

December 2007

The Emergence of On-Demand Software Aggregators: Implications for Developers, Customers, and Software Companies

Julie David
Arizona State University

Arti Mann
Arizona State University

Follow this and additional works at: <http://aisel.aisnet.org/amcis2007>

Recommended Citation

David, Julie and Mann, Arti, "The Emergence of On-Demand Software Aggregators: Implications for Developers, Customers, and Software Companies" (2007). *AMCIS 2007 Proceedings*. 194.
<http://aisel.aisnet.org/amcis2007/194>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2007 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

The Emergence of On-Demand Software Aggregators: Implications for Developers, Customers, and Software Companies

Julie Smith David

W. P. Carey School of Business
Arizona State University
Julie.Smith.David@asu.edu

Arti Mann

W. P. Carey School of Business
Arizona State University
arti.mann@asu.edu

ABSTRACT

A growing segment of the software market is On-Demand Software Aggregators (ODSAs): firms that provide a platform for many developers to provide preconfigured, pre-integrated functionality over the internet. These aggregators act as electronic marketplaces for software, intermediating between developers and customers, and they supplement their own development efforts with those of other Independent Software Vendors. This paper applies Transaction Cost Economics to the ODSA phenomena, identifies the risks and costs associated with this model, and makes recommendations to customers, developers and ODSA providers to help manage the relationships.

Keywords

On-Demand Software, SaaS, Transaction Cost Economics, Salesforce.com

Introduction

The market for Enterprise Applications is changing, and new models for software delivery are emerging. Firms have invested as much as 50% of their capital expenditures in traditionally-delivered, in house IT investments (Carr, 2003), and IT spending is projected to increase 7.6% between 2006 and 2007 (Alter, 2007). However, the On-Demand software market segment which offers software to customers over the internet is becoming more established (Weier, 2007) and is expected to grow as much as 21% annually (growing from \$5.5B in 2005 to \$10.7B in 2009) (Merrill Lynch, 2006). This approach to software delivery allows customers to substitute stable monthly payments for large and unpredictable expenditures in IT infrastructure, but the implementations are not always pain free. Firms with requirements for process customization and/or systems integration will face many of the same challenges they would experience with a traditional implementation (Overby, 2006).

Therefore, a secondary market change is currently underway: vendors are integrating multiple On-Demand products. Some of these vendors offer their services to mimic traditional integration efforts providing On-Demand Enterprise Application Integration (EAI). Other vendors are creating platforms to aggregate software applications, and are encouraging developers to design modules that will automatically integrate with the base product. These On-Demand Software Aggregators (ODSA), such as Salesforce.com's AppExchange, are working to provide much more choice to customers, a larger market for developers, and easier implementations for all.

This paper uses transaction costs economics, coupled with a case analysis of the Salesforce.com On-Demand marketplace, to develop a theoretical model to describe the ODSA phenomena. The following sections provide an introduction to the On-Demand software delivery model, the characteristics of Salesforce.com's ODSA platform, and a theoretical analysis. From these perspectives, a model of ODSA is developed, and from that model, recommendations are posed for the ODSA provider, their developers and customers, and future research.

ON-DEMAND SOFTWARE – BACKGROUND

On-Demand solutions, also known as SaaS (Software as a Service), are computing services which are made available to customers usually over the Internet. The need for On-Demand software emerged because small- to medium-sized businesses could not afford business application software, specifically Enterprise Resource Planning (ERP). Furthermore, such software is so complex that firms adopting them face high acquisition costs. Most On-Demand software has been designed to minimize the amount of configuration required, so implementation complexities and associated costs can be significantly reduced. Also, upgrading and patching the software is performed centrally by the SaaS vendor, and the customers face none of the challenges associated with these activities. The SaaS provider is responsible to maintain reliable services, including round the clock technical support, security, disaster-recovery, and business-continuity planning. Finally, due to the economies of scale from servicing a large number of customers from each data-center, the cost per customer for all these activities is cheaper than if similar services were to be maintained at each client's location.

The On-Demand market segment has grown from the pioneering companies, to today's marketplace in which a large number of vendors are offering SaaS solutions including Microsoft, SAP, Oracle, and Salesforce.com. Low-cost, round the clock availability, seemingly endless scaling, and access from anywhere have helped push the SaaS model into acceptance (Weier, 2007).

