International Journal of Advance Research and Innovation
Vol. 5(2), Apr-Jun 2017, pp. 57-64
Doi: 10.51976/ijari.521711
www.gla.ac.in/journals/ijari

Article Info
Received: 25 Jan 2017 | Revised Submission: 20 Feb 2017 | Accepted: 28 Feb 2017 | Available Online: 15 Jun 2017

Software as a Service (SAAS), Future of Information Technology & Business

*Ravindra Kumar**

ABSTRACT

Software as a service, or 'SaaS', is a software application delivery model by which an enterprise vendor develops a web-based software application, and then hosts and operates that application over the Internet for use by its customers. Customers do not need to buy software licenses or additional infrastructure equipment, and typically only pay monthly fees for using the software. SaaS applications provide customers with centralized, network-based access to data with much less overhead as compared to using a locally-installed application.

Keywords: *Cloud Computing; Risk Factors; SAAS Infrastructure.*

1.0 Introduction

Software is ubiquitous in today's business world, where software applications can help us track shipments across multiple countries, manage large inventories, train employees, and even help us form good working relationships with customers. For decades, companies have run software on their own internal infrastructures or computer networks. In recent years, traditional software license purchases have begun to seem antiquated, as many vendors and customers have migrated to software as a service business model.

Cloud applications or Software as a Service (SaaS) applications deliver software as a service over the Internet, eliminating the need to install and run the application on the customer's own computers and simplifying maintenance and support, and equipped with decomposable applications, managed services, shared hardware /software /admin resources and Web-based services.

2.0 Cloud

Cloud is used to explain everything today. Numerous software providers are offering cloud-based systems. Cloud-based software required internet to access that system. we need an internet connection and the ability to log into the system via a web browser.

Before cloud-computing, companies would need their own servers or computers at their office in order to use software. We would have to physically install a CD on our personal computer to use it. After that we could only access our data on that computer. For cloud-computing, however, a 3rd party is hosting our software on a remote server where they store and process our data. These servers are housed in data centers all over the world. When we want to access our data, we open browser and log into our account. we can be on a desktop computer, phone, or PC. we can be at the office, home, or at the airport.

3.0 Cloud Benefits

There are many benefits to cloud-computing. We don't have to host, maintain, or upgrade the servers. We have access to our data in real-time, whenever we need it. Costs are usually lower because we just pay to rent space on the hosted servers. we usually pay for a subscription to have access to our data.

A few examples of cloud computing are

- Amazon Web Services (AWS) – Amazon's global compute, storage,
- database, analytics, application, and deployment services.
- Google Cloud Platform – Google's core infrastructure, data analytics, and machine learning.

**Department of Computer Science Engineering, Delhi Technical Campus (IP University), Greater Noida (U.P.), India (E-mail: ravindrakumarchauhan@gmail.com)*

- Microsoft Azure – Open, flexible, enterprise-grade cloud computing platform.

With cloud-computing, we are responsible for maintaining whatever applications we run on the 3rd party's or Amazon's servers. The hosting company maintains the physical servers and operating system.

