Resource Guide: Making Sense of SaaS Pricing and Contract Terms

RIMINI STREET FUNCTIONAL SUPPORT SERVICES
AUGUST 2018

Rimini Street®
Engineered for Support™

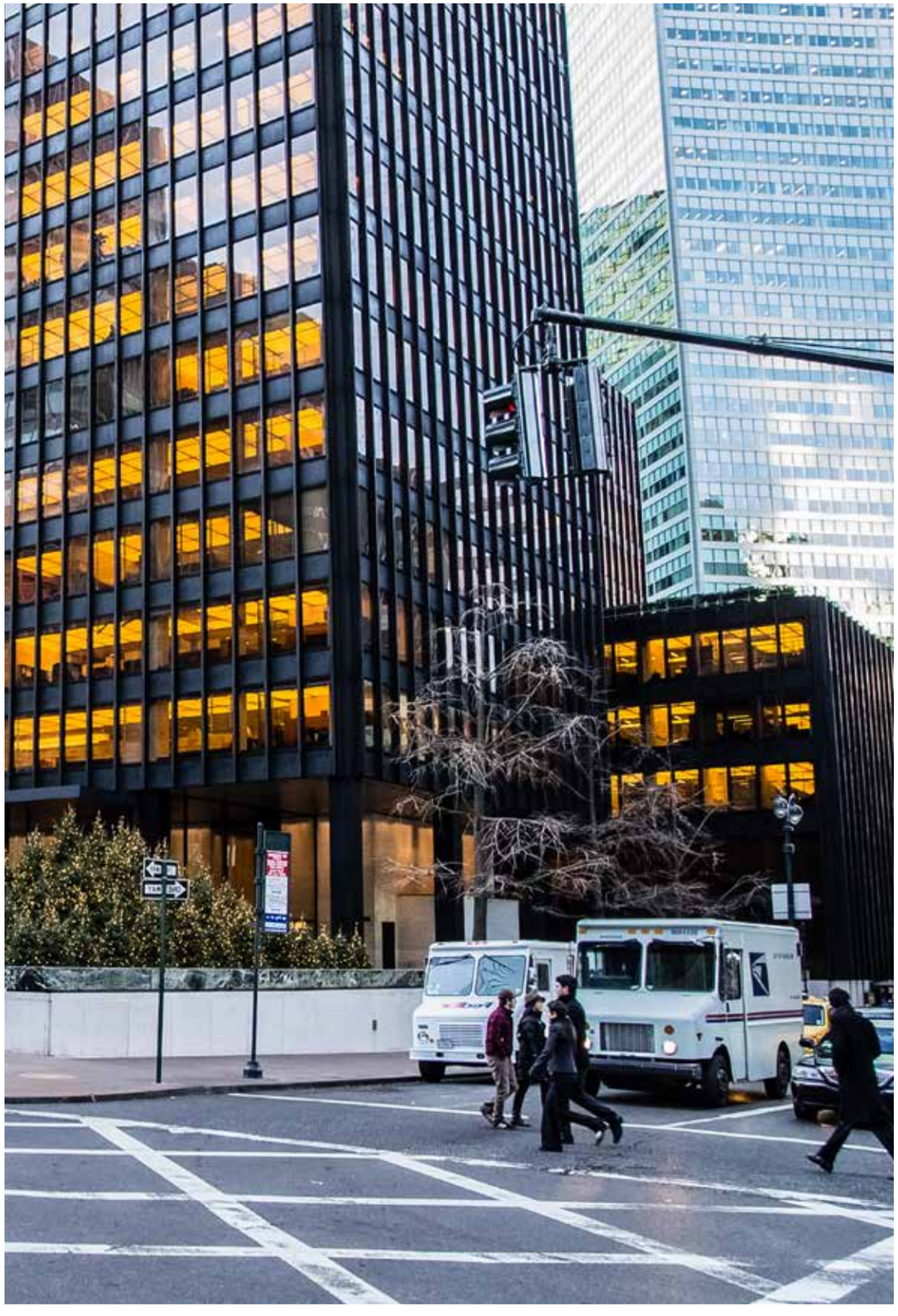
Contents

Executive Summary

Software vendors frequently use the term SaaS, so much so, that today, stating that an application is SaaS is now a main marketing tool used to increase the appeal of an application. As a general statement it can be said that there is far more pretense to SaaS than reality.