ON-DEMAND SOFTWARE AGGREGATORS

As On-Demand solutions have been gaining market share, customers have faced a new challenge: many of the SaaS vendors provide functional, rather than enterprise-wide, solutions. For example, RightNow and Salesforce.com began as CRM providers; Employease, and WorkDay provide human resources-related solutions. In addition, there are many even smaller point solutions that are being adopted to solve industry-specific (such as Vocus which provides on-demand solutions designed for the public relations industry) or never-before-automated business challenges (such as SuccessFactors which offers a range of tools including modules to gather employee satisfaction data, automate performance improvement, and manage goals). Because there are so many applications, from so many vendors, organizations needing a comprehensive solution to support their organizations' processes must integrate On-Demand functionalities.

Although some single-source, enterprise-wide solutions are being offered On-Demand (such as NetSuite or Intacct), ODSAs are also emerging to provide one on-line platform that can integrate disparate on-demand solutions. These aggregation platforms act more as electronic marketplaces, linking component developers to customers, and providing the integration architecture to link the components. Salesforce.com's Appexchange and NetSuite's Suitflex are two markets. In both cases, the vendors have developed proprietary platforms and are encouraging external developers to design and implement complimentary solutions, following their proprietary standards. The result is a portfolio of pre-integrated solutions that are available through a portal. Such offerings are the exemplars for the emerging ODSA market

The following section provides an overview of Salesforce.com's important role in establishing the On-Demand market segment and in defining the role of ODSAs.

SALESFORCE.COM

Salesforce.com was founded in 1999 by Marc Benioff and a small team of engineers. It was founded on the principle of offering Software as a Service with its focus primarily on CRM applications. Since then it has grown significantly: in 2006 its 1304 employees generated \$309.9 million in revenue (a 76% increase from 2005) and profit grew 288% to \$28.5 million (Shafer, 2007).

Salesforce.com initially targeted small to midsize companies and helped them in automating their sales operations. By making their offering purely web-based and selling it through monthly subscriptions, Salesforce offered a competitively priced product that required no maintenance or local personnel. They were quickly able to garner customers and rapidly build a sizable client base. Salesforce grew from a small web-based automation software company to a full CRM product offering company over three years.

Salesforce.com's ODSA solution has two major components: Appexchange is the customer-facing marketplace that showcases and delivers applications, and Apex is the developer platform with tools and infrastructure components that can be used to create and host pre-integrated applications. The following sub-sections provide more insights into these components.

Appexchange

Although the market viewed Salesforce.com as a CRM provider, Benioff had a much more comprehensive vision for his firm. Salesforce launched Appexchange, an On-Demand software aggregation platform, at Salesforce.com's annual user conference in September, 2005. Benioff described his ambitious plans for Appexchange as “an On-Demand eBay type of market place that would radically transform the software industry (McCabe, 2005).”

At the highest level, Appexchange is a platform that, for a monthly fee, provides customers the choice of applications from hundreds of web-based, pre-integrated modules. Although Salesforce.com initially seeded the Appexchange website with small applications and components to start building the application/component ecosystem, they also made their development standards public and encouraged other developers and ISVs (independent software vendors) to contribute to Appexchange. Now there are over 500 applications available, ranging from generic extensions to the Salesforce.com platform to niche components targeting specific solutions such as analysis tools, dashboards, and workflow-related extensions to the main system. Appexchange hosts applications which interact with the Salesforce systems and access individual customer's data, and it can host generic applications that developers want to sell through Appexchange. Additionally, because the development standards are public, customers are free to create their own solutions, and even to make these available to others (such as trading partners) through Appexchange.

Apex

To foster the creation of the depth and breadth of high-quality modules that customers require, Salesforce has created Apex, their proprietary environment targeted towards developers. This platform enables developers to program, build and test applications without making any investments in infrastructure or operations. As such, developers can focus on their core competencies without having to worry about network reliability, system availability, scalability, etc.

Although the Apex platform shares some of the concepts with traditional platforms like PC operating system, the technologies and models behind Apex are quite different. The main features of Apex platform are shown in Table 1.

Table 1: The Apex Platform: Key Concepts and Features (Directly from Salesforce.com 2007b)

Concept	Description
Multi-Tenancy	Application model in which all users and apps share a single, common infrastructure and code base.
Apex Builder	Metadata-drive app development model that allows apps to be defined as declarative “blue prints,” with no code required.
Apex API	Web services API that provides direct, low-level access to all data stored in the Apex Platform from virtually any programming language and platform.
Apex OS	Capability to run multiple applications within a single Salesforce deployment that share a common data model, security model, and UI.
Apex Code (upcoming)	An on-demand programming language that lets developers create new business logic and behaviors that run entirely on Salesforce.com servers.
AppExchange Directory	iTunes-like Web site where visitors can share, review, demo and install apps built on the Apex platform.

