WHITE PAPER

www.progress.com

COMING TO (YOUR) TERMS WITH PLATFORM-AS-A-SERVICE (PAAS)

BY MICHAEL BENEDICT

Platform-as-a-Service (PaaS), which provides a complete development platform and solution stack in the cloud, is gaining widespread acceptance as a way to deploy packaged software or build new applications without the overhead of a development environment. Enterprise adoption of PaaS is growing, as is PaaS use by ISVs who want to create client-facing software offerings without having to invest in platform infrastructure. PaaS does offer many advantages, including speed, cost-effectiveness and simplicity. As use of PaaS increases, a growing body of best practices has grown to guide IT departments, as well as the very important group of non-IT users, in the optimal adoption of PaaS. This paper explores some of the practices most relevant to achieving success with PaaS, focusing on ways to identify how PaaS can be best utilized within the specifics of a given organization's overall IT needs. Practices include understanding scalability and infrastructure aspects of PaaS deployment, the need for coding wisdom in otherwise "codeless" environments, integration with on-premise systems and other cloud-based IT assets, security and more.

PROGRESS

TABLE OF CONTENTS

INTRODUCTION

There's an old saying in business, mostly uttered by people who are nervous about change, which is that some proposed initiative will be like, "changing a tire while the car is moving." Unfortunately, every business change ever contemplated is like changing a tire while the car is moving. The "car" is always moving, or at least it should be. If it's not, the company is going under. New IT paradigms present this type of challenge, the newest one being the evolving cloud computing technology known as Platform-as-a-Service, or PaaS.

You may already know what PaaS is, but to summarize, PaaS makes a complete software platform—including infrastructure, application servers, development tools, databases and storage—available over the Internet. It's one of the revolutions arising from the IT field's broader move to cloud computing. While Infrastructure-as-a-Service (IaaS) makes the foundation of cloud-computing available to those who know how to build their own platforms, PaaS delivers complex and highly labor-intensive middleware technology patterns available "as a service." Typically, PaaS allows self-service and self-provisioning of resources to support cloud architectures. Through a browser, you can switch on a fully-configured cloud platform without having to worry about standing up hardware or installing software. PaaS scales elastically. There's a sort of everlasting gobstopper effect. Whatever platform resource you need, there's no practical limit to what you can get on demand, and pay for, as you require.

The real question is what can, or should, you do with PaaS? Answering that question about PaaS cracks open two big issues facing IT professionals today. For one thing, the definition of PaaS has changed frequently over the past year as companies develop PaaS solutions For instance, PaaS is blending with IaaS. It's also evolving into new flavors, such as integration PaaS (iPaaS), which offers a complete application integration capability in the cloud; business process management PaaS (bpmPaaS) which makes possible business process execution in the cloud; and application PaaS (aPaaS), which gives developers the ability to fast-track application development.

Then, there is the matter of users. Who will use PaaS in your organization, and for what purposes? Ready or not, PaaS opens up enterprise computing to a variety of new user groups. These include power-users outside the IT department who may want to create their own applications on the fly or even, as the IDC analyst Al Hilwa would put it, "Citizen Developers" who embrace PaaS to build programs you won't even know about.

These two issues are at the heart of this paper. The intent here is to help you come to terms—your terms—with PaaS in your organization. PaaS can serve your specific business goals. As an ever-changing, new-user-friendly technology, PaaS is the ultimate tire to be changed on a fast moving car. This paper takes a look at several recommended practices that will help you make the best use of PaaS in your organization.

THE CURRENT STATE OF PAAS

The IT world is well out of the starting blocks with PaaS. Technavio, the research firm, projects that the global PaaS market will top $6 billion by 2016. The growth rate is projected to be over 48% per year for the next four years.[1] Gartner valued the worldwide PaaS market at $1.2 billion in 2012, up from $900 million in 2011. Though this may seem small considering that the global market for all hosted services, including SaaS, was $109 billion in 2012.[2] Still, considering that it's a new technology, the size of the market and the projected growth rate speak to the potential of PaaS as a driver of business value.

THE TWO DOMINANT MODES OF PAAS TODAY

The constant evolution of PaaS notwithstanding, the consensus is that there are two dominant modes of PaaS now in use:

- Model-driven PaaS involves higher-level programming languages, or even template-based software building programs that enable users with little coding experience to create business applications.[3]
- Deployment PaaS is newer. It refers to platforms in the cloud that can host applications that were created with standard programming languages such as Java or PHP. The assumption is that the development is being done off of the PaaS solution on integrated development environments (IDEs) such as Eclipse.[4] This may sound a lot like IaaS, but it's different. With deployment PaaS, the developer does not have to worry about architecting, managing, or scaling the virtual machines that underlie the application.