4.0 SaaS Model Incoporates a Number of Unique Charactersitics

- Although the customer loses some level of control, the SaaS model shifts the burden of getting and keeping an enterprise application up and running from the customer to the vendor. It permits users to leverage the software functionality without the burden of deploying and managing the software themselves. The applications are accessed through a Web-based interface typically run from a Web browser. This delivery mechanism allows for easy scalability as new users are added.
- Generally, rather than licensing, installing and maintaining software on clients' computers or servers, the SaaS model lets users access the vendors software via the Internet on a "pay-as we- go" basis.
- Each customer can opt either to share access to the software with other customers, thus enabling shared total costs and creating economies of scale, or decide to be a single tenant, thus providing greater control and security.
- The SaaS model includes systematic support of the software, rather than annual maintenance and upload of fixes and patches, to all subscribers.
- The SaaS model enables every customer to benefit from the vendor's latest technological features without the disruptions and costs associated with software updates and upgrades.
- The SaaS model eliminates the added costs and complexities of deploying additional hardware and software, or dedicating additional staff resources to support an enterprise application on an ongoing basis.
- Although the customer loses some level of control, the SaaS model shifts the burden of getting and keeping an enterprise application up and running from the customer to the vendor. It permits users to leverage the software functionality without the burden of deploying and managing the software themselves. The applications are accessed through a Web-based interface typically run from a Web browser. This delivery mechanism allows for easy scalability as new users are added.
- Generally, rather than licensing, installing and maintaining software on clients' computers or servers, the SaaS model lets users access the vendors software via the Internet on a "pay-as we- go" basis.
- Each customer can opt either to share access to the software with other customers, thus enabling shared total costs and creating economies of scale, or decide to be a single tenant, thus providing greater control and security.
- The SaaS model includes systematic support of the software, rather than annual maintenance and upload of fixes and patches, to all subscribers.
- The SaaS model enables every customer to benefit from the vendor's latest technological features without the disruptions and costs associated with software updates and upgrades.
- The SaaS model eliminates the added costs and complexities of deploying additional hardware and software, or dedicating additional staff resources to support an enterprise application on an ongoing basis.

5.0 Key Benefits and Risk of Saas Solutions

5.1 Short-term benefits

- Accelerated software deployment with generally less risk. Quicker time to deployment (value).
- Lower up-front costs. Higher productivity/ROI, at a lower Total Cost of Ownership (TCO).
- No additional hardware and lower internal staffing requirements for system implementation, maintenance and support.
- May provide greater reliability, security and privacy but tradeoffs must be considered.
- Greater agility to scale software to meet changing business requirements.
- Immediate, easy wins for proof of concept projects across the Enterprise.
- Successful proof of concept projects will validate and inform the longer term SaaS strategy.
- Opportunity to "lean" the state procurement process to better advantage the SaaS benefit of a short implementation cycle.

- Ability to re-allocate IT resources from legacy systems to more strategic projects.

5.2 Long-term benefits

- Resource efficiency gains.
- Quality of Service improvements, (citizen service improvements).
- Reduced risk as compared to conventional project/system development methods.
- Lower total cost of ownership.
- Business process alignment across the Enterprise.

5.4 Risk factors

- Difficult to achieve agency agreement on common functions, solutions and procurement process.
- Threat of employee job loss--cultural change-impact to workforce.
- Transition from legacy system to multi-tenant, shared system or other hybrid SaaS model.
- Unable to realize savings that might result from SaaS deployment due to need for cost recovery of State Data Center (SDC) investment.
- Potential lack of alignment/interoperability with existing technology/data sharing environment.
- Potential complexity/risks associated with data management, security and privacy. Potential loss of data or data breach -- Public Records, Privacy.
- Time consuming legacy procurement process that may not fit acquisition of SaaS offerings.
- Inability to determine TCO of the SaaS Solution.
- The potential impact associated with the need to change vendors (disentanglement).
- Vendor support and reliability over time.
- Executive Approval--executives burned at the stake for taking risks.
- Funding and budget process – moving from a capital investment to an operational expense.
- Potential legislative opposition to hosting SaaS solutions and state data in out of state locations.
- General loss of control of provisioning and operational processes – from personnel practices through data handling practices to policies to operational procedures.

6.0 Kind of SaaS

6.1 Business applications

The Business Applications industry represents software that helps various businesses accomplish their tasks quickly and accurately. Magic Formula stocks in this industry include global leaders such as Microsoft (MSFT), Accenture (ACN), and Electronic Data Systems (EDS), as well as smaller niche players like STARLIMS Technologies (LIMS) and American Software (AMSWA). This is a pretty broad range of providers. Microsoft, of course, makes the ubiquitous Windows operating system and Office productivity software that is used by the vast majority of businesses today. Compare this broad based offering to STARLIMS, which focuses on a very specific niche - laboratory data management.