Making use of the different features and models of the Apex platform, developers can develop different types of applications: native or composite (terms coined by Salesforce), based on the needs of their clients. The native applications are built using Apex Builder. They are primarily created using metadata and these applications run entirely on the Apex platform. The main drawback of these applications is that these are limited by the capabilities of the Apex Builder. The composite applications, on the other hand, use a combination of capabilities provided by the Apex Builder and API, and run outside of the Apex platform. The developer is responsible for the infrastructure and systems that run, maintain and manage these applications/services.

Apex remains a highly proprietary platform, and applications developed using Apex and hosted on Appexchange can only be used by Salesforce.com's customer. From a developer's perspective, the availability of a platform which has a built-in

workflow management engine and enables the creation of complex applications using data models objects, with no need to worry about hosting, deployment, or server management is a key step forward in the evolution of the SaaS model.

THEORETICAL UNDERSTANDING OF ODSAs

Salesforce.com's impressive growth in Appexchange modules provide evidence that ODSAs are becoming a viable alternative to in-house, enterprise-wide systems. As such, it is important to rigorously evaluate the costs and risks associated with this new software delivery model. To frame this evaluation, three perspectives are integrated: 1) ODSAs are electronic marketplaces intermediating between developers and customers, 2) ODSAs and customers are electing to outsource significant portions of their IT infrastructure, rather than performing these roles internally, and 3) ODSAs are simultaneously outsourcing R&D activities while also performing them internally.

As discussed above, Salesforce.com explicitly recognized their shift from a leading SaaS CRM provider to "The World's On-Demand Applications Marketplace." Like other ODSAs, they help match software users with developer's products through their on-line marketplace which provides information about products from a wide range of developers and for developers provides visibility with these customers that they would not be able to attain on their own. ODSAs also facilitate the resulting transactions. Their platforms must be able to control access to the products, bill for system usage, and increase the trust between the customer and supplier. Finally, markets also streamline the contracting process and may provide dispute resolution services (Bakos, 1999).

The second way of understanding the role of ODSAs in the marketplace is to consider firms' outsourcing choices. Research has demonstrated that a wide range of shared and standardized technologies are adopted in knowledge intensive environments that require reach and range (Broadbent, Weill, and Neo, 1999). Such technologies become part of the firm's IT Infrastructure, and allow the firm to enjoy benefits from standardized data and processes that are available to the organization as a whole; additionally, managing the investments centrally allows for economies of scale to control costs. This trend appears to be continuing with the transition to On-Demand services – the costs to integrate disparate systems can be shared by a larger number of firms. Customer firms who outsource their IT have determined that focusing on core competencies is more critical than maintaining technological expertise in-house. Simultaneously, ODSA platforms are outsourcing at least a portion of their research and development efforts to external systems developers and ISVs.

The third lens to apply to the ODSA business model is the strategy of Taper Integration which recognizes the strategic benefits of simultaneously insourcing and outsourcing a function to enhance product introductions and profitability (Rothaermel, Hitt, and Jobe, 2006). By choosing this organizational structure, ODSAs are able to introduce new products (application modules) more quickly, and enjoy rapid knowledge sharing with their developers, improving their innovation processes. Salesforce.com is an example of this strategy as they are simultaneously adopting vertical integration (by developing modules internally) and strategic outsourcing (through opening Apex to the market and providing solutions by external developers through Appexchange).

These three views of the role of ODSAs are driven to a large extent by relative costs: how do costs differ with different software delivery models? Therefore, this paper applies this particular aspect of Transaction Cost Theory to this domain (for now governance structure is not included). This theory has been the foundation for explaining management's rationale for outsourcing and vertical integration choices (Afuah, 2003, Rothaermel et al., 2006, Williamson, 1975), and it applies to ODSAs because of the complex outsourcing activities that are taking place. Thus, this theory is applicable in case of Salesforce because it provides us with a unique vantage point to study the costs and interactions between the ODSA (using Salesforce's Appexchange marketplace as an exemplar), its customers and external developers.

