Chapter 12

Using Software as a Service

In This Chapter

- Looking at the origins of SaaS: Salesforce.com
- Understanding how the SaaS model works
- Understanding the economics and the ecosystem

"When did Software as a Service get its start?" might sound like a straightforward question, but it isn't. In one way, you could say that when timesharing systems were all the rage more than 30 years ago, all software was delivered to customers as a service.

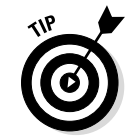
Mainframe systems were simply too expensive for most companies to buy their own systems. A couple of decades later, minicomputers, servers, and personal computers changed the dynamics of the market. Economically, it was feasible for any Tom, Dick, and Harriet to own their own systems and the software. Not all software moved to an internal model however. (Software such as ADP's payroll system, for example, remained Software as a Service.)

Two key events converged to create the model that we now call *Software as a Service (SaaS):*

- First, the Internet became a commercial platform.
- Second, software costs and complexities became so difficult that running, upgrading, and managing software become too complex for many companies to manage. This was especially true for small- and medium-sized companies that didn't want the expenses of managingall the components. These companies were the first to embrace this new generation of SaaS.

Today, SaaS is the most mature area of cloud computing. SaaS gained initial traction with the *customer relationship management (CRM)* market and has expanded into others — particularly the collaboration market and the enabling tools and management environments. In this chapter, we explain what SaaS is, talk about its business model, and discuss the types of vendors that are in the market today.

SalesForce.com's Approach to Evolving Software as a Service



What's inside the development environment? And why are we talking about this now? SalesForce.com's approach to its platform is similar to many of the platforms on the market. Therefore, understanding what a platform is built on will help you make decisions.

You may not see how the product is made, but you're still responsible for the integrity and security of your data and how well the application works.

Salesforce.com software environment

Here is a list of the components of the Salesforce.com software environment:

- **Multi-tenancy foundation:** Within a *multi-tenant architecture,* each user's private code is stored in a separate container and is isolated from other containers.
- **Metadata:** *Metadata* is an architectural approach that allows each user's customized logic and data to be managed separately.
- **Infrastructure:** This data center environment underlies the Salesforce.com application environment. It includes capabilities such as middleware, security, and database management. It also includes performance management and monitoring.
- **Database:** There is a set of database services that sits on top of an Oracle database. It includes ways to manage data objects and fields, as well as documents that are used by the Salesforce.com application.
- **Integration:** This set of standardized Web services APIs enables applications to have a common approach to access information from one application to another (as well as data from other enterprise applications). If there's a standardized way to link one SaaS application to another service, customers or implementers don't have to resort to custom coding.
- **Logic:** This component includes services that create business processes (such as workflow, approval processes, and so on) that the application uses.
- **User Interface:** This includes a framework and tools to build the way the application appears to the customer.

Digging into the origins of SaaS

You could probably find many examples over the years of companies that offered their software products as a service. But to keep things simple, we start with the company that really put Software as a Service as we know it today on the map — Salesforce.com. We think that the story of how SalesForce.com started and how it has evolved says a lot about this market.

Marc Benioff, the founder of SalesForce.com, had been a marketing executive for Oracle for many years. After leaving and going off on his own, he started Salesforce.com. Being a marketing executive, Marc had a bold marketing moniker for his fledgling company: No software. The plan was quite simple: Create a way to allow customers to use a popular application — *customer relationship management (CRM)* over the Internet. Customers would purchase a seat and could use the application over the Web. The customer never had to update the software, didn't have to store data on a server, and never had to worry about maintenance fees. If that customer was traveling to a remote location, he could access his sales leads from any PC. There were no capital expenses, with the exception of a PC and an Internet connection.

Initial Salesforce.com customers were small businesses that had no problem with a company managing its customer data. These customers were willing to take a risk in exchange for not having to buy hardware or hire staff. And because there was only a one-month commitment, they knew they could simply take their customer data and go home if it didn't work out for them.

