus, so to close deals and get the greatest user acceptance keep it simple, stupid. If you choose to ignore all of the other rules, keep this one in mind, otherwise you'll find your customer base kissing your business goodbye in the new software world.

Customers are demanding it, the technology is ready, and now is the time for the SaaS revolution, but it will be a huge failure if software companies don't learn to act like service companies. For the SaaS revolution to live up to its promises, software companies and management will need to learn a new set of rules to live by. These new rules will be hard for existing software companies to learn so I expect most of the successful SaaS companies to be new companies or spin offs. Salesforce.com and NetSuite were both start-ups that came out of Oracle, though they still suffer from software company hangovers. For those willing to learn and live by the new rules, margins far beyond traditional software await them. Hang on; this ride has just started and it's already at breakneck speed!

About the author: Kyle Murphy is currently the EVP/CFO of VantageLLM, an early stage SaaS company located in Pasadena, California that serves the finance industry. He is also working on his Doctor of Business Administration degree at The University of Manchester in England. Prior to VantageLLM, Kyle spent time as a management consultant, in investment banking, and as a CEO of an ad agency among other adventures. He received his MBA from Pepperdine University. More information about Kyle is available at www.kylemurphy.com.