Transaction cost can be grouped largely into two components— production costs and transaction costs. Production costs can be defined as the costs incurred by the supplier in the production of component. In the case of Salesforce, costs include the cost of developing a platform and also the cost of maintaining and providing services includes databases, servers, etc. In case of developers it is the cost associated with developing the applications either by using native platform or by using composite platform.

Transaction costs can be further divided into three distinct components: coordination cost, operation risk and opportunism risk (Clemons, Reddi and Row, 1993).

Coordination Costs include costs of searching for trading partners; collecting information about products and trading partners; negotiating, writing, monitoring and enforcing contracts; and transporting materials (Afuah, 2003, Oster, 1999, Williamson, 1975). ODSAs play a key role in helping to minimize coordination costs: they intermediate between multiple developers and customers, coordinating the contracting process, making product information available, and insuring that the systems work together seamlessly.

Operation risk is the risk that the other parties in transaction willfully misrepresent or withhold information or under perform their agreed upon responsibilities (Clemons et al., 1993). For instance, a developer who has agreed to provide a software component of a particular quality may actually provide one of inferior quality, fully knowing that it would be difficult for the customer to test it.

Opportunism risk includes the risk associated with lack or loss of bargaining power directly resulting from the execution of a relationship. Three important sources of opportunism risk were identified by Clemons et al (1993), namely, relation specificity, number of potential suppliers for the product and loss of resource control. Perhaps most importantly, the customer may lose control of their data when it is being stored and maintained at the developer's site.

Based on our discussion of TCE above, Table 2 provides an analysis of costs and risks associated with the different roles in the OSAD relationship, and variation in those based upon the production of native or composite applications.

Table 2: Factors Influencing Application Development Choices

		Developers	ODSA	Customers
Production Cost for Development	Native Applications	Reduced application development costs by using Salesforce standards, tools etc. Optional costs (webinars, training sessions)	Developing and providing platform, database design, servers etc. Development of standards	
	Composite Applications	Application development costs, including establishing network capabilities, web-platform etc. Optional costs (webinars, training sessions)	Development of standards, architecture, integration servers etc.	
Production Cost for Operations	Native Applications		Running and Maintaining DB, Servers etc.	Monthly Fee
	Composite Applications	Cost of maintaining the platform for providing the services, storing the data and running the code Annual Certification costs	Running and Maintaining servers and platform	Monthly Fee
Coordination Cost	Native Applications	Reduced search cost for customers Reduced contracting costs	Increased delivery costs (integration costs)	Reduced search cost for software providers Reduced costs for collecting info. about products Reduced contracting costs Reduced monitoring costs (only Salesforce)
	Composite Applications	Reduced search cost for customers Reduced contracting costs	Increased monitoring costs (Certification costs, random checks) Increased delivery costs (integration costs, packaged application maintenance etc.)	Reduced search cost for software providers Reduced costs for collecting info. about products Reduced contract costs Reduced monitoring costs (only Salesforce)

		Developers	ODSA	Customers
Operation Risks	Native Applications	Controlled risks because of use of License Management notifications App from Salesforce Survivability of Salesforce	Reduced risk because application developed only with components provided by Salesforce	Reduced risks because of coordination by Salesforce Increased risk from Salesforce survivability issues
	Composite Applications	Survivability of Salesforce Controlled risk of Salesforce withholding usage statistics because of availability of other outlets	Controlled application risk with extensive testing Risk of low quality support for applications	Risk of misuse as data might reside at developers site
Opportunism Risks	Native Applications	Increased risk because of sole dependence on Salesforce Salesforce uses its position and competes or promotes its applications	Non existent	Increased switching costs Reduced risks because data resides with Salesforce and not with developers
	Composite Applications	Reduced risk because of additional sales channels Salesforce uses its position and competes or promotes its applications	Exists if developers and customers develop direct relationship	Increased switching costs Increased costs because reduced bargaining power

From this analysis of the different costs and risks we find several interesting results. For example, the production costs for application development and operations range from medium to high for developers (composite applications) and Salesforce, but the costs for the customers are significantly lower as compared to the traditional client-server architecture based applications. In this case the customer only pays a monthly fee based upon the edition the customer has chosen.

Another interesting result is that although the operational risks for the developers and Salesforce are controlled, in some cases they are higher for customers either because of sole dependence on Salesforce or because of data related issues with developers for composite applications, where there is a possibility of misuse of data by developers. Opportunism risks are higher for native application developers as compared to that of composite application developers because of their sole dependence on Salesforce. They are also higher for customers because of the switching costs and reduced bargaining power.

