

Software As A Service: A Brief Study

Barinder Kaur

PG, CSE, G.N.D.U, Amritsar, India

Abstract - *Software as a Service (SaaS), sometimes referred to as "on-demand software" is a software delivery model in which software and related data are centrally hosted on the cloud. Users typically access SaaS via a web browser. For many business applications SaaS has become a common delivery model. In this paper SaaS is discussed in detail; its basic characteristics and its architecture. Various parameters which should be followed while purchasing software and its implementation is elaborated. The pros and cons of SaaS are also discussed.*

Key Words: *SaaS, Architecture, Selection Parameters, Advantages, and disadvantages etc...*

1. INTRODUCTION

Software as a Service (SaaS) is defined [1] as a software application delivery model, where a software vendor deploys and hosts software applications in a multitenant (cloud) platform for its customers to operate the application over the Internet as services. SaaS has some unique features. SaaS applications are usually hosted at the service provider's network instead of being installed on premise, delivered as web applications, and serve as services for multiple tenants. SaaS applications can be deployed in a cloud computing environment and accessed through Internet by web browsers or users. As a result, it drastically reduces the upfront commitment of resources. As a consequence, SaaS applications can be deployed with minimal effort and be available in a very short span of time to a large group of users, and therefore, it makes SaaS model quite attractive to enterprises.

1.1 Characteristics and Benefits of SaaS

Characteristics of SaaS are:

- Network delivered access to commercially available software.
- Application delivery is one-to-many model.
- Built on optimized and robust platform.
- Customer pays for as much as they need when they need it.

Benefits of SaaS are:

- No infrastructure or software to purchase or maintain. Application and data are available anywhere with network connectivity.
- Operating costs are reduced by managing infrastructure in central locations rather than at each customer's site.
- Improved availability and reliability.
- Lower Total cost of ownership.

In the traditional model of software delivery, the customer acquires a perpetual license and assumes responsibility for managing the software. There is a high upfront cost associated with the purchase of the license, as well as the burden of implementation and ongoing maintenance. Return on investment (ROI) is often delayed considerably, and, due to the rapid pace of technological change, expensive software solutions can quickly become obsolete [2]. But with SaaS this problem is alleviated as there are various revenue options for SaaS Firms like Subscription (monthly fee per seat), Transaction based pricing (like credit cards), Profit sharing, ownership sharing and Ad-based revenue (e.g. pay per click).

2. ARCHITECTURE OF SaaS

There are two types of SAAS architecture:

- Single Tenant SAAS architecture
- Multi Tenant SAAS architecture

Single Tenant SAAS architecture: As the names suggest, single-tenant systems house the data for one company only. Single-tenant systems give a user its own database and its own instance of the software application. Placed on its own individual server, or segregated via extensive security controls to create its own virtual server, users of single-tenant systems enjoy the benefits of significant configurability of software, robust functionality, and enhanced security. An on-demand model, single-tenant SaaS is best seen as a "custom fit" solution that many companies should use because their industry, geography or security requirements give them the need for configurability and customization. A single-tenant system will have a higher degree of inherent security.

Multi Tenant SAAS architecture: Multi-tenant systems house the data for multiple companies on one server. Multi-tenant systems place the data from multiple companies on the same server, generally separating them from each other via a simple partition that prevents the

data from migrating from one company to another. As the data is housed on the same server, each of the companies using the software is running the same basic application, with the same basic functionality and with the same limited configuration capabilities. A true ASP model, multi-tenant SaaS is an “off-the-rack” solution that can fit many companies without the need for much alteration. Functionality of software is significantly greater for single-tenant systems than for multi-tenant systems. In the multi-tenant option, modifications to the software are limited because multiple customers are running the same instance of the software and because their data is being housed in a pre-configured database format. This is not to say that the functionality isn't good. In fact, the opposite is generally true. For many companies with more basic database and information needs, configurability of software isn't a necessity. Multi-tenant SaaS providers generally do a very good job of anticipating the needs of current and prospective customers and the standardized functionality is often all that is needed by a company. For that reason, some vendors actually provide customers with a choice between a single- and multi-tenant option. a multi-tenant system as the potential for data migration from one company to another is eliminated when housed in a non-shared environment. Having said that, multi-tenant systems are still secure and the level of security offered by these systems may be adequate for a particular company's needs. [4] [5]