Knowing what classifies as true SaaS is important for corporate IT buyers because SaaS is often associated with lower total cost of ownership. Yet, even if an application is truly SaaS, than this assumption may not be true.

This resource can be viewed as a primer for corporate buyers. It can be used to make sense of the SaaS messaging that they receive from software vendors, generally, and SAP, in particular.



SaaS Benefits

There are many benefits of SaaS. According to Nucleus Research, SaaS applications can be developed and deployed five times faster and at half the cost of traditional applications. (*Source: November 2007 Report H90, "Business Critical: Software As A Service Vendor Strategies"*) As a result, SaaS applications also have a lower total cost of ownership (TCO) than either internally deployed or hosted-non-SaaS applications. The benefits of SaaS have garnered attention, but no one can control what software vendors call SaaS. It is a self-claimed title. As a consequence, SaaS has considerably morphed, and in many cases is no longer a traditional SaaS design.

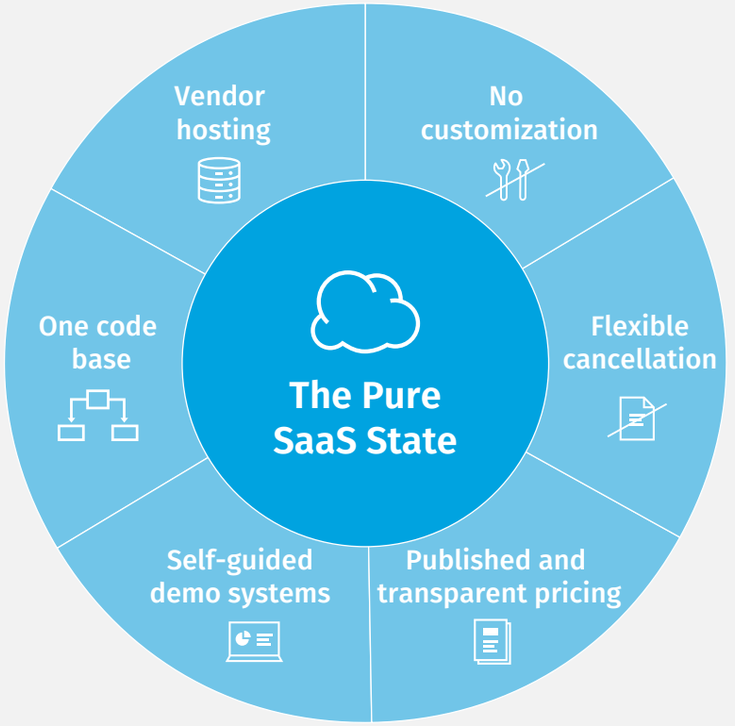
The Pure State of SaaS

SaaS is predicted to rise from $12 billion in revenues today to $50 billion by 2024. (*Source: Mike Wheatley, July 15, 2016, "The State of SaaS: Spending to hit $50bn by 2024"*) To provide some context, the four largest software vendors in the world have combined revenues of roughly $140 billion. The top 23 vendors, each with revenues of more than $2 billion, have combined total revenues of roughly $206 billion in 2014 dollars for software products and maintenance. (Sources: *List of the largest software companies*; *The World's Biggest Public Companies (last viewed in March 14, 2018)*; *Microsoft's revenue from 2012 to 2017 financial years, by segment (in billion U.S. dollars)*)

What we will refer to as the pure state of SaaS is a SaaS model that some software vendors follow. It is almost entirely followed by those software vendors who began their lives as SaaS vendors, and have not operated any other way.

