

Beware Of The “SaaS” Trap

SaaS In Name Only Won't Deliver The Same Benefits And May Cost You Much More

by [Liz Herbert](#) and [James Staten](#)
with [Chris Andrews](#) and Fraser Tibbetts



WHY READ THIS REPORT

There's a disturbing trend reemerging in our client inquiries of late: rising demand for software-as-a-service (SaaS) solutions and purchases of solutions that are not really SaaS. When we inform clients that the solution they're talking about isn't SaaS, we are increasingly hearing, "So what? As long as I don't have to run it, it's better." This is a trap that can cost you more, result in less agility, and lock you into a poor sourcing decision. Too many vendors today are seeing an opportunity to make big profits in software-as-a-service by taking advantage of customer naiveté and simply rebranding older deployment models as SaaS, on-demand, or cloud. These solutions inevitably fail to live up to customer expectations for agility, standardization, and efficiency. As SaaS continues to increase in popularity across a growing range of application areas, technology buyers (ranging from CIOs to line-of-business [LOB] execs) need to know what differentiates SaaS from application service providing (ASP) and other on-demand implementations cloudwashed with the SaaS term. This report helps you understand the true SaaS characteristics so you can make the right choices for your business.

SAAS OR NOT SAAS? HOW DO YOU KNOW? AND WHY DO YOU CARE?

Your aging human capital management system is on its last legs. You've customized it to the point that it is barely recognizable compared to the base application you started with, and the business processes you've built into it still reflect procedures you abandoned three years ago. You've got to move to a modern solution that's more flexible and better aligned to your current hiring and staffing practices. And you're a manufacturer with pretty standard HCM needs. So why not go SaaS?

One of your strategic vendors has an HCM SaaS solution that looks pretty appealing. It runs in the cloud and is accessible on mobile devices. And you get to decide when you take new feature upgrades, as well as define the SLAs, the security, and even the deployment configuration. But to get this level of control you have to commit to the solution for a five-year term at a minimum annual spend that seems a bit high. Huh? Is this really SaaS? Probably not. But, the deal might still look pretty good — in the short term. Unfortunately, the long-term picture is not likely to look as good if you are expecting benefits like fast, seamless upgrades or the ability to quickly expand into new modules or to flex your usage.

Clients who are misled into buying SaaS solutions that are not “real” SaaS will find that: a) the solution fails to deliver the business agility they hoped they would get (both initial time to “go live” as well as ongoing business agility); b) the solution is not truly consumption-based and often requires heavy upfront payment; c) the solution cannot deliver the cost advantages that was hoped for (or, if it does, the vendor cannot sustain those breaks for long and will fail or cut corners elsewhere); and d) they aren't really able to get off of the “major” disruptive upgrade cycle typical of classic apps.



Welcome To The SaaS-Washed World

With enterprises increasingly interested in SaaS and all things cloud, vendors aren't about to miss the party, and are smart enough to know that they have to at least market themselves as SaaS to get into your selection process. Although SaaS has begun to overtake traditional models in a few application categories (such as SFA and HCM), the vast majority of enterprise applications categories are still on the very early cusp of SaaS entrants. Often, the vendors hope that you won't know the difference or that once your full requirements come through, you'll realize it wasn't SaaS you really wanted after all. And, in some markets without real SaaS options, none of the vendors will call each other out on their misleading campaigns, which can propagate the confusion. Don't fall for the trap. There are real differences between true SaaS services and other hosted models such as application service providing (ASP) and application outsourcing.

SaaS Is More Than Just Cloud

SaaS is a particular approach to software delivery that yields high efficiencies for the vendor through economies of scale, greater business agility through continuous development methods and application componentization, and standardized service delivery. You know it's truly SaaS if it can be consumed almost immediately and its terms are take it or leave it. Forrester defines SaaS as:

A standardized software capability delivered via Internet standard technologies in a pay-per-use, self-service way.