[1] *Technavio, Global Platform-as-a-Service Market 2012-2016, April 3, 2013*

[2] *Nancy Gohring, "Platform as a service heats up," ComputerWorld, July 8, 2013*

[3] *Al Hilwa, "Analyst Watch: The Evolving State of PaaS," San Diego Times, October 1, 2013*

[4] *Al Hilwa, "Analyst Watch: The Evolving State of PaaS," San Diego Times, October 1, 2013*

BENEFITS OF PAAS

PaaS confers several business and IT management benefits. Though it's not a silver bullet, PaaS can be a huge source of advantage for certain types of in-house corporate projects and ISV commercial products. Organizations that write their own software can delegate infrastructure and platform management to specialists while concentrating on their core competency of developing applications. PaaS can facilitate cuts in IT overhead and capital expense by eliminating the conventional processes of acquiring and installing physical hardware, establishing or renting data center facilities, setting up servers, configuring, patching, maintaining, and so on. For early-stage companies, being able to operate on a lean basis can be an essential element of business viability.

CHALLENGES YOU FACE WITH PAAS

Like all new technologies, PaaS brings challenges along with its many benefits. Getting to PaaS success requires understanding what can get in the way of its fullest realization. Some PaaS challenges are organizational. Others are technical in nature. None are fatal.

PaaS challenges do not exist in a vacuum. They exist in your specific organizational setting. Depending on your potential PaaS users and choice of PaaS technology, you will face different versions of the major PaaS challenges. Figure 1 provides a perspective.

While there is obviously a continuum of users' technical sophistication and platform automation, the four groupings in the figure can be instructive. For instance, Line-of-Business (LOB) users may find themselves using either a model-based PaaS or a deployment PaaS, depending on which technology is available in the organization. LOB users can take advantage of each platform choice in different ways. With model-based PaaS, the LOB users might be able to build and launch their own application without much IT department involvement. With deployment PaaS they will need professional developers to build an app for them.

What is your PaaS use case, or use cases? The way that you envision PaaS being used will inform how you deal with PaaS challenges. It's possible that you don't know all of your potential PaaS use cases, and that's okay. In fact, the adaptability of PaaS is one of its strengths. There will be uses that people in your organization will invent for PaaS that might not have occurred to you. The trick is to think through how PaaS might fit with your existing IT landscape and be prepared to handle potentially difficult PaaS use cases as they arise.

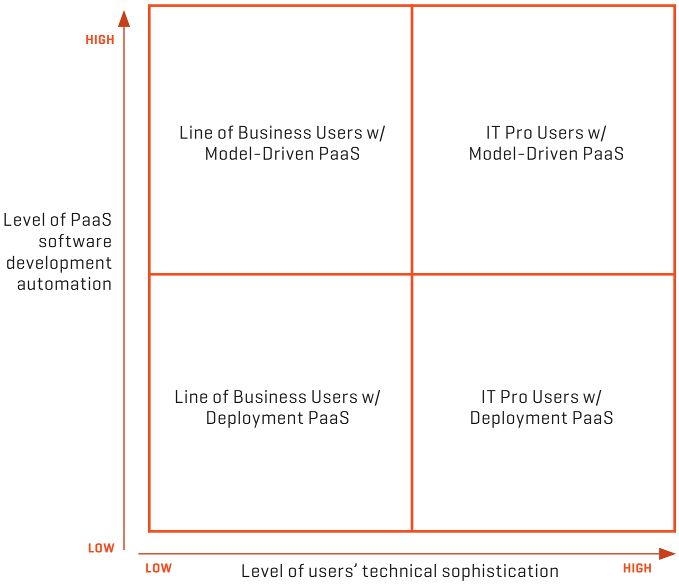

HIGH
Level of PaaS software development automation
LOW
Line of Business Users w/ Model-Driven PaaS
IT Pro Users w/ Model-Driven PaaS
Line of Business Users w/ Deployment PaaS
IT Pro Users w/ Deployment PaaS
LOW
Level of users' technical sophistication
HIGH

Figure 1
PaaS usage scenarios based on levels of user sophistication and software automation.