The pure SaaS state has the following characteristics:

- One code base
- Vendor provides the hosting (i.e., the vendor provides and maintains all infrastructure for the application)
- No customization
- Flexible cancellation
- Published and transparent pricing
- Uses self-guided demo systems



5X

SaaS applications can be developed and **deployed five times faster** than the cost of traditional applications. *(Source)*

$

SaaS applications can be developed and deployed at **half the cost** of traditional applications. *(Source)*

$50B

SaaS is predicted to rise from $12 billion in revenues today to **$50 billion by 2024**. *(Source)*

One Code Base — Multi and Single Tenancy

SaaS applications should have just one code base for the software used by all customers. This is referred to as a multitenant architecture. Multitenancy is also a commonly used term here; however, it is not a universally defined term.

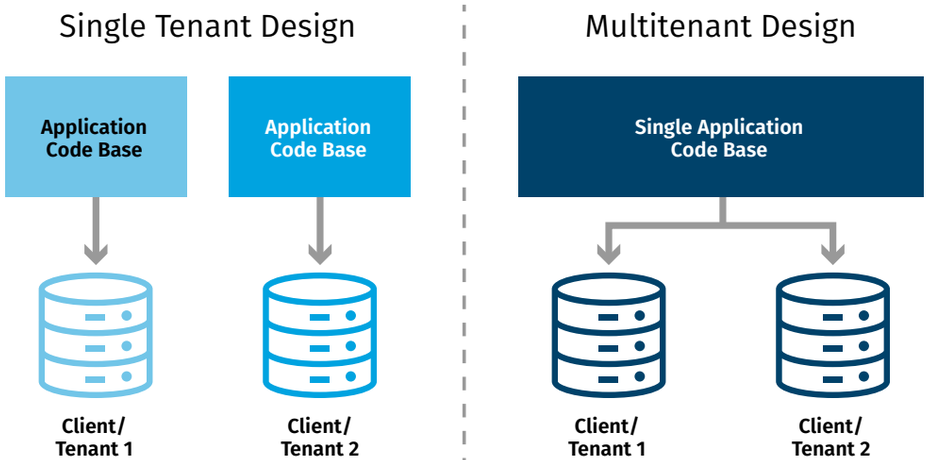


Figure 1. Multitenancy is a software design where a single instance of the software runs on a server that has multiple tenants. The tenants are customers. So Company 1, Company 2 and Company 3 are all using a SaaS application. They use the same code base, but they have different data, configurations and user management. Company 1 only sees Company 1 data; Company 2 only sees Company 2 data, etc.

Multitenancy is accomplished by setting up a separate database schema per tenant. SaaS vendors who are experienced in this are able to onboard new customers quickly and at low expense for an existing product already purchased.