When business users or CIOs look to SaaS for the promise of speed, flexibility, or lower upfront costs, they believe they are getting:

- **A multitenant approach that yields better cost efficiency.** SaaS services are usually designed as multitenant applications with highly standardized and automated capabilities and processes that are designed to achieve broad market appeal. The service's core functions are implemented in a uniform fashion for all customers and offered to clients as is — take it or leave it. Your customization options are typically very limited, and you cannot alter the service's underlying configuration. This lets the SaaS provider spread the costs of a single implementation across a volume of customers yielding lower operational costs that are usually passed on to you through lower prices. Solutions that lack multitenancy — such as ASPs or other hosting models — could mean higher costs and almost always mean minimums for the buyer (i.e., minimum of 100 seats); in a worst-case scenario, solutions that can't get economies of scale could lead to providers not being able to sustain business profitably and, ultimately, the risk of provider failure.
- **One implementation that yields greater agility.** Because all customers are running on the current version of the application, the SaaS vendor can deliver new features and capabilities as fast as it can productize them. Vendors employing continuous development practices can add new capabilities as rapidly as every two weeks, meaning enterprises gain much greater

value much faster.¹ And, it is easier for the vendor to support additional quick-to-deploy functionality through partner add-ons (such as salesforce.com’s AppExchange) because all partners and customers only have one version to worry about. Solutions that are multitenant at the infrastructure layer only (built on cloud) or other variations of SaaS mean that the buyer will be stuck in the usual rut of having access to major upgrades on a yearly basis (or every couple of years). They will also find it harder to take advantage of a true app store, in that vendors will have a harder time creating ready-to-go, downloadable solutions across multiple versions.

- **Subscription and pay-per-use pricing that means greater cost flexibility.** SaaS is usually priced on a subscription basis, frequently per user per month (or year), with a (generally) predictable cost that stays fairly constant over time. Some SaaS vendors truly let you “pay as you go” while others require a minimum contract and are experimenting with business models to differentiate themselves and better fit certain customers’ buying criteria. But nearly all SaaS vendors track consumption and factor this into their pricing. These pricing schemes make it easy for buyers to start small and discover/prove value before investing more heavily (which is a sharp contrast to the large upfront license fees required in on-premises deals or even some on-demand models).
- **Self-service and automatic provisioning that speeds implementation.** SaaS buyers demand more than a web page where you can order the service. They want fast, automatic provisioning. This means that the service can be provisioned with no human intervention, sales calls, service tickets, or lengthy procurement processes in less than 15 minutes. This *implies* that procurement of the service is fully automated — and to deliver cloud services cost effectively, this is essential. SaaS providers that cannot offer this need days or even weeks to get a solution deployed. And, they often incur extra costs in doing this provisioning, which ultimately affect the customers.
- **Increased autonomy for the business user.** Our research has shown that the top benefit clients receive from SaaS solutions is business agility — the ability to innovate more rapidly and to make changes to their technology quickly and iteratively in line with fast-changing business strategy. Why is SaaS better at this? With modern, born-in-the-cloud SaaS solutions, vendors put significant effort into selling and marketing to the business user and to the line-of-business heads. One significant advantage to SaaS vendors targeting business buyers is that the SaaS vendors have made tools that are more user-friendly, easier to configure by business, and that free up the business from needing IT to do (relatively) simple tasks. Business can modify forms and pages, can change the look-and-feel of the application, create their own reports, and add-on third party tools without technical know-how. These properties are especially important in the age of the customer.²

Distinguishing SaaS From Other Off-Premise Models

Okay, so if that's SaaS, what are all these other posers carrying the SaaS banner? The key non-SaaS models are below (see Figure 1). There are financial, operational, and agile implications with each SaaS alternative:

- **ASP and other on-demand flavors are close but no cigar.** Born in the late 1990s, this application hosting model takes a single-tenant application (that was not built for multi-tenant, cloud computing but usually for a client-server model) and deploys multiple versions on a common infrastructure (which themselves could be multitenant). (Today's versions of this deployment option will often use cloud for the common infrastructure.) Each customer gets its own instance of the application, which can be more palatable to the security and risk team, and the end customer gets a browser and/or mobile interface. The vendor achieves greater efficiencies through a common infrastructure and operational model. However, upgrades to ASP offerings follow a similar schedule to traditional licensed applications: typically, one upgrade every 12 to 18 months with patches and bug fixes released in between. And you, the customer, can determine when these upgrades are applied (because only your instance is being affected).

This model lacks the agility and cost efficiencies of SaaS but can be a fast way for a vendor or third-party service provider to offer up a well-known application as SaaS. These offerings often have a per-user subscription model but with a high minimum commit that covers the capital and operational costs incurred with each new instance deployment.