Business Applications as a sector is difficult to merit. The best opportunities here are either those companies with a large installed base or a very specific niche. For example, American Software produces Enterprise Resource Planning (ERP) software, a way for clients to track their resources and allocate them efficiently. However, this market is controlled by software giant SAP AG (SAP), who has much larger financial resources, name recognition, and client base. For the companies with a large installed base, like Microsoft, switching costs are high because it would require a lot of time and money to re-train employees to use a new system. Additionally, since everyone knows Microsoft tools, the network effect is in place, making adoption of Microsoft's software necessary for smooth sharing of information across business segments or with customers.

6.2 Development tools

Another broad based industry, Development Tools consist of software that is used primarily for product development and management, as opposed to business process management. Companies in this industry are usually focused towards one type of product. Stocks here include Cadence Design Systems (CDNS), a provider of semiconductor design tools, and Logility (LGTY), which provides inventory management systems.

In truth, the line between Development Tools and Business Applications is not well defined to me. A stock like Autodesk (ADSK) is classified by many as a Business Applications company, but in truth it's AutoCAD software is used in product development. On the other hand, Adobe Systems (ADBE) is correctly identified as a Development Tools

company. In any case, Development Tools is a less attractive sub-sector. Here, technology can wipe out any competitive advantages. New methods or advances in product design can lead to quicker development times, cheaper production costs, etc. As in Business Applications, the best way to build a lasting competitive advantage is to become an industry standard. Take Adobe for example. Although there exist many Photoshop clones, some free and almost all cheaper, there is little risk to Adobe. Graphics professionals are trained to use Photoshop in school, are comfortable with it, and do not want to take the time to learn something else unless it provides a very great advantage.

7.0 Four Driving Factor of Return on Investment (ROI) for SaaS

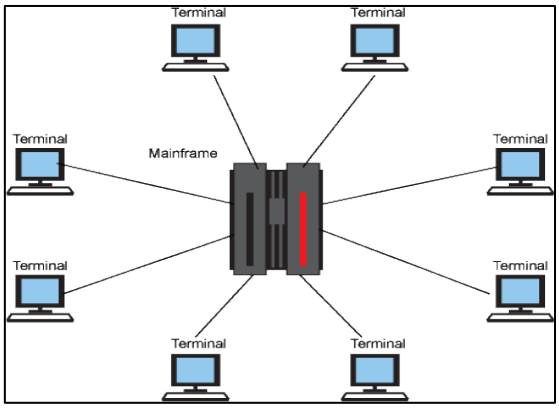
SaaS presents new opportunities for software vendors. In particular, four driving factors are cited by SaaS software vendors as leading to increased ROI:

- Increased speed of deployment
- Increased user adoption
- Reduced support requirements
- Lowered cost of implementation and upgrades

7.1 Speed of deployment

Traditional desktop applications have historically involved significant deployment hurdles. In fact, I've heard desktop application developers refer to updating their applications as a deployment nightmare on several occasions. As Tariq Ahmed states in the first chapter of *Flex 4 in Action* (Manning Press), The logistical complications of trying to get thousands, if not hundreds of thousands, of clients to run the precise version of software at the exact same time are immense. Ahmed goes on to say that such complications are so immense that most desktop software development companies don't even consider it reasonable or even feasible. Developers who have struggled with this in the past are good candidates for deploying SaaS versions of their software. However, the biggest barrier to entry into the SaaS marketplace that traditional software houses experience is enabling desktop applications to run as SaaS applications. In many cases, doing so involves re-writing the software on some level, which some companies find too cost-prohibitive. This is one of the primary reasons the movement to cloud computing has been a slow and gradual process. In most cases, the logical solution is to move software to the cloud in phases, beginning with a highly scaled-down version of the original application provided as SaaS.