Larger companies, on the other hand, were wary. What was this company? Was it financially viable; was the software any good? Would it be in business very long? If they liked the application, could they have the code and put it on their own server? SalesForce.com was able to convince at least some early customers that the company was well financed and safe to do business with. However, it refused to make its code available to companies to run on premise. Salesforce.com was able to break out of the small companies by selling directly to departments of large companies. These sales and marketing departments were able to put the fees to run Salesforce.com on their expense reports. Slowly but surely, Salesforce.com made inroads into large companies that appreciated the ability to avoid buying equipment.

What was behind Salesforce.com that customers did not see? Software, and plenty of it. The typical customer doesn't have to and doesn't want to understand the inner workings of Salesforce.com to use it. However, as you see later in this chapter, the underlying software that developed the offering has become the foundation of the company's partner ecosystem.

Like everything in cloud, there are overlaps between SaaS and the other areas of cloud computing. For example, Salesforce.com has a large partnering program for Platform as a Service called Force.com. (See Chapter 11 for more details on Platform as a Service). The foundation of Force.com as a development environment for partners is based on its own software development platform.

SalesForce.com ecosystem

Why are we telling you about what is inside the Salesforce.com software environment? We think it's important to understand that SaaS is a special instance of an enterprise application designed to support many different customers safely and securely with enough scalability to support changing situations.

In addition, this foundation then becomes the anchor for a rich partner ecosystem. Salesforce.com's partner ecosystem is called Force.com. It is a Platform as a Service (see Chapter 11) that allows complementary software companies to use this infrastructure and a set of tools developed by Salesforce.com to build on top of this CRM platform.

This isn't a new phenomenon. Companies have built partner ecosystems for decades. These leading vendors have encouraged independent software vendors to build their applications on top of their enabling software. Companies including IBM, HP, Microsoft, and VMware — to name a few — have used this approach to build success in the market. The difference with SaaS is that the ecosystems of partners are an essential part of the business model.

Today, Salesforce.com has revenues of more than $1 billion with a broad ecosystem of partners. Its brand is well regarded and large companies no longer ask the company to let them run the software in-house. A strong brand is essential to the success of SaaS and any cloud computing environment. But Salesforce.com isn't alone in the market. Companies such as Netsuite, Oracle, IBM, HP, Microsoft, Intuit, and hundreds of others have all entered the market.

But before we give you an idea of what types of products are out there, you should understand the economics of Software as a Service. While you will be reviewing the technical capabilities of solutions, you need to have a clear understanding of the economic implications.

Characterizing Software as a Service

What characteristics have to be in place for an SaaS to be commercially viable? Here's what we think is necessary:

- **The SaaS application needs to be generalized enough so that lots of customers will be interested in the service.** Here are some examples of these types of applications: accounting, collaboration, project management,

testing, analytics, content management, Internet marketing, risk management and of course, CRM. What doesn't work as SaaS? A specialized one-of-a-kind application with a small number of potential customers.

- **SaaS applications need sophisticated navigation and ease of use.** If an SaaS application isn't easy to use, customers will simply stop subscribing. Most SaaS vendors offer prospective customers a free trial for a month or so. If the customer doesn't start using the application during that first month, it's likely that the customer won't sign a contract. This is really important because it has been reported that less than 20 percent of users remain customers after the first month or so.
- **The SaaS application needs be modular and service oriented.** Without this modular approach, it will be hard to change and difficult to have third-party independent companies join the ecosystem.
- **An SaaS application needs to include measuring and monitoring so customers can be charged actual usage.**
- **An SaaS application must have a built-in billing service.**
- **SaaS applications need published interfaces and an ecosystem of partners who can expand the company's customer base and market reach.**
- **SaaS applications have to ensure that each customer's data and specialized configurations are separate and secure from other customers' data and configurations.**
- **SaaS applications need to provide sophisticated business process configurators for customers.** Each customer can change the process within the standardized SaaS application. For example, a company might want to add a process so a manager has to approve the price being offered to a new customer. A built-in configuration tool enables this to be done on an ad hoc basis without programming.
- **SaaS applications need to constantly provide fast releases of new features and new capabilities.** This must be done without impacting the customer's ability to continue business as usual.
- **SaaS applications have to protect the integrity of customer data.** That includes providing techniques for allowing data to migrate either to a private database inside the firewall or to a third-party storage capability.