- **Application outsourcing is just what you think it is.** Application outsourcing is when a service provider hosts and manages an application for you, from their cloud. Some ISVs offer this service directly, but firms can also hire third-party service providers like Accenture or Infosys for application outsourcing. In this model, the service provider has best practices it brings to deploying and managing the application, but you have final say over the configuration, SLAs, and when to upgrade to a newer version. And the more you specify the more unique the implementation — and the higher the cost and longer the contract.

Figure 1 Software-As-A-Service Versus Other Off-Premise Deployment Models

		SaaS	ASP	Managed
Tenancy	All customers run same code version?	Yes	No	No
	Significant modifications possible?*	No	Yes	Yes
	Multitenant architecture?	Yes	No	No
	Originally designed to be SaaS?	Yes	No	No
Upgrades	Who controls upgrade timing?	Provider	Customer	Customer
	Frequency of feature enhancements	As often as every week	Typically 2-4 per year	Typically 2-4 per year
	Frequency of major releases	Typically 2-3 per year	Typically 1 or fewer per year	Typically 1 or fewer per year
Payment model	How is the software priced?	Subscription	Subscription with multiyear commit required	Upfront license + maintenance fee
	Customer owns license?	No	No	Yes
Location	Where does the code and data reside?	SaaS provider	ASP	AO provider (but sometimes at the customer site)
Responsibility for managing the application	Who develops application?	SaaS provider	Software provider (may not be the ASP)	Software provider (not hosting provider)
	Customer control over operation, deployment and configuration of the app	None	Limited	Negotiable
	Vendor examples:	salesforce.com, Workday, SAP's SuccessFactors	BTRG's PeopleSoft-As-A-Service offering	PeopleSoft run by Oracle On Demand, Accenture managing SAP

*Significant modifications include: adding unique modules, changing application configurations, adding custom code, unique infrastructure, etc.

WHAT IT MEANS

WHY THIS MATTERS: EXPECTATIONS AND OUTCOMES

Does the technical implementation and deployment model really matter if all three deliver the application to my users via the browser? Absolutely. And these differences may help you realize if SaaS is the best solution or if you really would benefit more from ASP or outsourcing. Where it matters most is in your business requirements. If you:

- **Value agility and customer experience, go with SaaS.** Of the three deployment models above, true SaaS yields the fastest delivery of new capabilities and features. If ensuring the best customer experience, adapting to changing employee needs, and rapid change are key business goals then seek out SaaS solutions. Historically, these offerings have been first to support mobile, tablets, HTML5, SAML, REST, and other key customer enablement technologies. A word of warning: In many applications categories, these solutions are still sparse, either available only from new, niche vendors or available only for a very small subset of functionality.
 - **Have privacy and compliance worries, go with ASP.** Consistently in our Forrsights surveys, enterprises list security as a key concern with SaaS and cloud solutions. This concern may be technical, experiential, or psychological in nature, but might not be arguable in your organization — yet. If this is the case, your organization will be more comfortable with a solution that architecturally isolates your application instance and your data from other customers like ASP. You will trade off some of the agility and customer experience benefits of SaaS but achieve peace of mind which might better match your company’s comfort level. What’s new? While the word ASP may have gone out of fashion, this model has morphed into a new breed of offerings built on infrastructure-as-a-service (IaaS) technology, such as Amazon.com or the Computer Science Corp (CSC) cloud. ASP solutions that leverage underlying IaaS will get you closer to your goals of agility and flexibility — as opposed solutions built on more traditional hosting and shared services platforms.
 - **Need a custom implementation, go with application outsourcing.** As with most companies, you need this particular application but you need to heavily adapt it to your unique business processes, need custom modules, and unique integrations. SaaS isn’t well suited to this use case and ASPs won’t be thrilled with an implementation that doesn’t leverage their common operational best practices. Application outsourcing gives you the flexibilities you are seeking. But be prepared to pay for uniqueness. As with the ASP model, look for modern AO solutions that include cloud technologies at the infrastructure tier, which can help you balance your need for custom with a desire for economies of scale, flexibility, and efficiency.
-

ENDNOTES

- ¹ For more information on continuous delivery, please see the November 6, 2013, “[Use Forrester’s Continuous Delivery Assessment Model To Increase Delivery Speed](#)” report.
- ² Empowered customers are disrupting every industry — and CIOs need to understand how technology management must adapt in this rapidly evolving world. For more information on the age of the customer, please see the October 10, 2013, “[Technology Management In The Age Of The Customer](#)” report.