Fig 1: A Simple Diagram Showing the Relationship of Client Terminals and Mainframe System in a Basic Lan



This makes obvious sense when considering the level of control the developer has on version control. It's also where the specific anatomy of SaaS plays a significant role. We can see many similarities between cloud computing and LAN computing" of years past. A typical LAN architecture consisted of an array of on-site workstations, often referred to as *dummy terminals* that ran applications by connecting to a powerful mainframe, usually provided by IBM, as seen in figure1.

This type of computing worked out nicely for enterprises, because IT departments had the ultimate level of control over versioning and updates could be deployed on an iterative and seamless basis with little or no hassle. Similarly, the logistical barriers that prevented version control for developers of desktop software applications in the past is nonexistent in the cloud, because the software runs on an infrastructure that the developing company has immediate access to.

Size and scale of a SaaS infrastructure is clearly massive in comparison to that of a LAN in consideration of the number of clients that it must be able to serve. But the underlying concept is the same. Whereas, figure 1 shows a single mainframe that is able to host enough software instances to serve all of the clients connecting to it within the local network, figure 2 shows a cloud that consists of many

different computer resources, all contributing to the total compute power able to run the many software instances required to service clients around the world.

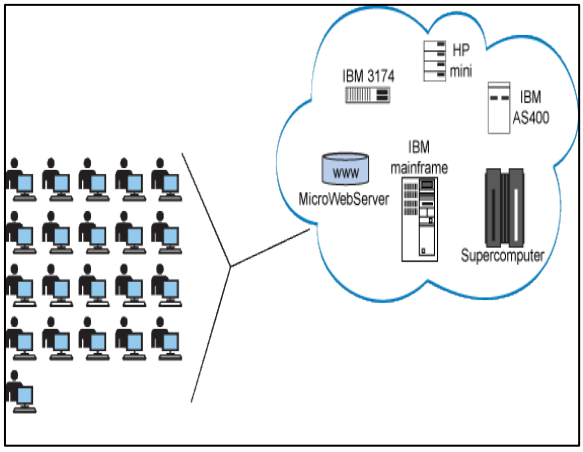
7.2 Increased adoption

If we step away from business or a moment and look at what SaaS does for the general consumer, we find that SaaS makes software available that previously may have carried too high a licensing fee to be reasonable for general consumer use. A good example is Adobe's efforts to make Adobe Photoshop an SaaS. Although this is still a work in progress and has been an evolving experiment for Adobe, it still shows progress.

For instance, I've noticed that a growing number of my friends and family members are starting to prefer the use of Photoshop.com for basic photo editing as opposed to launching the full-blown version when they need to carry out rudimentary photo-editing tasks. The significance of this is in the fact that people who doesn't require the functionalities contained in the full version will now save money. Meanwhile, others are using Photoshop.com who would not have otherwise tried it, which presents an opportunity for Adobe to capture long-term customers that it would not have otherwise had access to. The various business models for SaaS are particularly intriguing. For example, Intuit offers QuickBooks Online as an SaaS with a monthly service charge. As a business owner who does a lot of traveling, I've found this to be particularly useful, especially because my business partner lives 400 miles away in a different state. Meanwhile, Adobe leverages the power of SaaS with Photoshop.com and Acrobat.com by offering software as a *freemium* service—a term that was coined to describe a particular business model based on a scaled-down SaaS offering of a licensed software product.

Fermium SaaS is based on a revenue model where it is anticipated that a certain percentage of free users will eventually find it useful enough to upgrade to either a paid version of the SaaS that has additional features enabled or a licensed copy of a desktop version of the software that includes all the additional features and functionalities available. This tends to be a preferred method of trying out software in "restricted demo" mode compared to having to install an application on desktop that we may not end up buying. In addition, this model can be further supplemented with advertising if the ratio of free users to upgrades is less than expected. This is a common method that traditional desktop software vendors use as a way of adapting to the changing marketplace as cloud computing continues to evolve.