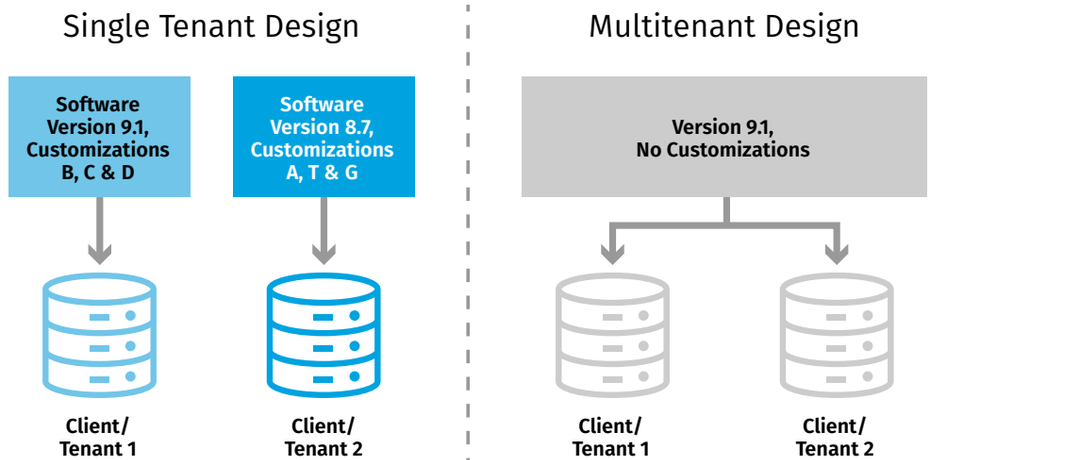


Figure 2. The disadvantage of multitenancy is that each customer must be on the same version. However, for most SaaS customers, an upgrade is transparent. Also, multitenancy does not allow for customizing the application. Doing both of these things (upgrades and customizations) means different code bases for the application. This means a reduction in efficiency and an increase in costs.

Why Multitenancy Is So Important

Multitenancy is an important component of SaaS. While the single-tenant design outsources the hosting of the application to a third party, multitenancy provides economies of scale to the management of users, because only the data is separate. The applications that each user logs into are actually the same applications. The ability to share data elements across tenants may be the most important characteristic of SaaS. Multitenancy functionality is part of Oracle 12c, Oracle's new database. In Oracle 12c, the database has functionality designed to help SaaS vendors scale better than with earlier releases. This allows Oracle not only to sell 12c to SaaS vendors but also allows SaaS vendors to manage their own hosting, because Oracle is an IaaS provider, as well as a database provider.

The second benefit in multitenancy scenarios is that more of the resources are shared among customers. If 10 companies are sized in 10 single-tenant instances, the resulting size (and cost) will be higher than 10 tenants using a single, multitenant application. It turns out that server capacity is under-utilized on average, with estimates running from 10 to 25 percent of server capacity being utilized. (*Source: Tom Gillis, September 02, 2015: "Cost Wars: Data Center vs. Public Cloud"*) SaaS vendors are simply able to share computer resources in a way that other companies with internally deployed applications cannot.

The Reality of One Code Base

Many vendors who market their applications as SaaS are not truly multitenant, because they are not operating from one code base. There could be many different versions of the application running on each hosted instance. Good examples of this are SAP and Microsoft. Accordingly, the one code base SaaS model is not as prevalent as so-called SaaS providers would have us believe.

Vendor Provides the Hosting for its Applications

Pure SaaS vendors provide their own hosting, and this allows them to upgrade their application as needed. When another company performs the hosting, this may contribute to a staggered upgrade. When hosting is moved to a third party, it can indicate there may no longer be a multitenant environment. Although there can be a multitenant environment within the third party (as they can have multiple customers), a single code base is not being utilized at that point. Also, hosting companies provide other products beyond ERP (e.g., bolt-on applications).

With a third-party host, there are additional inherent complexities with access to the system on the part of the software vendor. For years, the vendors most dedicated to SaaS, vendors like Arena Solutions and Salesforce, hosted their own applications. As the IaaS market has matured, that model is increasingly being called into question. Even Salesforce, the largest SaaS vendor, recently began outsourcing its hosting to Amazon Web Services. (*Source: Barb Darrow, May 25, 2016, "Salesforce Inks Major Deal With Amazon Web Services"*) Even for SAP, who tries to heavily market that it hosts its own applications, many companies are now offering third party hosting for SAP software. Overall, it seems that SAP is actually not all that interested in providing hosting, which raises questions about how a vendor like SAP deals with third parties hosting their software.

