

Cloud versus Traditional Software – From a business software developer's perspective

Cloud versus Traditional Software –
From a business software developer's perspective

[Randy Wheeler](#)

Much has been written about the Cloud and the many technological advancements which have recently ensued. Lately, however, from a software development business perspective, programming in the cloud has reached a certain maturity. It seems, in talking to developers, that the software industry has become more focused on the end-user or customer experience and less on what is going to be the next 'hot' technical breakthrough. Their attention on understanding business needs versus priming the subsequent technical solution has been welcomed by users and their management alike.

Today, most papers, articles and blogs comparing Cloud-based solutions to traditional software concentrate on features such as lower costs, painless upgrades and a variety of other benefits. What's not been discussed, however, is the software development experience.

With something as revolutionary as Cloud computing, it is appropriate to compare and contrast that which software developers, designer/architects and others have experienced with Cloud versus traditional software. The transformation of the software industry from traditional enterprise solutions to the Cloud is one of these changes that deserve a review from that perspective.

Of course, as with all new technologies, the definitions are fluid and overburdened with acronyms. If you look up Cloud Computing in Wikipedia, you will find over a dozen variations: Infrastructure as a Service (IaaS), Software as a Service (SaaS), Platform as a Service (PaaS), etc. For most people, SaaS is the cloud computing layer that they best understand and are most likely to use. If you've run Salesforce, Google Apps or even Netflix from your home TV, you've used SaaS products. [SaaS](#) solutions normally utilize a multi-tenant architecture in which the application serves multiple businesses and users, while partitioning its data individually.

Since it sits in the middle of the cloud layers, [PaaS](#) is sometimes referred to as middleware. This PaaS layer contains virtual or cloud servers as well as operating systems, letting you develop,

deploy and manage multi-tenant applications. Let's dig a little deeper and discuss the differences between the multi-tenant PaaS applications versus "traditional" applications which have a software development "model" that is not multi-tenant.

Although applications created on the traditional platform might be deployed through the Internet with resource-sharing technologies that provide a more balanced use of resources, they do not have the benefits of true SaaS or PaaS solutions because the traditional model simply can't deliver the same benefits of [Cloud ERP](#), including lower costs, painless upgrades and a variety of other benefits. And, neither do the On-Premise/in-house, ASP and hosted suites which represent 'traditional' software.

Here's Why – Software Experts' Views

Let's review the experiences of [Lance Ford](#) and Sam Bhongale, who architected, designed and developed Insurance Applications for Valley Oak Systems/Aon. Their primary J2EE application for Claims and Policy Administrations was deployed in both On-Premise and ASP modes. Their current experience in PaaS solutions comes from deploying Rootstock Software's manufacturing application on two different Cloud platforms, NetSuite's SuiteCloud and Salesforce's Force.com. According to Ford, for a developer, the productivity benefits of the Cloud are significant and come with the platform, letting the developer focus on "meeting the business need, little else." Security, reliability/up-time, browser compatibility, platform performance/scaling and file system nuances become "nothing to worry about" when working on a PaaS platform.

Ford explains that both SuiteCloud and Force.com PaaS platforms have governance requirements that put controls on the resources the applications use, minimizing the impact on the platform. Contrarily, traditional software development platforms provide no governance. Governance requirements are not necessarily a negative, though.

"Governance forces the developer to be efficient and not be sloppy," says Ford. This has a downstream benefit of "fewer issues during Unit Testing, QA, and deployment." Although all software development should use a proven testing methodology prior to release and deployment, Ford prefers the Force.com requirements that require all code to be compiled and have at least 75 percent code coverage prior to deployment.

Another big difference is that the setup of a new customer is markedly different between Cloud applications and traditional solutions. For traditional software, this could involve capacity planning, hardware and software acquisition and configuration – all which must be done before actual deployment of the application. After that, a separate software installation needs to occur, in addition to the setup of the data repository/database.

Conversely, to add a customer to the Cloud, new deployment software is not required. The platform is already configured to have a new customer/tenant, utilizing the existing application software and data structure. The advantages to both the customer and software provider become staggering when one considers providing solutions to hundreds or thousands of different customers.

Caveat emptor! Many software development organizations have gone to an ASP or hosted environment to provide a solution that, to the customer, might appear to be a Cloud solution. These organizations want to say that their package is on the Cloud. Unfortunately, these approaches have separate software and databases (not multi-tenant) so, from a software provider standpoint, the majority of the deployment and ongoing maintenance issues and costs still exist and, therefore, get passed onto the customer. When hearing the word “Cloud” ascribed to the software application, the buyer must ask the question: “Is it multi tenant?”

