How to Differentiate SaaS vs. Cloud: "Just Because an Application is Hosted in the Cloud Does Not Necessarily Make it SaaS"

With cloud computing all the rage these days, it's no surprise that many software vendors are looking to capitalize on the momentum by releasing "cloud versions" of their conventional software products. Delivering software as a service (SaaS) from the cloud has distinct advantages both economically and technologically. But does hosting an application in the cloud make it SaaS? Not necessarily! Add this bit of confusion to a market already filled with hype and jargon, and it gets even more difficult to decipher exactly what you're buying. Don't despair. As a consumer, here are five tests you can use to help determine if the application you're evaluating is SaaS or a hosted version of a conventional software product.

Test #1: Multi-Tenancy

This is the big one. In a true SaaS model, there is only one copy of the application (single instance) which all customers use (multi-tenant) and can customize to meet their unique requirements. Contrasted with a hosting model in which, yes, the application is "in the cloud" but every customer has their own copy of the application (multi-instance, single tenant). This is effectively the same as the ASP hosting model of the late 90s – even if your vendor is using "virtualization" technology, it's not multi-tenant and therefore not SaaS. The basic difference is that instead of maintaining only one version of a SaaS application for all customers, vendors who host need to maintain a version of their application for *every* customer. This is one of the major reasons why SaaS can be delivered so much more economically than conventional software products. So dig in and ask questions. For example, ask how many versions of the application the vendor supports (answer should be one). Ask if you can choose not to upgrade (answer should be NO; although, you should be able to turn off certain features you don't wish to deploy). Or, simply ask how many code bases the vendor is maintaining (again, answer should be one).

Test #2: Self-service

Self-service capabilities are a hallmark of SaaS applications. Since there is truly no software or appliances to be installed and configured (either physically or virtually), all interaction with the application can happen in real-time directly from a web browser. Generally speaking, SaaS companies tend to operate like internet-based consumer companies. Visit the vendor's Website to see how much of your "shopping" experience you can complete online. For example, does the vendor offer a free trial? Can you make a purchase with involving the vendor's sales personnel? Provisioning trial and production accounts on demand is a relatively trivial matter for a SaaS vendor and is easily automated. On the other hand, attempting to offer trials for conventional software products can become another scaling and maintenance challenge for the vendor – even if that software is "virtualized." Vendor costs will escalate and that cost will ultimately be reflected in its pricing. If at any step you get a message to "fill out this

form and a salesperson will be in touch," there's a good chance you're not dealing with a SaaS offering.

Test #3: Community

Another trademark of SaaS applications is the ability to leverage its user community to deliver enhanced product capabilities such that the application gets "smarter" and more valuable as the user community grows. Since companies using conventional software products are essentially each on their own island, sharing any kind of assets or collective wisdom is extremely difficult. Contrasted with two companies using Salesforce, for example, where they can natively share sales pipeline data if they so choose (without compromising the overall security of their data). SaaS applications have the ability to harvest the collective intelligence of its entire community of users and bring that insight to bear for the benefit of each user – think Amazon.com's "people who bought this book also enjoyed…" Ask the vendor in what ways it leverages its user community to deliver enhanced functionality (beyond product steering groups and the annual user group meeting).

Test #4: Maintenance and Enhancements

This is where SaaS really shines and where it gets hard to hide if you're deploying conventional software products. Because if you are deploying conventional software products, every customer has their own instance – whether that's a physical on-premises instance or a virtualized instance in the cloud. As such, releasing even a simple bug fix can become a major effort, not to mention deploying maintenance releases or product enhancements. These maintenance and scaling challenges result in greater cost, more complexity and less time for innovation – all of which impact end customers. In a true SaaS model, a fix, upgrade or enhancement is made once and all customers enjoy the benefits immediately. SaaS vendors can then focus their resources on building and releasing great new product capabilities versus maintaining multiple versions of their product and managing releases, etc. It's what the analysts at Saugatuck Technology Inc. recently called the "endless cycle of innovation." Ask your vendor how many major new enhancements or features have been released in the past year and also ask what percentage of the vendor's customers are on the current release. With conventional software products, you'll find customers in "rev lock" situations where they've customized their instance of the application to the point where they can't keep up with the latest releases. With SaaS vendors, you'll generally find a very steady stream of enhancements and features and because all customers are on the same code base, all will have access to the latest and greatest product capabilities.

Test #5: Talk to Customers

Finally, talk to customers – and not the references the vendor gives you. You can usually find a list of customers on the vendor's Website but you might find even more useful information in public user forums, blog posts or on Twitter. No one will know the

answers to the above questions better than the customers who have implemented and been using the vendor's application for any appreciable amount of time.

Summary

I've heard it argued, what does all this matter to the end user? Why do they care how the application is being delivered? True, the typical end user probably does not care. However, end users generally aren't making the purchasing decisions and paying the bills or responsible for the information technology strategy for the entire enterprise. Those folks tend to care a lot more.

There is a reason why SaaS eclipsed ASPs for application delivery. It is hands down a superior delivery model from cost to technology to scalability. Multi-tenancy is what makes all the benefits of SaaS possible: rapid time to deployment/value, faster innovation cycles, infinite scalability, etc. SaaS has a dramatically reduced cost structure because it leverages one platform. Vendors who deliver their application via SaaS have a competitive advantage over those who do not.

If you're in the market for SaaS applications, I encourage you to ask questions when making your selection. Go beyond "we're in the cloud" to understand exactly how your vendor delivers its application. Presumably, you've come to the cloud because you want to exploit the economics and advantages of utility computing and applications. Remember, just because an application is hosted in the cloud does NOT necessarily make it SaaS. Caveat Emptor – if it's not multi-tenant, it's not SaaS.

About the author:

Bob Moul is president and CEO of Boomi, the market and technology leader in on-demand integration. Previously, Bob was president of the global education software business at SCT (now SunGard Higher Education) and group president responsible for several software businesses at Maximus. Bob began his career with EDS initially as a systems engineer and advanced into senior management positions including director of EDS' operations in Hong Kong and China, and executive director of its federal government business in Australia. His 28-year career has spanned all aspects of IT services, software and consulting in a variety of executive leadership, technical management and engineering roles.

Bob graduated from the University of Maryland, University College, with a Master of Science degree in technology management. He also completed executive programs at the University of Michigan Business School and the Center for Organizational Learning at MIT. He serves on the advisory boards of several start-up companies and is a frequent speaker and panelist at various SaaS and cloud computing conferences.