SCALE AND AVAILABILITY

Availability should be addressed early with PaaS, especially if you've built an application that can go viral. The "as-a-service" trend is part of the broader cloud computing and mobile device revolutions. You have to think through the implications of the new world of IT as you adopt PaaS. Application load levels can be difficult to predict, both for your application and others that are hosted at the PaaS provider. With social media and mobile apps, your PaaS creation could be hit with requests from a huge crowd of new users overnight. That could be an amazing thing or it could cause outages and all kinds of disruptions. You must ensure that your PaaS solution can scale easily in response to demand. If your PaaS application is integrated with back-end systems behind the firewall, an ungoverned runaway success could disrupt your business. The problem doesn't even have to be with your application. "Bad Neighbor Syndrome" occurs with some PaaS providers. Bad neighbors are other PaaS customers who share cloud infrastructure with your application. Even though all PaaS tenants are isolated from each other, tenants who consume more than their fair share of IaaS resources can cause your application to perform poorly.

RAPID RELEASE CYCLES

Compared to traditional software development, the release cycle for PaaS is like playing a movie on fast forward. This is especially true for model-based PaaS but it also applies to deployment PaaS. In both situations, code can be pushed into production more quickly than in the conventional developer-to-operations cycle. In traditional software development, a new feature request involves multiple stakeholders in a clumsy, partially manual review process. A series of handoffs ensues as the feature goes from request to development, test, security review, and installation. The process can take a long time. The new feature would be developed and tested, then put into production in an overall application update. The whole process might last several months or at very minimum, a couple of weeks.

Model-based PaaS can make the creation of a new feature possible within a couple of hours. Generally speaking, this a good thing. Business ideas can be quickly translated into working software. Agile methodology and new streamlining techniques such as "continuous integration" of new code help provide a disciplined framework for aligning business and IT stakeholders with PaaS' rapid development tooling. Yet, these methods cannot create this alignment on their own. The organization has to be ready and willing to change its approach to software development. The reality is that most organizations will have software projects moving at different speeds. As PaaS becomes one of the accepted modes of development, there has to be a parallel organizational commitment to keeping up with the new speed levels in the processes of requirements gathering, development, review, test, approval and release.

SECURITY AND COMPLIANCE

Concerns about cloud security are not new, but PaaS can bring risk exposure to a whole new level. The cloud tends to blur the security perimeter in general. In the old days, you had a good idea of where your infrastructure ended and the rest of the universe began. With PaaS, your business extends to multi-tenant servers in vague geographical areas. The ability for non-technologists to set up software on their own, as might be the case with LOB users on model-based PaaS, can wreak havoc on controls over data access, authentication and authorization. If the PaaS application connects to other enterprise systems, it can become an inadvertent conduit for improper access and potential mischief.

Responsibility and accountability are major factors to consider when making the move to PaaS, especially if there is a "shadow IT" effort in the works. Business stakeholders tend to complain about all the "red tape" and slow-moving processes in IT, but those processes exist for a reason. Anyone who has ever been asked to find an email subpoenaed in a lawsuit will know the value of rigorous controls and data governance. A shadow IT project hastily thrown together with PaaS can accidentally create a security and compliance minefield.

INTEGRATION WITH OTHER SYSTEMS

Connections between cloud-based applications built on PaaS and other enterprise systems present security, operational and governance challenges. PaaS-based software is inherently service-oriented. It has the ability to call on application programming interfaces (APIs) exposed on numerous systems. These include APIs that use Simple Object Access Protocol (SOAP) as well as the increasingly popular Representational State Transfer (REST).

Without adequate controls, systems can be exposed through APIs along with the business processes they support. Of course, few organizations simply leave an API totally open to the world. However, the difference in development and change cycles between legacy systems and PaaS software can lead to a "Tortoise and Hare" syndrome where the legacy system cannot keep up with new PaaS features. If external users can access internal business process through APIs that are out of sync, that can cause operational and compliance difficulties. Alternatively, if an API is not available because a change in the PaaS solution has broken the connection, that is also bad for business.