What about the traditional on-premise software model?

The traditional way companies used software was to buy a *perpetual license,* because it doesn't end, and implement that software on their own systems internally. You pay once for the software and continue to pay a maintenance fee. This is quite different than the newer model of Software as a Service. A company offers to sell you a CRM capability. You decide on how many users will need the software and you pay on a per-month, per-user fee. The company takes care of all the maintenance of the software, the data center, the backup, and the support of the system.

Clearly, this varies from the perpetual license model of software acquisition. A perpetual license model means that the customer pays once for a license to the software. In the old days, you purchased a server, an operating system, a database license, and a license to the CRM system. You also probably needed some systems management and security software and needed to buy a backup drive and assorted other components. Every year, you paid a fee of between 10–25 percent of the purchase price of the software to get updates and software patches. Many companies still buy many products this way and we don't expect that to change any time soon. Some products are too specialized to be sold as SaaS anytime soon.

Understanding the Economics and the Ecosystem

The economics of the SaaS market are different than the traditional perpetual license software model. In the perpetual license model, the customer pays for the total cost of licensing the software and agrees to pay a per-year additional cost to cover maintenance and support. Maintenance can be as low as 10 percent or as high as 25 percent of the purchase price. One of the key differences with the SaaS model is that the economics are entirely different. The most important difference is that there is actually a lower barrier to entry when a company is trying to sell you a SaaS product.

Pretending you're a customer

Say you're a customer who's looking for a CRM product. If you decide that SaaS might be the way to go, you can shop around at various vendor Web sites, find a product that looks promising, and try it out for free for 30 days.

If at the end of that trial you decide that this product is really good, the company may decide that it is time to buy. Even though you might eventually want to have the product used by 50 people in the company, you might actually buy an entry-level configuration like a 5-user pack to get started. If the individuals in the company really like the product, you can add packages until you support all 50 users.

Determining the right revenue model costs

What does this mean in terms of the revenue model for vendors and how customers should think about weighing the costs between traditional perpetual licenses and SaaS-based license? Look at these numbers over a five-year period. It can be complex to work out all the details, but here is a general rule:

- Take the initial cost for the traditional software purchase.
- Add an annual fee of 20 percent for maintenance and support.
- Consider IT costs (including support services and hardware renewal, and so on. (For example, does your data center have enough room for the new CRM application? Will you need to add support staff or new management software?)

The other factor to consider is that the vendor might do everything it can to make you a customer. They might have some special incentives. For example, many SaaS vendors offer packaged deals. (An instance is if you decide to pay for a full year upfront, the price will be less; if you purchase large numbers of licenses, the costs will also be less.)

Calculating two examples

If you buy a traditional software product, it will cost you a one-time fee of $100,000. Now you have to add an annual fee of 20 percent for maintenance and support. If you look at the costs over five years, for example, you may determine the following: Software will cost $100,000;maintenance expenses will add another $100,000 over five years, for a total five-year cost of $200,000.

You have to consider all the related infrastructure costs. (Take a look at Chapter 21 for a full discussion on the economics of the cloud.) We can't begin to give you a sense of what that will cost you because every situation is different. For example, you might already have a sophisticated data center with excess capacity and sufficient staff to support an additional application. Or you might have to add everything from new hardware to networking to backup and support personnel. Do you charge each department based on their percentage usage of data center resources? Do you divide costs evenly between all departments as you would utilities such as electricity? No matter how your organization calculates expenses, that must be taken into account.

Many small- and medium-sized businesses lack or don't want the data centers that their larger counterparts have. Larger companies that can calculate the long-term impact of adding applications are also looking seriously at the SaaS cloud model.

If you go the SaaS route, here's what you're looking at: You determine that to support 50 users, it will cost you between $10 and $150 per user, per month. That figure includes support, general training, and data center services. Even if you take the high-end estimate of $150 per user, the cost of using the CRM SaaS application for those 50 users for 5 years will run about $37,500 — far less than the $200,000 cost of on-premise software, even when you add other costs (such as customization of business processes within the application and personnel training).