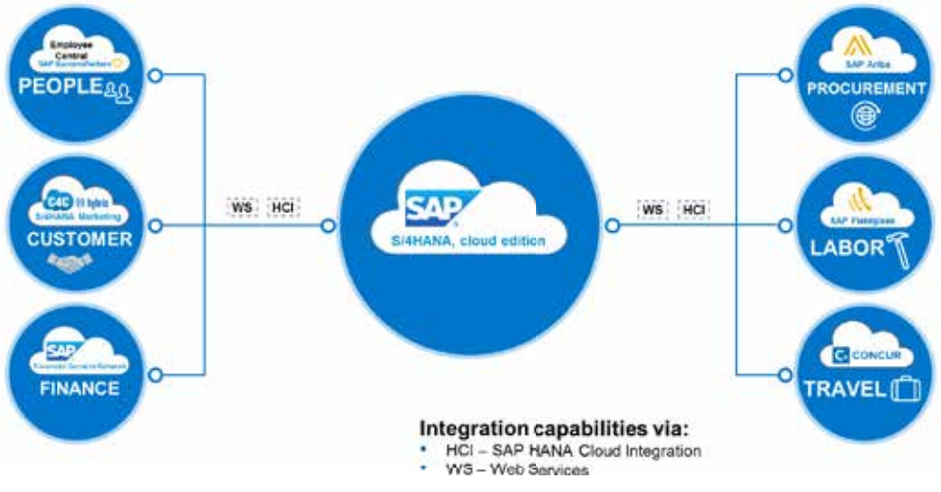


Figure 3. SAP's marketing materials (including the graphic above) show many cloud-based third-party applications. However, SAP does little of its own hosting except for its newly acquired applications that were already cloud-based before they were purchased. Therefore, most of SAP's database and applications revenue is from applications not in the cloud. (*Source: Javit Gellaw March 01, 2016, Integration Capabilities of SAP S/4HANA to SAP Cloud Solutions*)

The Reality of the Vendor Providing the Hosting

Neither SAP nor Microsoft actually host many of their applications. For Microsoft, it recently moved Office365 to the cloud, however, Microsoft Dynamics is hosted through a third party. Even Microsoft Dynamics was never actually designed to be a multitenant application.

Similarly, SAP frequently talks about hosting applications itself, but the vast majority of its hosted applications are only the result of acquisitions that were originally designed for the cloud. The reality is SAP sells few of its internally developed applications as SaaS to be hosted by SAP.

No Customization

SaaS customization was first reviewed above in the section on multitenancy (i.e., One Code Base — Multi and Single Tenancy). SaaS's underlying data model and system architecture is not customizable. "No Customization" allows the SaaS software vendor to spend less time dealing with various platform patching, compatibility issues and individual upgrades. In fact, a true SaaS vendor is supposed to upgrade a single instance, which then updates all customers simultaneously. This should then imply less support expense.

The Reality of No Customization

SaaS applications tend to predominantly be simple applications that are not customized for each client. The fertile areas of SaaS tend to be applications such as bill of materials management, CRM, HRMS, travel and expense, online meetings and email. However, the current trend indicates that software vendors are increasingly moving more complicated applications, which have a long history of being customized, to the cloud.

- The need for customizations will not end just because applications move to the cloud. SAP's ERP system, called ECC, "shows 90% of ERP systems range from minor to extreme customization," and those with significant or extremely customized systems represent 27% of instances. (*Source: Eric Kimberling, June 2013, The Long-Term Effects of Heavy ERP System Customization, last viewed on Feb 17, 2018*) SAP's replacement for ECC, called S/4HANA, will be made available in the cloud.
- Questions around customization and how these customizations will be ported to a SaaS environment have not been fully answered. In fact, such questions are already starting to emerge.

Flexible Cancellation

The SaaS customer has a month-to-month contract with the software vendor. This prevents vendor lock-in, and as a result, SaaS vendors have a stronger (more frequently reviewed) incentive to retain customers.

The Reality of Flexible Cancellation

When SaaS was conceived, one of the inherent characteristics was providing customers with high flexibility. Pure SaaS vendors, such as Marketo, ConvertKit and others, operate this way, and all offer month-to-month pricing, which gives customers the ability to cancel on short notice.