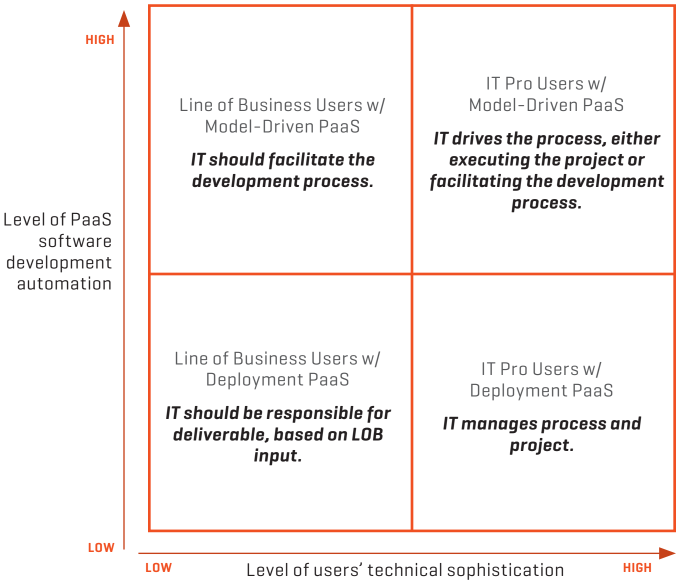
ACHIEVING SUCCESS WITH PAAS

Best practices for PaaS are new and evolving. There is not yet a single, mature set of practices for the technology. That being said, it is possible to frame your IT department's approach to PaaS in the context of both the targeted user groups and choice of platform. Figure 2 revisits the usage grid from Figure 1 and adds some possible approaches to managing software projects on PaaS with optimal outcomes.

The main point of Figure 2 is that there is no such thing as PaaS in a monolithic sense. There is always PaaS in a specific organizational and technology context—your context. Although PaaS is a technology, the best practices for PaaS need to be developed from the perspectives of both business and technology.

Yes, we've all heard this before. IT is about business. Software development on PaaS is potentially so fast and easy, however, that is requires a rethinking of the business-IT relationship. Success with PaaS will flow from a PaaS culture that unites business, IT managers and developers.

The following suggested practices, therefore, blend business and IT agendas. Each can be adjusted to the scenario depicted in Figure 2 that best represents your specific situation.


HIGH
Level of PaaS software development automation
LOW
LOW
Level of users' technical sophistication
HIGH

Line of Business Users w/ Model-Driven PaaS

IT should facilitate the development process.

IT Pro Users w/ Model-Driven PaaS

IT drives the process, either executing the project or facilitating the development process.

Line of Business Users w/ Deployment PaaS

IT should be responsible for deliverable, based on LOB input.

IT Pro Users w/ Deployment PaaS

IT manages process and project.

Figure 2

Possible approaches to managing software development with different user and PaaS scenarios

SETTING UP PRACTICES AND PROCEDURES FOR RAPID SOFTWARE RELEASES

To make the most of the development speed inherent in PaaS, there has to be cohesion between PaaS software development practices, procedures, and the tools that are employed to manage the process and create the software itself. Fundamentally, this is a people issue. The more quickly software stakeholders can collaborate and communicate about the software they are developing—or updating—the more effective the PaaS solution will be. The way this is done will vary depending on the interplay between users and platform, as suggested in Figure 2:

- When LOB users use model-driven PaaS, they can theoretically create applications all by themselves. But they shouldn't. IT should facilitate the development process to ensure that the end product adhere to corporate standards and doesn't create data management or security problems.
- When the organization has deployment PaaS, LOB stakeholders need to have input on the application but IT should be responsible for development.
- IT pros using model-driven PaaS have a choice of executing the project on behalf of LOB stakeholders or facilitating the process.
- If IT pros are using deployment PaaS, it's an IT-centric process that will resemble a conventional software development project, just one that deploys on PaaS.

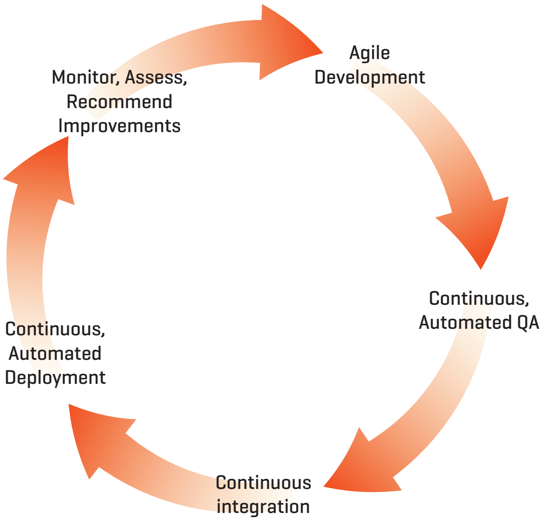

Agile Development
Continuous, Automated QA
Continuous integration
Continuous, Automated Deployment
Monitor, Assess, Recommend Improvements

Figure 3

The DevOps cycle, which involves continuous iterations of agile software development, review, integration and deployment.