3. SaaS PRODUCT SELECTION PARAMETERS

When several vendors offer SaaS based products, the selection of product becomes a key issue. It involves analysis of selection parameters and product offerings of the vendors. As multiple criteria are involved in decision making, it is a multi-criteria decision-making (MCDM) problem. Being a problem involving multi-criteria and multi-products, it can't be solved with mere judgment or intuition. The judgments may work fine, only when the selection parameters are few. Many factors are involved in selection of a software product. Based on experience and interviews with the experts, we propose factors for SaaS selection such as:

- Functionality
- Architecture
- Usability
- Vendor Reputation
- Cost

These factors are selected primarily considering our case study of sales force automation. [3]

3.1 Functionality

Functionality factor includes attributes that are typically called as functional modules of SFA. It includes:

- Contact and activity management for tracking customer contacts. It ensures sales efforts are not duplicated. Opportunity Management helps track and manage opportunities through every stage of the sales pipeline. It includes functionality such as lead creation, lead-to-opportunity conversion, opportunity tracking, etc. Sales Performance Management supports territory and quota assignment to multiple levels of sales organizations from regions and districts to individual sales persons. Sales Analysis module provides dashboards and reports.

3.2 Architecture

The architecture factors are as follows:

- Integration attribute includes ability of product to integrate with other applications. Integration attribute becomes quite relevant for SaaS products as SaaS products are hosted off premise and hence can be perceived as difficult to integrate with the on-premise legacy systems.
- Scalability refers to the SaaS product's ability to maintain reasonable response time for users even during peak load.
- Reliability refers to the SaaS product's ability to remain available for the users for given time windows. It requires vendors to deploy monitoring and diagnostic tools.
- Security is considered to be the major concern for SaaS products. Vendor having certifications such as ISO 27000 helps ensure security adopted for handling of customer data.

3.3 Usability

Usability related attributes are as follows:

- User interface includes facets such as intuitiveness, ease-of-use for frequently required tasks and aesthetic nature of graphical elements.
- Help attribute refers to availability of easy-to-use user manuals, eLearning modules, and
- Context-sensitive help.
- Support for mobile device has become important as modern sales workforce extensively
- Depends on the mobile devices such as PDA etc.
- Offline support is important. It means the SaaS products support a mechanism to let users work on system in offline mode and let them synchronize once connected to internet.

3.4 Vendor Reputation

Vendor reputation factor includes two attributes:

- Number of clients/users indicates the level of usage, which roughly indicates whether the

product is fairly new entry or is well-established one.

- The brand value of vendor is also important, as sometimes a new product from well-known vendor may be preferred over a product having vast customer base but being provided by not-so well-known vendor.

3.5 Cost

Cost factor includes two attributes:

- Annual subscription.
- One-time implementation cost.

Usually, cost of hardware and support personnel is covered under annual subscription, while cost of initial consulting, configuration efforts, etc is covered under one-time implementation [3].

4. SaaS IMPLEMENTATION STEPS

- Understand your business objectives and definition of a successful outcome (idea)
- Select and staff your services delivery team (people)
- Define and understand the infrastructure needed to deliver your SaaS application (hardware)
- Select your hosting facility and Internet Service Providers (ISPs)
- Procure the infrastructure and software required to deliver your SaaS application (security your platform)
- Ready to Run
- Deploy your SaaS delivery infrastructure
- Implement disaster recovery and business continuity planning
- Integrate a monitoring solution
- Establish a Network Operations Center (NOC), Client Call Center and ticketing system
- Design and manage Service Level Agreements
- Document and manage the solution while open your business [2].

5. ADVANTAGES AND DISADVANTAGES OF SaaS

Advantages of SaaS are as follows:

- Lower Cost of Ownership
- Focus on Core Competency i.e. the IT saving on capital and effort allows the customer to remain focused on their core competency and utilize resources in more strategic areas.
- Access anywhere means users can use their applications and access their data anywhere they have an Internet connection and a computing device. This enhances the customer experience of the software and makes it easier for users to get work done fast.