- The ability to offer this short-term notice is a testament to the confidence that the vendor has in the value that its application delivers. However, even true SaaS vendors entice customers with discounts for signing longer contracts. Wall Street notes how well SaaS companies retain their customers.
- Customer retention is high because it is well known in the industry that SaaS customer relationships are not immediately profitable for SaaS vendors. (Evidently, this is why a slew of SaaS software vendors are unprofitable. It is also why SaaS vendors are not expected to show a profit with a customer for 12 to 24 months.) In many cases, it often takes at least one to two years before the customer relationship produces fruit for the vendor. Most of the companies that did not begin as SaaS vendors do not offer this flexibility. It is quite rare for a hosted solution to be priced as a one-time charge. More commonly, most of the vendors that have historically sold software for internal deployment offer a yearly subscription. The flexibility is considerably lower. For example, when Oracle, SAP or Microsoft (and many others) say they are offering hosted applications, they are not observing the historical rules of SaaS regarding the ability of the customer to get out of the obligation. And that is only considering the vendor side (and not also considering the third-party vendor side). While the statements and documentation by cloud vendors are all geared to how much freedom the customer has, there is much more lock-in than is generally acknowledged. The following from AWS's pricing guide is a good example of this type of language:

"AWS offers a range of Cloud computing services. For each service, you pay for exactly the amount of resources you use. There are no minimum commitments or long-term contracts required. This pricing model helps replace your upfront capital expense with low variable cost. For compute resources, you pay on an hourly basis from the time you launch a resource until the time you terminate it. For storage and data transfer, you pay on a per gigabyte basis. At the end of each month, you are charged only for what you've used that month. The number and types of services offered by Amazon Web Services (AWS) has increased dramatically, but the AWS philosophy on pricing has not changed: At the end of each month, you pay only for what you use, and you can start or stop using a service at any time. No long-term contracts are required." (*Source: Amazon Web Services, Inc., March 2016, "How AWS Pricing Works"*)

- This pricing commitment sounds great, of course. However, what is conspicuously left out of the analysis is that by using cloud software, as is being used today, a natural dependency is created.
- While all the cloud product vendors may speak in terms that make it seem as if they are offering customers the ultimate in flexibility, the objective of the subscription model is to make the service license complicated so the customer never leaves. Investors in cloud vendors are all looking for guaranteed continual revenue streams.

Published and Transparent Pricing

SaaS vendors often provide published and easy-to-follow pricing. Customers should be able to price the software without ever talking to a sales rep. This is important, because published pricing cuts down on the communication time and effort required to determine and negotiate a final agreement. This is demonstrated by vendors such as Amazon and Salesforce (see image below) who offer fully published pricing with virtually no negotiation. For AWS, Amazon chose to outsource much of the presales effort to its customers. It did this by doing the unthinkable a few years ago: publishing its pricing and making it virtually non-negotiable.

In doing so, Amazon removed an enormous obstacle for customers to adopt its services. Think about it: by the time a potential customer approaches Amazon for AWS, he/she has already determined if the economics make sense.

"Enterprise IT sales historically involved building relationships with senior executives at prospective customers. Once a relationship is formed over several rounds of lunch, dinner, and select sporting events, it's possible to entice the CIO to consider a new cloud provider. AWS responded to this time-honored relationship ritual by combining the ease of swiping one's credit card, transparent pricing, and 1 year of free services for everyone." (*Source: Blake Huber, August 17, 2015, "4 Reasons AWS Dominates the Cloud Industry".*)

The Reality of Published and Transparent Pricing

As with cancellation terms, software vendors whose products have traditionally been internally deployed by the customer do not offer published pricing. There are important reasons for this:

- This software vendor model has maximum flexibility in pricing through this lack of publication.
- It makes sales volume "the gatekeeper" for pricing information.
- It prevents prospects from performing comparison pricing without investing heavily in the sales process. Prospects are often told that pricing is not possible without "learning more about your requirements." There is certainly validity to this quotation, but it can also be used to defer providing the information as long as possible.

As a clarification, many vendors may say that their price lists can be found, but they are so complex that they are in practical terms opaque. SaaS vendors on the other hand traditionally have provided a high degree of pricing transparency and even have configurators on their websites.

