Help Your Development Team Deliver

OneConnect

Despite their similar names, software as a service (SaaS), infrastructure as a service (IaaS), and platform as a service (PaaS) are not merely variations on the same theme.

SaaS, IaaS, and PaaS each consists of a collection of services designed to more flexibly and efficiently perform a variety of business tasks that traditionally were performed via capital assets. Due to the as-a-service nature, each method has the ability to provide much greater functionality and flexibility for what often is exponentially lower capital costs.

That is, for the most part, where the similarities end.

SaaS, IaaS, and PaaS are more different than they are similar to one other—though each compliments the other well. Of them all, PaaS is the most fundamentally different. To understand just how different PaaS is, look no further than the origins of each.

SaaS is a logical extension of familiar software licensing models. This extension was made possible by the increased speed and bandwidth of the modern Internet.

IaaS is more akin to the practice of leasing capital rather than purchasing it. Both cases involve a shift in perspective from external *ownership* of necessary business capital to external operation of necessary business capital.

PaaS, on the other hand, does not merely involve a shift of responsibility for certain assets—such as an outsourcing of capital. Instead, it represents a shift of paradigm in how a business handles all aspects of its IT and software development. The change is not just one of responsibility; it is a change in capability with regards to the development and deployment of software resources both internally and externally.

This whitepaper examines the benefits of PaaS for businesses as a whole and for the IT and software development teams specifically.

Defining PaaS

Before addressing the implications of PaaS, it is important to have a clear view of exactly what the term includes. As with the other as-a-service models, defining the boundaries can be somewhat problematic. PaaS is not a one-size-fits-all option; rather, it is a custom set of modular features that are uniquely combined on a case-by-case basis to fit the specific needs of a specific business at a specific time.

Generally provisioned as part of a cloud services package, PaaS is an abstraction of some combination of the following: infrastructure, operating system(s), configuration, and development. It can be set up in combination with IaaS or as a stand-alone product, though it is more effective when paired with IaaS.

With PaaS, software can be installed, run, developed, and implemented in a private cloud, a public one, or a hybrid combination of both. This streamlines the following processes:

- Provisioning
- Development
- Building
- Testing
- Staging

In fact, PaaS allows for a more seamless experience for the duration of the lifecycle management process. An application is pushed to the cloud, where it is analyzed and hosted in the appropriate runtime container based on what resources it requires. This simplifies or completely automates the process of scaling for changing access needs and traditionally manual processes such as load balancing and configuration tasks.

The added flexibility of a PaaS system includes the ability to run multiple instances of applications in cloud frameworks. Internal and outward-facing cloud systems are two examples. This simplifies compliance issues in some industries, and it can allow mashups of public and private data and resources on an ad-hoc basis.

Benefits to Business

The general benefits of PaaS to the business as a whole are similar to those offered by SaaS and IaaS: lower capital costs, increased flexibility, and improved responsiveness.

The biggest gains for most businesses are found in terms of scalability and resource provisioning. Because of the decreased reliance on physical infrastructure, as the business grows or as its needs change, adding capacity or reallocating resources can be accomplished more quickly and efficiently. In shifting to a PaaS environment, resource provisioning moves from a scale of weeks or even months to one of hours.

PaaS also makes good on the promise of the BYOD workplace. Because most or all of the actual processing is done in the cloud, the device itself becomes almost irrelevant. Web-based interface standards have been around for a long time and are generally well implemented across most consumer devices, so supporting a wide variety of personal devices in a PaaS system is a null question.

More subtly, PaaS widens the scope of cloud application usability across the entire business, and this makes a tremendous impact. Widespread adoption of cloud applications means more efficient use of the resources expended in developing those applications. The system-wide consistency and reliability provided by PaaS allows for faster time-to-market on new software packages, and on updates and patches to existing products. It also provides access to many cloud-only services that either are not available or possible in a locally hosted paradigm.

Benefits to Developers

With PaaS, the abstraction of components involved in actually running and maintaining software, particularly OS and infrastructure, automates many previously manual steps in the development process. This frees up development teams to focus on other important aspects of their business mission: developing software. In addition to providing a narrowed focus for IT and software development teams, PaaS also allows for the development of robust, flexible, and easy to implement standards with regards to operating systems, middleware, and configuration.

This standards-based environment, in turn, allows developers to code interoperability into new applications from the ground up and to vastly improve compatibility with existing apps. It also becomes easier to port existing, stand-alone applications and legacy apps to the cloud, because they are being ported to a known, predictable environment. From a software management perspective, a standards-based environment also means fewer issues with updates, patches, extension of functionality, and migration. Additionally, the flexible resource allocation that PaaS provides can greatly benefit the business as a whole. Where it really makes a difference is with the development team.

It is often difficult or impossible to accurately predict what resources a project will need. This leads to over-allocation of resources, because teams would rather have too much of everything than too little of anything. PaaS solves that problem by allowing for dynamic,

real-time resource allocation. Additional resources can be allocated as they are needed and then released to other projects again once the need has passed.

All of this adds up to faster and more efficient development. Development teams can get more services into the field more quickly. In turn, those services will be more useful and more responsive to changing end-user and business needs.

The faster time-to-market for developing new apps results in software developers being able to support shorter-term projects than previously would have been possible. This means, for example, that the development team could support specific marketing initiatives through the creation of custom software applications; this would have been impossible within a legacy framework.

Conclusion

At the end of the day, PaaS—much like SaaS, IaaS, or any other solution—is not a good fit for *every* business. A company with few needs beyond off-the-shelf software solutions, that is geographically centralized, or that doesn't anticipate significant expansion is probably better served by sticking with its legacy systems.

However, businesses with unique technology needs that operate in a dynamic, fluid market will find tangible benefits from choosing PaaS. Other businesses for which PaaS is well-suited include those that need their software applications to scale effortlessly with them as they grow and change. Both are best advised to look into what PaaS can do for them.