Fig 2: A Simple Diagrams Showing the Relationship of Client Devices to The Cloud in Saas



7.3 Reduced support requirements

The cost of large customer service help desks and the problem of increased support issues that occur as a result of having to support multiple platforms is largely mitigated with SaaS. For starters, ease of deployment allows developers to implement fixes shortly after bugs are initially found, which means that most bugs can be fixed before the vast majority of users encounter them, resulting in fewer phone calls for customer support as well as a much higher likelihood for customer satisfaction and retention.

In addition, manufacturers of traditional desktop software applications often have to support more than one platform. For example, a developer that must support the Windows 7 and Apple Mac OS X version 10.6 operating systems has nearly doubled the cost of development just by adding support for the second operating system—and this is before getting into the problems associated with supporting the many different versions of such operating systems. Supporting multiple versions of an operating system presents limitations, as well. For example, if we are building a program to run on Windows 7 but it has to be compatible with Windows XP, we have to be very careful that the features and functionalities will be able to run on both versions; otherwise, we'll be

forced to branch the project again, leaving we with separate code bases for each, which inevitably decreases productivity and efficiency and increases the time-to-completion estimates. One of the quickest ways to give a business executive a heart attack is to tell him or her that the estimated time lines for the next two years' worth of iterations have been doubled. Add to that the increase in budget for supporting different operating systems and the different versions of those operating systems, and (among other things) it's no wonder that we see such a high rate of failure for software development projects presently.

7.4 Lowered cost of implementation and upgrades

The fourth driving factor of ROI for SaaS is somewhat similar to the first. However, *speed of deployment* describes the advantages gained from being able to quickly and painlessly deploy application updates. In contrast, *lowered cost of implementation and upgrades* describes the financial benefits to the development company, which are gained as a result of having control over versioning and the infrastructure that runs the software. A big savings to the developer comes from not having the added overhead of testing and deploying bug fixes and new features on multiple platforms, because the developer has control over the platform on which the software runs—typically completely transparent to the user. This makes the upgrade path for SaaS applications less cost-prohibitive. The indirect financial result comes from customer satisfaction and retention, because the substantial savings in both time and money provide the developer with the opportunity to have a greater level of responsiveness to feature requests and enhanced usability.

8.0 SaaS User Experience Design

SaaS applications represent a sort of next-generation approach to application design. Although it might not be technically stated in any of the documents that I have seen to date, it seems that SaaS programs also include a modern approach to UI design that is more consistent with the product design process seen in most other industries. This approach includes a process known as *user experience design* (UXD), where the GUI is designed by a product team rather than the development team. The primary objective of UXD is to identify what will make the application most useful for the intended customer base and including that knowledge as part of the design. Although it can be logically argued that this should take place in the development of any kind of software, it seems to be most prevalent among SaaS application development. Perhaps the reason for this has to do with the different business models available with SaaS compared to that of traditional software as well as the substantial savings gained from developing SaaS.

9.0 Saas for Developers

As we've learned, full-blown cloud computing is a massive transition for both businesses and consumers, and many challenges must be overcome. As a result, the process will take time and go through several periods of gradual change. During this evolution in computing, it is critical for software houses to be able to adapt to the changing environment to continue to fulfill the needs of businesses and consumers alike. Just as businesses must be able to move with the changing environment as computing in the cloud evolves, software programmers will need to adapt their skill sets and understand SaaS programming models in order to stay sharp and keep them in demand. Cloud computing is not just about scalable infrastructures and platform portability through virtualization. It also takes software to an entirely new level and represents what could reasonably be considered the next generation of computer programming. For instance, affordability means wider availability, which equates to a larger potential customer base. Add to that the savings that come as a direct result of having control over the platform, infrastructure, and versioning of the software, and it quickly becomes evident that SaaS brings with it a certain level of "democratization," where small and medium-sized development shops can play on the same field as the majors.