One solution to the speed of PaaS that is gaining traction is known as DevOps, which is the merging of software development and IT operations. Using specialized management tooling, DevOps can effect a process of continuous development and integration of new code into production. As depicted in Figure 3, the DevOps work cycle assumes never-ending iterations of new software development.

It is possible to do DevOps without PaaS but the two concepts were made for each other. The DevOps pattern, which can be implemented in varying levels of intensity with different tooling options, provides a framework for uniting the varied stakeholders who must collaborate on PaaS. Figure 3 shows how PaaS and DevOps can fit together to form a fast-moving, continuous software development process that brings the speed and agility advantages of PaaS to life. Stakeholders are continuously notified of software development activities that affect them. People involved in various stages of the application lifecycle have a view of what others are doing. There are fewer blind handoffs.

PLANNING FOR SCALABILITY

PaaS introduces a number of new elements to the scale planning process. The ability for systems to be connected with relative ease in the cloud makes for great agility, but it disrupts the conventional notion of scalability. When you factor in the potential for exponential scaling that can arise with viral app adoption, it's an altogether different conversation. The best practice is to think through the ramifications of increased demand at the outset, rather than waiting until scale becomes an issue. Modeling future load may actually affect the selection of PaaS tools, because not all PaaS offerings handle scale the same way.

Scale challenges with PaaS can be sorted into ordinary and extraordinary scenarios. In the ordinary case, it is necessary to plan for orderly expansion of an application as it expands from pilot to adoption by a single business division, then the entire organization, and so forth. An application might have 500 users at launch, be released to 5,000 users according to plan after one year, and then to 20,000 users after the second year. Each planned growth period is met with a well-planned increase in server compute capacity, storage, network, database instances, and so forth.

IT managers who are planning the ordinary, orderly scaling of a PaaS application should adopt a practice of modeling their future usage predictions with distinctive PaaS characteristics, including:

- User seat license costs
- Cost and performance impact of increase PaaS solution usage:
  - Server units (VMs)
  - Databases
  - Storage
  - Network
- Cost and impact of downtime

These factors are universal for any scale planning. Indeed, they should look quite familiar to anyone who has had to model the way that growth of application usage will affect IT resources. What's different with PaaS is that increase in demand can produce some unpredictable results. Take them one at a time: Seat licenses can be a big deal with PaaS. If you are using a solution with a high annual fee per user, you could get walloped with licensing costs if you want to grow your user base. A good planning practice is to research your PaaS provider's pricing policies and see if they offer an enterprise option or special contract terms for high numbers of users.

Server unit expansion, as well as database, storage and network can affect both cost and performance. Not all PaaS solutions scale out well. Those that offer more IaaS-type services can be better in this regard. There are two things that can go wrong. The amount of additional VM, database, storage and network capacity you provision may not be linear. You might have to over-provision to ensure the same level of service that you have at smaller scale. This is a potential side effect of multi-tenancy. In addition, increasing the scale of the application can result in more downtime. This difficulty is related to the unpredictable nature of multi-tenant applications. Server failures will happen in any environment. The difference with PaaS is that you need to know how outages will be remediated. Assessing the server failover process is an essential practice to engage in when planning for scale.

Extraordinary scaling scenarios require a little more planning. The first step is to think though the potential causes of an extraordinary scaling situation. That is, a scenario where demand for an application might rise by an order of magnitude with little warning. For instance, with the relative speed and ease of connecting applications to one another in the cloud, an organization might want to enter into partnerships that require systemic integration. There is nothing new about this, but the cloud means that the integration could be pushed on a fast track that outpaces planning for PaaS scaling. The best practice is to plan for "when" such alliances will lead to big increase in scale, not "if."