One argument often heard against the Cloud platform is that customer specific requirements and customization can't be accommodated. That argument suggests that the customer requirements for unique fields, workflows, screens, triggering of special processes and other “personalization” is something that is not all that prevalent in the Cloud environment

According to Ford, this is not the case. Ford and other force.com developers know all too well that the customer specific needs of their client base have all been covered by the multi-tenant design. There are some platforms which do allow for what he calls “un-managed packages” and this is not conducive to multiple applications and customer personalization working together harmoniously.

“They let that customer have their own unique software deployment. However,” Ford adds, “we haven't run into a compelling reason to use unmanaged packages. The multi-tenant platform has the tools to customize the system to the customer requirements while staying true multi-tenant.”

For Bhongale, finding strong developers for the multi-tenant platform hasn't been a problem as the development environments look very similar. However, in traditional development, developers would need skills to work on both the low-level framework level as well as handling the business and applications layers.

“The skills required to develop at these technical levels are different, which underscores the difficulty of finding developers for traditional software,” Bhongale asserts. New developers easily move more quickly to the Cloud with a shorter ramp-up time as they don't need to work on the framework.

Software development management is also simplified on a multi-tenant platform. In the past, development priorities were balanced between the development of the infrastructure/framework and the actual application functionality. With deadlines and cost pressures, development often focuses on application functionality resulting in an incomplete, untested infrastructure that leads to lower quality and higher maintenance costs in the long run, impacting the company and the clients.

“By starting on a ready-built platform, the inherent managerial conflict when building new applications is eliminated,” says Bhongale.

A Manufacturing Expert's Perspective

From a manufacturing and QA perspective, Marty Browne shared his experience with multi-tenant platforms. Browne's experience in manufacturing software development began in the mid-70s with ASK software, where he spent 20 years in key roles leading the product development and the platform roadmap for the company's MANMAN product. Both during his time with ASK and later with various companies and consulting, he was always looking for a development environment that freed the developer to spend more time working on the business solution.

When asked about the current state of PaaS platforms, Browne had analyzed a number of Cloud ecosystems. Browne said that "Salesforce is the furthest along for building large, complex applications like Manufacturing." Browne also added that the current PaaS capabilities are just the "tip of the iceberg in terms of getting where PaaS can go."

"Efficient" is the word Browne uses when describing the effect of the PaaS on the people in the software organization: developers, QA, implementation, everyone. Efficiency as it relates to QA is "reduction of work as testing of the framework isn't needed. You only need to test your business application. There is simply less stuff you need to look at."

The upgrade process is also markedly different, especially as it relates to customizations. Upgrading on a PAAS platform is done wholesale, with client running one version one day and a new version the next. Sounds crazy, right? Risky?

Not really says Bhongale, given that the PaaS platform was designed to provide "push upgrades without the typical amount of work to manually manage customizations as in traditional software." The upgrade process, including client customizations was designed into the platform and not as an add-on with traditional software.

Browne spoke of the "manufacturing world view where multi-currency, multi-language, multi-time zone support are all critical but, because they are part of the platform, they don't need to be developed or tested, making everyone more efficient." Bhongale mentioned a popular feature of global text search that the platform does quickly and securely without development or QA. If developed on traditional software, it would be a sizeable task to search the whole system, efficiently and securely by role.

In the discussion with Ford, he finished up with probably the most revealing comment of all. Reflecting on developing solutions on the Cloud versus traditional platforms, Ford said "Cloud platforms are a far superior solution. I can't image developing software on a traditional platform. Multi-tenant Cloud platforms are the future and I wouldn't go back".

###

ABOUT THE AUTHOR: [Randy Wheeler](#) is the founder and CEO of Valley Oak Systems (acquired by Aon Corp in 2007) and has over twenty-five years of experience in the technology industry with the last thirteen years focused on the design, development, and implementation of Claims

and Risk Management Information Systems. Valley Oak Systems is the fastest growing national provider of Claims and Risk Management solutions to the insurance industry. Randy is the chief architect for the company's products and oversees development.

Prior to Valley Oak Systems, Randy was the Director of Sales and Client Services for Marathon Systems where he managed sales and technical professionals providing installation, training, custom report and interface development, release management, testing, and client [support services](#).

Before entering into the claims management systems area, Randy held multiple Regional Client Services Manager positions for Saddlebrook and GRiD Systems. While at GRiD Systems, Randy led a group supporting the President of the United States through the White House Communications Agency (WHCA). The WHCA used over 300 portable GRiD computers to provide all computer and communications services used by the Presidential staff during worldwide travel.

Randy is a recipient of the prestigious Ernst & Young Entrepreneur of the Year® 2004 award in the "Realizing Business Potential" category. He is also a member of the Rootstock Advisory Board.