10. Cloud Vs SaaS

We can see that cloud-computing and SaaS are two different, but closely related terms. With cloud-computing, a user is able to customize and manage an application on a server that is hosted by a 3rd-party.. We are given access to our data on those servers via the internet. With SaaS, the user no longer has to maintain either the physical servers or the cloud-

based software application. Instead, we pay a subscription to access an already developed software application via a web browser.

We don't have the responsibility of maintaining the software. We lose some control over the management and customization of the application.

nChannel is a good example of both a cloud-computing and SaaS application. nChannel provides a cloud-based multichannel management platform. We are also a SaaS platform.

Cloud-based developed a software platform that helps merchants with order management, inventory synchronization, and product information management by integrating their ERP, POS, eCommerce, and marketplace systems. This application we built can be accessed via the internet.We maintain, manage, and process data that is kept on remote servers.

We don't maintain the physical servers themselves. We just maintain the application that is run on them, which we call nChannel's multichannel management platform.

For merchants, we deliver a SaaS model. We pay us a monthly subscription fee to have access to our cloud-based application. Multiple users can use it and access it via any web browser.

We're in charge of maintaining the application and processing our data.

As we can see, cloud-computing and SaaS are two different ideas, but they work with each other to bring easy-to-access, cost-effective software applications to users.

9.0 Conclusions

We pointed out about the opportunities and challenges presented by the growing popularity of SaaS applications. SaaS is a very different model than the traditional software license and maintenance and client server model SaaS will be the way most apps will be delivered Technology innovations are the primary driver for SaaS adoption.

All but high-volume data entry for large corporates and specialized apps Much higher proportion of staff will have only PDAs or small footprint notebooks SaaS is an attractive delivery model for high-volume and commoditized business processes in back-office banking SaaS does not have to be an all or nothing value proposition.

References

[1] Saugatuck Technologies, Strategic Research Report SSR-342, Three Waves of Change: SaaS Beyond the Tipping-Point, 2007.

[2] Saugatuck Technologies, 2006 Strategic Research Report SSR-239, SaaS 2.0: Software-as-a-Service as Next-Gen Business Platform, 2006.

[3] Graham, Gordon, Editor, SoftwareCEO, 13 habits of highly effective SaaS companies, 2008.

[4] M West, B Guptill, Saugatuck Technologies Strategic Perspectives STR-458, Saugatuck SaaS Research: Waves and Platforms in the Cloud, 2008.

[5] Timothy Chou.The End of Software: Finding Security, Flexibility and Profit in the On Demand Future, SAMS Publishing, 2005.

[6] Messaging Total Cost of Ownership: Microsoft Exchange 2003 and Lotus Domino in Small and Medium Organizations, META Group 2004.

[7] R Bentley. Call in the SaaS. ,Caterer & Hotelkeeper. April 24, 2008.

[8] Fonseca, Brian. SaaS Benefits Starting to Outweigh Risks. Computerworld, 2008.

[9] D Greschler, T Mangan. Networking Lessons in Delivering Software as a Service. International Journal of Network Management 12, 2002, 317-321.

[10] Hall, Mark. On the Mark: Microsoft Pits S+S vs SaaS. Computerworld, 2008.

[11] GL Neilson, L Karla Martin, Elizabeth Powers. Software as a Service: A Comprehensive Look at the Total Cost of Ownership of Software Applications. Software & Information Industry Association (SIIA) Executive Council on Software as a Service. September 2006.

[12] GL Neilson, L Karla Martin, Elizabeth Powers. Proponents of Software-as-a-Service Say it Will Wipe Out Traditional Software. The Economist (US) 379, 8474, April 22, 2006

[13] GL Neilson, KL Martin, Elizabeth Powers. The Secrets to Successful Strategy Execution. Harvard Business Review, 2008.