We can't give you an absolute figure; do your homework and compare all aspects of running software before you decide which approach is best for you. Prices can vary widely from an open-source version that offers support for a price to vendors that provide the software plus full integration services.

For example, you might look at an open-source CRM product. Although the basic product is free, you get no support or software upgrades, and must rely on finding patches and bug fixes from the community. If you're very technical, that might be a fine choice, but many customers want to pay for support to avoid a lot of headaches.

The value of the ecosystem

When SaaS vendors become well-established brands in the market, they attract an *ecosystem* (a set of partners that works directly with a key vendor, both in technical and go-to-market terms) that sees the value of linkage.

This is how it works: A SaaS vendor with thousands of paying customers opens up its programming interfaces to other independent software vendors. These vendors create software that sits on top of the infrastructure of the SaaS vendor. Therefore, they can get to market quickly because they only have to write their industry-specific code. They don't worry about messaging middleware, or business process services, or other complex programming. In addition, they can market their software to the SaaS vendor's happy customers (either through the SaaS vendor's portal or through the partner's direct sales force). This has become a standard model used by SaaS vendors to build their brand and power in the market.

If you're a customer who has licensed an SaaS application, you'll probably find another application that's built on the same infrastructure that easily integrates with what you already have.

Building an app on top of Salesforce.com

CODA is a software company that has been in the financial services packaged software market since the 1970s. The company had always partnered with on-premise software vendors such as HP, Digital Equipment Corporation, and IBM. In addition, the company liked to move to new platforms as they emerged (including the mainframe, the minicomputer, and client/server).

There came a time when CODA wanted to move quickly to take advantage of the movement to Software as a Service. Moving to a new platform was based on the ambitious plan to do for financial products what Salesforce.com has done for CRM. Needless to say, it was an ambitious goal. CODA management began to appreciate the potential for SaaS as a way to build customers faster than the sales process of on-premise software. Before deciding to use sForce (Salesforce.com's development platform), the company performed a return-on-investment analysis.

The challenge was the cost of writing the code from scratch internally. Basically, development management realized that they would have to write for a multi-tenancy environment that would have required several years of work to get the right infrastructure services in place. They simply couldn't justify the expense or the time required for development. Without worrying about any specific software infrastructure, CODA's developers focused on customer-facing features such as specialized processes for different industries.

Unlike some of the smaller companies that have built on top of sForce, CODA is a large company that serves mid-market companies. Salesforce.com needs CODA as much as CODA needs them. Salesforce.com needed to prove to the market that its platform could support a major application. CODA's application is happy with its relationship and is saving time and money. The test will be if customers adopt its new SaaS platform.

CODA wrote its application with Salesforce.com's Java-like language called APEX. Therefore, the company's locked into the Salesforce.com platform. From a go-to-market perspective, however, this is a plus because Salesforce.com will help CODA sell into its customer base.

Examining Types of SaaS Platforms

Because SaaS has been around longer than most other types of cloud computing, hundreds — if not thousands — of companies are trying to become leaders. It isn't easy. They face many obstacles. For example, it costs a lot of money initially to build the type of data center and the applications that can scale to support thousands of companies (and potentially millions of individual users). It takes time to turn a one-month free trial into a long-term contract. Despite these obstacles, some very successful SaaS companies exist, ranging from emerging players to the big IT companies.

We don't have the room to give you an exhaustive list of every company you might find, but we plan to give you a taste of what is out there. (In Chapter 23, we list resources that will help you identify even more players.)

It can be overwhelming when you look at how many companies have created SaaS versions of their products — even companies whoseprimary focus is the on-premise model feel compelled to offer customers a SaaS version of their offerings.