Table 3 summarizes the ranges of different costs associated with different functions in low, medium and high category.

Table 3: Relative Costs Associated with Application Design Choices

	Developers		ODSA		Customers	
	Native Framework	Composite Framework	Native Framework	Composite Framework	Native Framework	Composite Framework
Production cost for Development	Low	High	High	Medium	NA	NA
Production cost for Operations	Medium	High	High	High	Low	Low
Coordination Costs	Low	Low	Medium	High	Low	Low
Operation Risks	Low	Medium	Low	High	Low	Medium
Opportunism Risks	High	Medium	Low	High	Medium	High

Based on our discussion of Salesforce as an example of an ODSA and the different costs and risks associated with different aspects of trading partner interactions, we propose recommendations for developers, customers and ODSAs.

RECOMMENDATIONS

Developers

The major benefit for developers participating with an ODSA is market penetration. The ODSA can provide access to a very large market, so partnering with them can be valuable, especially for small and medium size developers/ISVs that have limited resources. Additionally, coordination costs and contracting costs are significantly reduced for developers, thus it is doubly beneficial for small to mid size companies.

We predict that developer choice will evolve over time. Early stage developers will be attracted to the native environments with their low infrastructure requirements and start up requirements; they will develop relatively small modules that integrate more directly with the ODSA base products. Over time, however, as the developer acquires assets and experience, it is likely they will choose to develop more sophisticated applications, and will, at some point, need to move beyond the native platform tools. If they develop solutions on their own technologies, they will suddenly have the ability to extend the range of their distribution channels: they can continue to participate in the ODSA, but they will also be able to partner with other ODSAs and even provide the functionality directly to customers. Thus, we predict that more sophisticated developers will be more likely to use composite development techniques, whereas less sophisticated developers will use the native techniques.

Finally, we predict there will be opportunities for developers of large- and small-scale solutions. If the developers create large applications, they will require more components, and, as such, will need to develop their solutions on the composite platform. If they integrate the components in a unique way that solves customer's pain points, they will be able to generate revenues.

Developers of smaller solutions will need to offer high volume, niche web services. For example, credit checking services are relatively simple (compared to full-scale sales processes), but could be used with every order entry transactions. In this case, even micropayments per transaction could produce significant revenues and profits. To limit imitation, niche developers will need to gain market share quickly so customers face switching costs, their services must be extremely secure and reliable, and they must scan the market constantly, looking for competitors with improved functionality.

Customers

Customers have several reasons for sourcing software services from ODSAs. In addition to the motivators for using On-Demand solutions (stable, monthly payments, reduced IT requirements, automatic version updates, etc.), the ODSAs provide access to a wide range of pre-tested and pre-integrated software, and the coordination and contracting costs are significantly reduced because customers rarely interact with the ISV developers whose software is available on the integrated platform. The customers need to be aware of the tradeoffs involved in choosing native or composite applications. Native applications

are completely hosted on Salesforce.com's platform and are developed using Salesforce components. However, in case of composite applications, even though the infrastructure is provided by Salesforce, the developers have a higher degree of freedom when developing the application, including the ability to store customer data on the developer's servers. This increases customers' operation and opportunism risks because even if Salesforce can be trusted and has high service standards, composite developers may not be able to meet those standards. Additionally, in this case, Salesforce does not accept the responsibility for data stored by external developers (See Figure 2).

Does the publisher of an app I install from the AppExchange have access to my data?

In some cases, yes. The access the publisher of an app has to the data inside your deployment depends on how that app was built and whether it integrates with external services (such as an email marketing system or a third-party data service). Contact the publisher of the app to determine if this is true for a given app, and remember that when it comes to apps that have access to your data, you should pay close attention to the trustworthiness of the publisher.

Figure 1: Salesforce.com FAQ for Customers Regarding Developer Access to Customer Data (Salesforce.com, 2007c)

On-Demand Software Aggregators

As discussed above, ODSAs act as the market intermediary between large number of developers and customers. From a technical perspective, they need to make large investments in developing the platform and related technologies, standards, training materials, certification, etc. Therefore, production costs associated with development and operations can be significantly higher than if they offered only their own solutions. However, if they achieve the synergistic benefits of their taper strategy, they will be able to release new functionality much more quickly than if they were the only developer or merely an ASP host.