Seasonal spikes are another driver of extraordinary scale. A web retailer, for example, might experience a 10-20X jump in site traffic on "Cyber Monday," the huge online holiday shopping day that comes right after Thanksgiving. This can be anticipated, so the best practice is to simulate the load on the specific PaaS solution you are using. The failover architecture should also be tested prior to the big day. This kind of planning and testing is standard for on-premise systems as well as IaaS-based solutions. Application PaaS should be subject to the same requirements. Indeed, as IaaS and PaaS merge in certain solution offerings, capacity management is a standard feature. However, it is not a good practice to assume that the PaaS solution will scale properly during a seasonal spike just because a comparable IaaS-based solution functions well at scale. Some PaaS solutions are built on top of Amazon Web Services (AWS) IaaS. There is no reason to doubt that AWS scales well, but it's necessary to test the specific PaaS software that's running on top of it because of the idiosyncrasies of PaaS. For example, a seasonal spike might cause an increase in data IO calls that overloads the database servers as they are deployed in PaaS. That database performance has to be measured in specific terms in order for the failover to be calibrated properly.

The worst extraordinary scale event is one that comes without warning. Massive bursts in demand can occur as a product becomes a hot seller or a story attracts global attention. The proliferation of mobile devices compounds the problem through sheer force of numbers. And, this is definitely an enterprise computing issue even if it seems to be somewhat consumer-oriented in nature. For one thing, corporate-developed, customer-facing apps are growing in popularity. It might be a high-quality problem to have your business app downloaded by a 1,000,000 people, but it's still a problem if you can't handle the traffic. Then, there are dependencies. Your PaaS application might connect to a few harmless partners, who in turn expose it to processes in their own mobile apps—which can go viral. The resulting traffic can throw you into a scaling crisis.

The best practices for handling extraordinary scale events involve defining service levels and throttling access if certain thresholds of demand are reached. Practices should include tooling for access limits that are measured in terms of known PaaS capabilities, i.e., if you know that your PaaS solution can handle 10,000 simultaneous users in a spike, that is your access ceiling. You should not assume that it can handle 20,000 users because a comparable on-premise system can manage that level of load.

UNDERSTANDING YOUR PLATFORM'S LIMITATIONS

Every application development platform has its limitations. If you own your application platform and manage it on-premise, you can do whatever you want to it, making adjustments to virtually any aspect of it as conditions change. This is the case with deployment PaaS. With model-based PaaS, your ability to exert control over the platform can be limited.

It's best to think through future use cases and compare them to the PaaS solution's capabilities. You might plan to create a small-scale internal application with a well-defined feature set and user base. For this type of use case, a simple, model-based PaaS solution that is low cost, but affords very little control, might be a good solution. However, given the rapid release cycle potentially available in PaaS, especially when teamed with DevOps, you might start with one incarnation of an application today and be in possession of something quite different a year later. A good practice is to map out where you might need flexibility in deployment and levels of support and control, including the following important factors:

- **Automation levels.** Most PaaS solutions enable self-service provisioning of platform elements. However, the ease of use for the automated self-service options, and the actual range of processes that are automated, can vary widely.
- **Customization.** Does the PaaS solution enable you to customize your stack, assuming you want to? For example, if you want to use Postgre SQL instead of Microsoft SQL Server, can you make the change? And, if you can make the change, can you do it on a self-service basis? If making a change like that requires a special support request, which must be overseen by a person on your end, you may be increasing the IT overhead required to manage PaaS.
- **Platform templating.** Some model-based PaaS solutions make it possible to set up a reusable application platform template. For example, your application needs a specific application server, database, and storage array, each with numerous custom settings. Building it from scratch over and over might not be the best use of your IT resources' time. Again, this may or may not be relevant, but if you are contemplating setting up multiple instances of a PaaS application, it could be a driver of efficiency.
- **Hands on service, vs. "self-help."** Support quality levels differ between different PaaS providers. Some are completely self-help oriented, or nearly so, requiring you to consult a knowledge database to solve problems. Or, they require written tickets that get answered within some preset period of time. This may or may not be suitable for your needs. Other PaaS providers make specialized, dedicated support staff available to you. Of course, this latter option is going to be more costly, but might be worth the expense if your application is critical to your business.
- **Public vs. private cloud.** Some PaaS providers force you to use a public, multi-tenant cloud infrastructure. This might be fine. It might be a big problem. The challenge is that it can be hard to predict where you need to be in a year or two, so the best practice is to work with a provider that at least gives you the option to change from public to private cloud at a later date. At issue are performance, security and scalability. A private cloud can mean one of two things. It can mean being able to run your PaaS software in a dedicated VM that is not sharing infrastructure with other clients of the PaaS vendor. It can also mean having the ability to place your PaaS wherever you want it, including your own on-premise data center. This may seem puzzling. If it's cloud-based, how can it be run on-premise? The answer is that the cloud and PaaS refer to software architectures, not location, per se. You can set up a cloud architecture in a private datacenter. The only difference will be the "finite" nature of your private data center, versus the "infinite" capacity of the cloud provider. There are hybrid alternatives as well, such as "bare metal" hosting providers that will manage your cloud-based application completely on dedicated hardware that is provisioned exclusively to you.
- **Portability.** Vendor lock-in is a risk facing developers who work with PaaS. It is a good practice to figure out in advance if you will be able to move your PaaS application