To help you make sense of this complicated world, we divide SaaS into three categories:

- **Packaged software:** This is the biggest area of the SaaS market. Packaged software comes in many different flavors: customer relationship management, supply chain management, financial management, and human resources, to name the most common. These integrated offers focus on a specific process, such as managing employees' benefits, salaries, and annual performance reviews. These products tend to have several characteristics in common: They're designed with specific business processes built in that customers can modify. They have moved in great numbers to the cloud because customers were finding the platforms too hard to manage.
- **Collaborative software:** This increasingly vibrant area of the market is driven by the ubiquitous availability of the Internet, combined with the fact that teams are located all over the world. This area is dominated by software that focuses on all sorts of collaborative efforts including Web conferencing, document collaboration, project planning, instant messaging, and even email. In a sense, it was inevitable that these platforms would move to the cloud: These tasks occur throughout the organization and need to be easily accessed from many locations.
- **Enabling and management tools:** We brought these two areas together because they support the development and the deployment of SaaS. What's in this category? Think about the development tools that developers need when creating and extending a SaaS platform; also think about the testing, monitoring, and measuring that a customer and the developer need. Also consider the compliance issues related to the use of this type of software in the real world. These issues are included in this third category.

In the next section, we give you a taste for the vendors in each of these categories, what they offer customers, and the issues you should consider. We can't possibly do this topic justice, but we give you a road map for how to understand the offerings and issues.

Packaged Software as a Service

We write a lot about how Salesforce.com created *customer relationship management (CRM)* as a service. It took a few years, but the company invested in its infrastructure, built a flexible and modular application, and made the navigation easier. But as with any successful venture, Salesforce.com competitors soon began entering the market in droves.

What companies are out in the market today that you should look at? It isn't as straightforward as it might sound. This is a dynamic market, so whatever company looks promising today could be gone tomorrow. On the other hand, the small emerging company that looks too new to consider could become a major force. Likewise, companies that have been successful as on-premise software providers are streaming into the SaaS market and could become viable competitors.

Companies in the packaged software market include the following:

- **Netsuite**, like Salesforce.com, offers a CRM foundation. Since its founding in 1998, Netsuite has added a number of modules for *enterprise resource planning (ERP)* application including financial capabilities, e-commerce, and business intelligence.
- **Intuit** provides a Financial Services Suite of products that support accounting services for small- and medium-sized businesses. The company provides a rich set of interfaces that enables partners to connect their services and applications into its environment.
- **RightNow** provides a CRM suite of products that includes marketing, sales, and various industry solutions.
- **Concur** focuses on employees spend management. It automates costs control via automated processes.
- **Taleo** focuses on talent management tasks.
- **SugarCRM** is a CRM platform built on an open-source platform. The company offers support for a fee.
- **Constant Contact** is a marketing automation platform that partners directly with Salesforce.com and other CRM platforms. They automate the process of sending emails and other marketing efforts.

Some of the traditional on-premise software companies have also moved into the packaged SaaS market, including

- Microsoft with its Dynamics package
- SAP with its By Design offering for the small- to medium-sized business market
- Oracle with its On Demand offering based on its acquisition of Siebel Software

Collaboration as a Service

Collaboration is one of the natural markets for SaaS. There's enough bandwidth and all companies are connecting to the Internet. In addition, more companies than ever have remote offices and workers across the globe. A team may be easily be spread across 100 locations in 40 different countries!

With the availability of SaaS-based collaboration services, things have changed dramatically. Although it hasn't yet surfaced as a major market, we expect that there will be companies that offer *unified communications* (an integration of telephony, instant messaging, and email) as a service. These offerings will come from the large telecommunications companies in partnerships with companies like HP and IBM. GoogleVoice could emerge as an important player in the future.

What companies are focused on collaboration as a service today? The following is a list to get you started:

- **MicrosoftLive** has made its first foray into collaboration as a service with its Meeting Live offering. Today Microsoft offers Meeting Live and live messaging services. In addition, Microsoft offers the ability to run its email server (Exchange as a Service). In the future, the company will have online versions of many of its collaborative applications.
- **LotusLive** is IBM's collaborative environment that includes a set of tools including social networking, instant messaging, and the ability to share files and conduct online meetings. IBM is publishing interfaces to allow other collaborative tools to be integrated into the platform.
- **GoogleApps** from Google, which has as many as 1.5 million businesses that use its various collaborative applications including e-mail, document management, and instant messaging. It publishes APIs so third-party software developers can integrate with the platform.
- **Cisco Webex Collaboration** platform comes from Cisco (which bought Webex in 2007) and it has become the centerpiece of its collaboration SaaS platform. It will probably use this platform to add unified communications as a service.
- **Zoho**, an open-source collaboration platform, includes email, document management, project management, and invoice management. It offers APIs to its environment and has begun to integrate its collaboration tools with other companies, such as Microsoft. Zoho offers support for a fee.
- **Citrix GotoMeeting** offers an online meeting service as part of its larger suite of virtualization products. See Chapter 17 for more about virtualization.