To earn adequate returns on its significant infrastructure investments, the ODSA must act as "market maker" and adopt strategies that will not only reduce operation and opportunism risks for customers but will also reduce development costs for developers. In general, a key role of the ODSA is to build interorganizational trust between the parties (such as described in (Ratnasingham and Kumar, 2000, Sydow, 1998). They have to maintain a balance so that they can attract both developers as well as customers. The following provides several suggestions for how they can add value to these relationships.

First, because the ODSA expects a long-term collaborative relationship with its developers, literature extending transaction cost theory would predict that the contracts between the developers and ODSA firms will be recurrent and/or relational, relying on non-contract mechanisms to control the situation, build trust, and help insure that the developers will be successful (Ring and Van de Ven, 1992).

Second, because the developers rely on the ODSA for the platform and access to the customers, the ODSA must be reliable, stable, robust, and follow standards to help them streamline development and integration efforts. The ODSA must support open communication with the developers (such as giving timely adoption and usage statistics, alerts for operational problems, etc.). Additionally, they must embed functionality in their platform that will simplify the developers' administrative requirements, such as providing robust billing and credit features, contract management, collecting funds from customers, and transmitting revenues to the developers automatically.

The major concern for customers is the operations risk they face, and the ODSA must strive to enhance trust to minimize the customer's risk assessment (Ring and Van de Ven, 1992). The most obvious risk to customers is that of security. In all On-Demand environments, the customer must trust that the provider has adequate security abilities (and, in fact, with economies of scale, the ODSA should have higher security capabilities than individual customers could afford). In the case of ODSAs, the situation is even more imperative: the ODSA must provide assurance that the developers (especially those of composite applications) operate securely. Salesforce has made great strides in this area. First, they have a certification program in which they perform rigorous tests of developer applications, especially those that have been developed as composite applications. Additionally, their Appexchange marketplace provides several different types of information for customers making a software selection decision as shown in Figure 3.

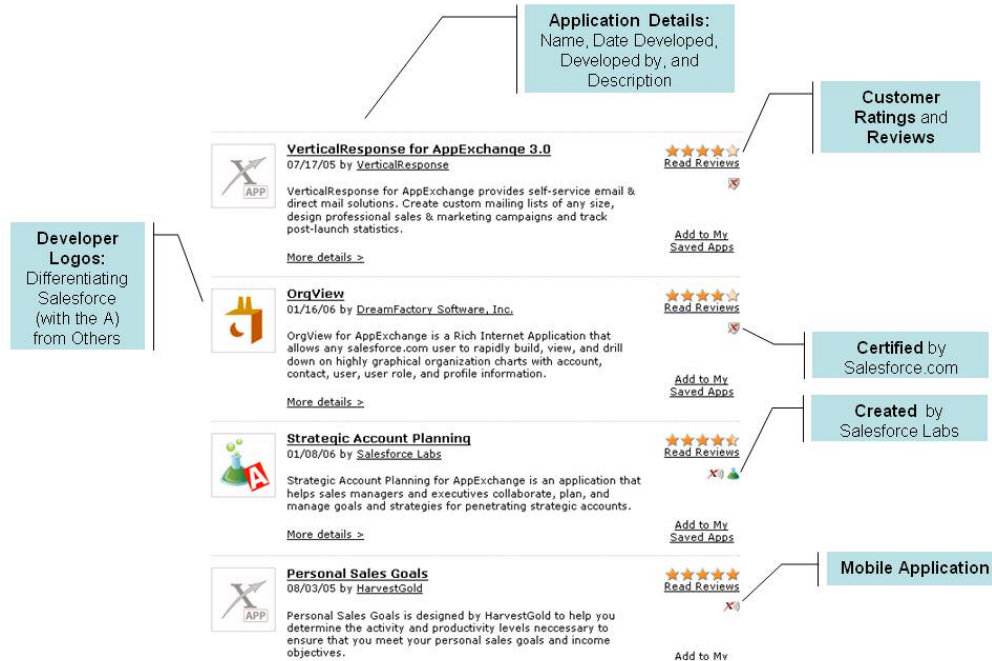


Figure 2: Information to Build Confidence (Salesforce.com, 2007a)

In the future, we predict that ODSAs will embrace additional certifications to help customers. For example, SAS 70 Type II certifications of financial applications (such as Intacct has already earned), will help customers who are audited and must meet Sarbanes-Oxley requirements either because they are publicly traded or because their key customers are.