- Freedom to Choose (or Better Software): It means the pay-as-you-go (PAYG) nature of SaaS enables users to select applications they wish to use and to stop using those that no longer meet their needs. Ultimately, this freedom leads to better software applications because vendors must be receptive to customer needs and wants.
- New Application Types: Since the barrier to use the software for the first time is low, it is now feasible to develop applications that may have an occasional use model. This would be impossible in the perpetual license model. If a high upfront cost were required the number of participants would be much smaller.
- Faster Product Cycles: Product releases are much more frequent, but contain fewer new features than the typical releases in the perpetual license model because the developer know the environment the software needs to run. This new process gets bug fixes out faster and allows users to digest new features in smaller bites, which ultimately makes the users more productive than they were under the previous model. Additionally, it is not necessary for the customer to continually upgrade the software. Each time the user accesses the software, it is the "latest and greatest" version that's available. [2]
- Increased Total Available Market: Lower upfront costs and reduced infrastructure capital translate into a much larger available market for the software vendor, because users that previously could not afford the software license or lacked the skill to support the necessary infrastructure are potential customers. A related benefit is that the decision maker for the purchase of a SaaS application will be at a department level rather than the enterprise level that is typical for the perpetual license model. This results in shorter sales cycles.
- Enhanced Competitive Differentiation: The ability to deliver applications via the SaaS model enhances a software company's competitive differentiation. It also creates opportunities for new companies to compete effectively with larger vendors. The tangible value that customers can realize from the SaaS model versus the perpetual license model is a compelling selling point.
- On the other hand, software companies will face ever-increasing pressure from their competitors to move to the SaaS model. Those who lag behind will find it difficult to catch up as the software industry continues to rapidly evolve.
- Lower Development Costs & Quicker Time-to-Market : The main saving is at testing (35%).
- Small and frequent releases i.e. less to test. Application is developed to be deployed on a specific hardware infrastructure, far less number

of possible environment – less to test. This, in turn, provides the software developer with overall lower development costs and quicker time-to-market.

- Effective Low Cost Marketing: Between 1995 and today, buyers' habits shifted from an outbound world driven by field sales and print advertising to an inbound world driven by Internet search. The SaaS delivery model is perfect for marketing programs that exploit this shift.
- Predictable MRR Revenue: Traditionally, software companies rely on one major release every 12-18 months to fuel a revenue stream from the sale of upgrades (long tail theory). This puts a lot of pressure on the organization to hit an arbitrary date to meet corporate financial commitments. In the SaaS model the revenue is typically in the form of Monthly Recurring Revenue (MRR), which is far more predictable and less tied to the development schedule of the next release of the software.
- Improved Customer Relationships: SaaS contributes to improved relationships between vendors and customers. In the traditional model once the software is sold, it is largely up to the customer to make it work. The SaaS model creates a more symbiotic relationship between vendors and customers and provides vendors with greater opportunities to please their customers
- Protecting of IP: Difficult to obtain illegal copies. Price is low as a result making getting illegal copies totally unnecessary. [2]

Disadvantages are as follows:

- Core functionality out-sourced
- Broadband risk
- Limited personalization/tailoring
- No competitive uniqueness advantage
- Not suited to high volume data entry [2]

5. CONCLUSION

While working on the term paper, it is concluded that SaaS is becoming the mainstream. Considering its benefits both to user as well as vendor, SaaS has a widespread adoption and is becoming the most preferred software delivery model. SaaS will be the way most applications will be delivered. A single server handling multiple customers thereby reducing hardware cost. The numbers show that SaaS is a far more attractive economic model than the perpetual license model.

REFERENCES

- [1] Feng Liu, Weiping Guo, Zhi Qiang Zhao, Wu Chou, SaaS Integration for Software Cloud, IEEE 3rd International Conference on Cloud Computing, 2010.
- [2] Jie Liu, SaaS-What is it and why is it important to you?, Sapient LLC, available at <http://www.wou.edu/~frli06/SAAS-2.ppt>
- [3] Manish Godse, Shrikant Mulik, An Approach for Selecting Software-as-a-Service (SaaS) Product, IEEE International Conference on Cloud Computing, 2009.
- [4] Single-Tenant vs. *Multi-Tenant* SaaS Architecture, <http://www.sapiensoftware.com/multitenant.aspx>
- [5] Software as a Service, http://en.wikipedia.org/wiki/Software_as_a_service.
- [6] Barinder Kaur, "Parametric Analysis Of Cloud Computing Security Models", International Journal of Information And Computation Technology (IJICT), vol., no., pp.1499,1506, June 2014.
- [7] Barinder Kaur, "Cloud Computing And Security Issues: A survey", International Journal of Computer Science Trends And Technology (IJCST)", Mar-Apr 2015.
- [8] S. Patidar, D. Rane, P. Jain, "A Survey Paper on Cloud Computing," Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on , vol., no., pp.394,398, 7-8 Jan. 2012.

BIOGRAPHY



Barinder Kaur was born in Hoshiarpur, Punjab, India. She has obtained her B.Tech degree in Computer Science and Engineering from PTU in 2012. She has completed her Masters degree in Software System from G.N.D.U., Amritsar. She is interested in research work on Cloud Computing.