Enabling and management tools

How you use all sorts of software in your organization is changing dramatically — whether you're considering a supply chain as a service or a word processor as a service. As we discuss in Chapters 10 and 11, many companies are looking to service providers for needed functionality.

Underneath many of these environments is the open-source Eclipse framework. A set of enabling and management tools is being offered on a service basis. Although some of these services might actually be delivered within a private cloud in your own data center, many vendors will enable you to use their data center services. In this section, we talk about the enabling technologies that are being offered as services.

Over time, a lot more software and capabilities will be offered as a service, but we talk about five different areas in this section, including

- Testing as a service
- Monitoring and management as a service
- Development as a service
- Security as a service
- Compliance and governance as a service

Testing as a service

Testing is one of the biggest uses for cloud computing. Even when a company moves to using a public or private cloud, it still needs to conduct the same testing it would need in an on-premise data center , including

- Functional testing
- Unit testing
- Stress testing
- Compatibility testing
- Performance testing
- Requirements management
- Integration testing

One of the biggest problems for developers is accurately simulating the conditions (expected and unexpected) when software is deployed.

In addition, more companies are looking at testing as a service and development as a service as a way to keep track of development teams that are often distributed across the globe.

Having developers rely on SaaS-based services for testing can save tremendous amounts of time and money. When developers embark on testing, they often ask for hardware and software to get the task done. Typically, these organizations can't recoup the systems they hand over to developers. Many vendors produce testing as a service platforms, including HP, IBM, Sogeti (a United Kingdom–based IT services firm), Compuware, as well as smaller companies such as iTKO and SOASTA. We could actually name hundreds that are pouring into the testing-as-a-service space.

Monitoring and management as a service

Is what you see what you get? Maybe. That's why companies using SaaS need to do some of their own monitoring to determine if their service levels have been met by their SaaS providers. Even more complicated is when companies are using more than one SaaS application. And to complicate things even further, you must monitor not just a single application but also the *combination* of applications.

Companies in the systems management space are positioning themselves for this world. Vendors come at this market from two different perspectives:

- From the top down, large telecommunications are packaging their capabilities so they can help provide cloud management and monitoring.
- You also see traditional Web services monitoring companies offering services that will tell you if your Web site has added new services to support the cloud.

Development tooling as a service

Developers beginning to create new software are increasingly turning to development as a service. (In other words, development is done in a cloud-based environment instead of implementing development within a single internal-development environment.) This delivery model of development infrastructure can be done through one of the Platform as a Service vendors such as Google, Intuit, Microsoft, Force.com, and Bungee Labs. (See Chapter 11 for more on Platform as a Service.) Likewise, Infrastructure as a Service vendors such as Amazon.com offer support services for developers. (See Chapter 10 on Infrastructure as a Service.)

Security as a service

Almost without exception, vendors providing antivirus software are offering their products as a service. These vendors include Symantec, McAfee, CA, and Kapersky Labs. In addition, companies such as Hewlett-Packard and IBM have tools that scan environments for vulnerability scanning and testing.

Identity management is an important aspect of on premise as well as cloud services. Lots of companies in this market will begin offering identity management as a service.

Compliance and governance as a service

Compliance and governance tasks are time consuming and complicated tasks that large companies are required to do. Therefore, offering these capabilities as a service is critical.

Not surprisingly, hundreds of companies are moving into this market. Services that are becoming SaaS include the following:

- Patch management
- Business continuity planning
- Discovery of records and messages
- Various governance requirements such as SOX (Sarbanes-Oxley) in the United States and SaS 70 (Statement of Audit Standard) controls for data

For more on governance, see Chapter 16.