Per-user pricing is more straightforward than IaaS or PaaS pricing, and is not as simple as it sounds. AWS's *'Simple Monthly Calculator'* is a great example of this. AWS has created a series of web pages for its offering that allow pricing to be calculated by the customer. Although this reduces the workload on the part of the vendor's salesforce, it's more complicated for the purchaser.

Using Self-Guided Demo Systems

Generally, traditional software application sales teams tightly control exposure to the application. The prospect is only allowed to see a demo of the system for short periods of time. Specialized resources called presales consultants walk prospects through a demo. This approach does not provide or allow a thorough evaluation of the usability of the system.

- Demonstration consultants, who are very familiar with the application, are able to do many things that normal users often cannot.
- Alternatively, when SaaS vendors provide access to a SaaS demo environment, the experience becomes self-guided. This gives you more control and allows them to understand whether the application is a good fit for them in a shorter period of time.

As an example, *with Arena Solutions, a provider of product lifecycle and bill of materials software*, access to a self-guided demo of their systems is normally quite fast.

The Reality of Using Self-Guided Demo Systems

True SaaS vendors want to get prospects to use their demo system as quickly as possible because it is their primary tool for turning a prospect into a customer. Faux-SaaS vendors use access to a demo system (either online or in the standard demo presentation) to move the prospect to the standard sales process used for traditional software. If a prospect cannot access a demo system on his or her own, then the control is given back to the software vendor. In almost all those cases, the traditional software sales process for internal deployment is being followed.

SAP is a prime example of a vendor that historically never provided demo environments. It has recently reversed course though and now provides some, but they are quite limited. We still would not generally categorize SAP as a cloud vendor though, except for its acquired applications. This shows how much SaaS is influencing the traditional software vendors.

Interestingly, SuccessFactors was a well-regarded SaaS vendor even before it was purchased by SAP. However, *SuccessFactors' website has no pricing published and no public trial available*. We checked back in 2010, prior to SAP's acquisition and also found no pricing published and no public trial available. While SuccessFactors followed some of the rules of being a SaaS vendor, it is another example of a vendor that did not follow all of the SaaS rules.

Parting Thought: Don’t Forget about Data Migration

Moving data and transitioning to a new SaaS vendor’s application is almost always a significant effort. The concept of the theoretical flexibility offered by SaaS is appealing, but the mechanics of moving from one SaaS vendor to another SaaS vendor, particularly in a long-term use of SaaS vendor product, are much less than ideal.

Conclusion

Understanding the details of SaaS will reduce a lot of risk in your IT roadmap. Particularly important, is understanding whether the SaaS vendor you are dealing with is actually offering you a pure or true SaaS solution that conforms to all or most of the SaaS requirements. Or, if the terms SaaS is simply being used as sales terminology.

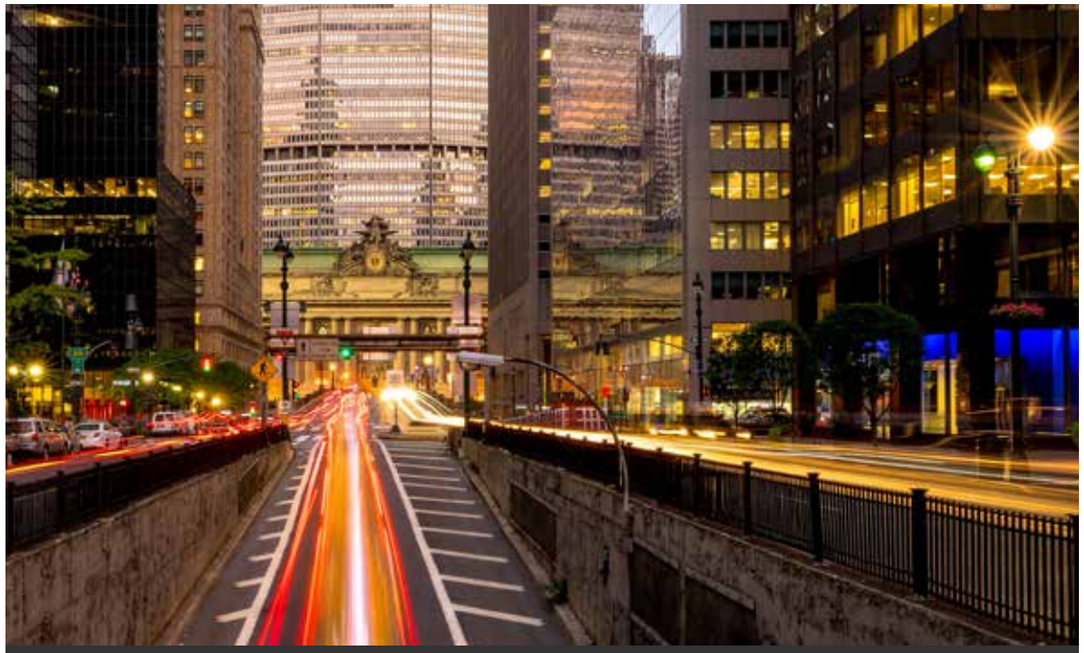
Remember, SaaS is not merely provided as a hosted solution, and it is easy to get trapped if you aren't careful.