CONCLUSION

With increasing adoption of web services, Internet access, and growing confidence in On-Demand software services, On-Demand Software Aggregators are gaining market share. These firms are developing their own, proprietary standards, and are locking in their customers who test and adopt functionality from multiple sources. Although we see firms gaining traction, we believe that the market is currently striking a delicate balance between risk and trust. Customers appear to trust the major ODSAs, but it is unclear if they are fully aware of the inherent risks, especially of the composite applications being developed. Similarly, the ODSA firms must trust that developers (again, especially those of composite applications) will be able to deliver reliable solutions. If a customer fails due to a third party developer application, the ODSA model will suffer significantly.

There are many opportunities for future research to explore the intricacies of the ODSA business model from at-least three different perspectives (Customer-ODSA, Developer-ODSA and Customer-Developer). The ODSA firms are collecting detailed data regarding the applications, adoption and usage, and customer feedback. These data would provide rich insights into Taper integration and TCE. Further analysis of the relationship between trust, risk, and contractual structures could be performed in this setting. Finally, longitudinal studies of customer adoption patterns will likely provide a better understanding of the software selection process and the value of both IT infrastructure (given that it is being largely outsourced in ODSA environments) and in emerging functional solutions produced by developers who are enjoying easier access to a customer base than they could in the past.

REFERENCES

1. Afuah, A. "Redefining firm boundaries in the face of the Internet: Are firms really shrinking?," *Academy of Management* (28:1) 2003, pp 34-53
2. Alter, A. "IT Spending: Turning on the Afterburners," C. Insight (ed.), Ziff Davis, 2007.
3. Bakos, Y. "The emerging role of electronic marketplaces on the Internet," *Communications of the ACM* (41:8), August 1999, pp 35-42.
4. Broadbent, M., Weill, P., and Neo, B.S. "Strategic context and patterns of IT infrastructure capability," *Journal of Strategic Information Systems* (8) 1999, pp 157-187.
5. Callaghan, D. "Siebel closes on UpShot CRM deal," in: *eWeek*, 2003.
6. Carr, N. "IT Doesn't Matter," *Harvard Business Review* (81:5), May 2003 2003, pp 41-50.
7. Clemons, E.K., Reddi, S.P., and Row, M.C. "The impact of information technology on the organization of economic activity: The "move to the middle" hypothesis," *Journal of Management Information Systems* (10:2), Fall 1993, p 27.
8. McCabe, L. "AppExchange: Saelsforce.com's next big bet," in: *AMI-Partners SMB Research Partners*, AMI-Partners (ed.), New York, 2005.
9. Merrill Lynch, quoted by Internet Capital Group "Investor Presentation," 2006.
10. Overby, S. "The truth about on-demand software," in: *CIO*, 2006.
11. Ratnasingham, P., and Kumar, K. "Trading partner trust in electronic commerce participation," in: *Proceedings of the Twenty First International Conference on Information Systems*, Association for Information Systems, Brisbane, Queensland, Australia, 2000.
12. Ring, P.S., and Van de Ven, A.H. "Structuring cooperative relationships between organizations," *Strategic Management Journal* (13:7), October 1992, pp 483-498.
13. Rothaermel, F.T., Hitt, M.A., and Jobe, L.A. "Balancing vertical integration and strategic outsourcing: Effects on product portfolio, product success, and firm performance," *Strategic Management Journal* (27) 2006, pp 1033-1056.
14. Salesforce.com "AppExchange - On-Demand Applications from salesforce.com," 2007a.
15. Salesforce.com "Creating Applications with the Apex Platform," 2007b.
16. Salesforce.com "Frequently Asked Questions: Understanding the AppExchange," 2007c.
17. Shafer, S. "salesforce.com inc. information and related industry information from Hoover's," Hoovers (ed.), 2007.
18. Sydow, J. "Understanding the constitution of inter-organizational trust," in: *Trust Within and Between Organizations, Conceptual Issues and Empirical Applications*, C. Lane and R. Bachmann (eds.), Oxford University Press, Oxford, 1998, pp. 31-64.
19. Weier, M.H. "New Deals Show Software-As-A-Service More Than A Niche " in: *InformationWeek*, 2007.
20. Williamson, O. *Markets and Hierarchies* Free Press, New York, 1975.