

SaaS to PaaS: Part 1—SaaS

What's the next evolutionary step to buying and selling software?

The ability to provide services to knowledge workers consistently, securely and reliably are the keys to application stickiness and the job security of information technologists. Until late, knowledge workers' applications were separated by firewalls and other organizational boundaries. However, with the advent and maturity of the Internet and its ability to be a services platform, the position of the applications that knowledge workers use has changed.

The application doesn't solely have to live behind an organization's firewall. In modern computing, they have migrated to the cloud, are incredibly robust in capability and provide a fully functional user experience, regardless of location. In an attempt to keep pace with these demands, service oriented architecture (SOA) and software-as-a-service (SaaS) technologies were developed and have now migrated to the center of business processes.

SOA was one of the first attempts to unite the strengths of best-of-breed applications. In short, it "allows different applications to exchange data with one another as they participate in business processes." The key technical enabler in this scenario is interoperability, while the key nontechnical enabler for this technical philosophy is business process, or, more specifically, a particular solution's alignment with it.

In other words, the brilliance of SOA is its ability to loosely couple disparate applications together for the purpose of streamlining a business activity. Using these precanned applications reduces time to market and the associated expenses of development. In addition, SOA has helped in reengineering the application lifecycle, a point we'll touch on later.

Marshalling behind SOA is SaaS. The fun really begins here. SaaS can be best thought of as the next evolutionary step of the service-oriented architecture model. In recent months, the pathway has been clear. 'Clear' to Platform-as-a-Service, however that was a set up. True nonetheless, but still a set up. Now let's see what SaaS really is.

On the heels of these benefits, IDC predicts the SaaS market will grow to be a US\$10.7 billion industry by 2009.

When SaaS was first introduced, it attempted to mitigate a number of issues that plagued businesses through the alignment of IT with business processes.

Simply, business processes lead the conversation because technology for the sake of technologists doesn't work. Moreover, it doesn't possess a broad marketable appeal.

Technology development for the sake of solving a business need is quite different. At a high level, the business needs and the associated challenges technology solves are often quite horizontal. Likewise, the reliance and types of solutions deployed to satisfy these business pains are similar if not identical. For instance, the methods and tools a Marketing, HR, or Finance department uses are fairly consistent across the industry even if the method of use is unique.

In summary, the business benefits of SaaS include:

- Ease of administration
- Fewer bugs
- Controlled versioning
- Faster deployment
- Broad accessibility

On the heels of these benefits, IDC predicts the SaaS market will grow to be a US\$10.7 billion industry by 2009. That said, the many CIOs and decision makers who welcome the software delivery mechanism do so with a grain of salt. For the most part, SaaS is often thought of as playing within one of two camps. It is either in a hosted-application management model, which is best thought of as hosting commercially available software or the other camp of software on-demand, which is where customers have access to a single copy of an application.

The best situations for SaaS, regardless of model, are for fairly vanilla, noncomplex business processes. These processes allow the organization to focus its resources on more complex and personal processes that have a lower market appeal. Per Modruson of Accenture, a core benefit here is the ability to deliver highly customized code through a standard but highly configurable code base. This further reduces costs and speeds and increases ROI as knowledge workers spend less time needing to come up to speed while lowering associated training costs per function.

At its core, SaaS has an incredibly high value proposition as outlined earlier by the abridged business benefits list. This begets the following questions:

- Where doesn't SaaS fit in?
- How has SaaS altered the development model of software?
- What is the next generation of SaaS?

The next generation of SaaS is PaaS, or, Platform-as-a-Service. This component will be discussed in next month's Connection.

> Where doesn't it fit in?

Applications that require a high degree of customization are typically not a good fit for SaaS computing. These applications include business intelligence, manufacturing systems and ERP solutions, to name a few. In general, applications that are at the core of a company's differentiation are typically too complex, too expensive and have too low of a degree of change to be good candidates for a SaaS platform.

A second issue relates to integration. We are in the age of interoperability combined with the need for information to be shared amongst multiple systems. This can be a challenge. The challenge might not be seen in the first generation of the offering, but because there are disparate systems within the mix, separate upgrade and patching paths for each can render the solution useless or at the very least a management nightmare. By no means has this problem been a deal breaker; nevertheless, it has become a point of concern lately. For example, successful vendors have turned

their attention to the data transport layer of the equation because of this issue.

> How has SaaS altered the development model of software?

So how does all this computing in the cloud change the software development model? It changes the model by altering the 'how.' We'll touch on how SaaS alters the 'where' in a later article.

> Prototype → Develop → Deploy → Host → Maintain

Within this lifecycle of development, the *where* and the *how* are not clearly visible. At this point, you simply need to understand that the major shift in design centers around developing where you'll host and deliver, opposed to developing in one environment then migrating and testing within another.

> What is the next generation of SaaS?

The next generation of SaaS is PaaS, or, Platform-as-a-Service. This component will be discussed in next month's Connection. Next month, we'll walk through not only the market relevance of PaaS, but also who the players are in this space and what it will mean to decision makers and CIOs.

Last, and most important, we'll put them together to produce a complete picture of Web 2.0, SOA, SaaS, PaaS and where together they all lead. **N**