When it comes to SAP, they follow very few of the criteria that makes software SaaS. SAP argued against SaaS for roughly a decade before finally accepting the idea, and this was external forces rather than some internal vision within SAP. Our view is that if SAP could keep the stock valuation of being known as a 100% SaaS vendor, while in actual fact being a 100% internally deployed / hosted applications vendor, this would be their preference.

Many other traditional software vendors would likely feel the same way. This is the heart of cloud-washing, the desire to be accepted as a SaaS vendor without meeting the criteria of actually being one.

True SaaS is differentiated from faux SaaS by the answers to questions around multitenancy, configurability, customizability, the handling of upgrades, published service level agreements, scalability, the ability to test a fully functional demo system, pricing transparency and severability.

These are the key areas to explore, so you don't get trapped.



How Rimini Street Can Help

Want to understand the role of SaaS in your IT roadmap and avoid unnecessary traps? In addition to cutting your enterprise software support costs in half (just for starters) and supporting your custom code, we also offer cloud management, security and roadmap services that will help optimize your current IT operations and maximize the value of your ERP and database investments

Learn more about how we can be part of your innovation success story at https://www.riministreet.com/about-us.

Disclaimer

This document was created by Rimini Street, Inc. (“Rimini Street”) and is not sponsored by, endorsed by, or affiliated with Oracle Corporation, SAP or any other party. Except as noted herein or required by law, the contents of this document and any documents associated with it are subject to the intellectual property rights of Rimini Street. This document and the documents associated with it may also contain proprietary or confidential information intended for disclosure only to the intended recipient, and accordingly the document’s contents and any documents associated with it shall not be reproduced, posted, adapted or redistributed directly or indirectly, in whole or in part, in any medium without the express written consent of Rimini Street. The information in this document is provided “as is.” Except as otherwise expressly provided in writing by Rimini Street, Rimini Street assumes no liability whatsoever and disclaims any express, implied or statutory warranty relating to the information presented or to Rimini Street’s services, products, or inventions, including, without limitation, the implied warranty of merchantability, fitness for a particular purpose, or non-infringement. In no event shall Rimini Street be liable for any direct, indirect, consequential, punitive, special, or incidental damages (including, without limitation, damages for loss of profits, business interruption, or loss of information) arising out of the use or inability to use the information. Rimini Street makes no representations or warranties with respect to the accuracy or completeness of the contents of the information presented, and reserves the right to make changes to the information, services, products, or inventions contained herein at any time, without notice.

Worldwide Headquarters
3993 Howard Hughes Parkway, Suite 500. Las Vegas, NV 89169, USA Phone: 702.839.9671 Toll-free in the USA: 1.888.870.9692
riministreet.com | linkedin.com/company/rimini-street | twitter.com/riministreet

Rimini Street®

Engineered for Support™