onto another PaaS solution or on-premise infrastructure if necessary. Making this kind of move is a great option to have. In some cases, it is possible to migrate an application off of PaaS through a replication process, while not moving the actual application itself. For example, Progress® Pacific™ gives you the ability to replicate the data architecture and application logic of software developed on Force.com and run it on Pacific.

ALIGNING THE PAAS APPLICATION WITH SECURITY POLICY

Cloud computing can still raise the hackles of information security specialists. As the industry evolves, though, security concerns about cloud-based software and data are shifting from shrill panic to practical embrace. It is essential to work through security issues in a PaaS application carefully. The first step needs to be engaging with the information security people in your organization. If you are looking at PaaS from the business unit, non-IT level, it may be tempting to go "shadow IT" all the way and keep it secret from InfoSec. This is not a recommended practice.

The security conversation needs to start at an appropriate level of risk assessment. That is, if you're planning to use PaaS to plan the company softball league, you probably don't need a major security audit and exhaustive risk mitigation plan. If your PaaS application contains confidential information or touches financial transactions, in contrast, you will want to do a pretty thorough risk and business impact assessment. Based on the amount of risk your PaaS application is exposing the organization to, security best practices suggest that you work through the following issues at a suitable level of seriousness:

- **Who owns the code and the data?** The answer to this question may at once be obvious and far from it. Your contract with the PaaS provider likely states it clearly. However, a good practice is to go get a legal review of your PaaS agreement to ensure that your organization will be protected if there is a legal conflict with the provider, a bankruptcy or other types of legal incidents, such as third party subpoenas. There are also theoretical-versus-practical aspects to this issue. For instance, you may own the PaaS code, but if you're locked into the platform and cannot get off it, that code ownership will be meaningless.
- **Segregation of Development/Test/Production environments.** In the fast-moving cycle of PaaS software delivery, it might be possible to overlook, or at least blur, the classic dev/test/run separations. For a critical application, the best practice is to enforce the strict separation of environments no matter how inconvenient it might be.
- **Segregation of duties.** This is relevant to financial systems where compliance is an issue. Most transactional systems require that certain financial actions be made by more than one person. For example, it should not be possible for a single user to generate a purchase order, approve the purchase and authorize the writing of a check. That is a formula for fraud. However, if you're taking a shadow IT approach and doing PaaS "off the books," so to speak, you might accidentally create just that kind of problem for yourself.
- **Threat protection.** Most PaaS providers have robust defenses against viruses, malware, breach of confidential information, Denial of Service attacks and so forth. That shouldn't be an issue. If it is, you're working with the wrong provider. What's significant, though, is how they handle security incidents and how an incident might affect your application. A best practice is to undertake due diligence on the PaaS provider's internal failover architecture, recovery time objectives (RTOs) and disaster recovery architecture and procedures.

Security is a massive topic that cannot be completely covered in a paper such as this. The best practices takeaways are:

- Security is a high-priority with PaaS
- It's a matter that involves both personnel and policy
- Some PaaS security risks can be mitigated with software tools

For example, application-level security concerns can be addressed with a new generation of API security platforms. Under-secured APIs can expose a PaaS application to risk. API security tools can take care of authentication and authorization of PaaS application users. They can throttle API access and enforce service-level agreements. They can manage the OAuth protocol to authorize access to data by third and fourth parties.

FOLLOWING CORPORATE IT'S LEAD ON PAAS INTEGRATION

PaaS needs to be part of the IT department's purview, even if it's a model-based PaaS solution with its primary use driven by a business unit. While it may be tempting to take the "shadow IT" route, there are many good reasons to treat PaaS just like any other corporate software development platform managed by IT. If the PaaS application is going to connect with systems at third parties, it is imperative that the IT department be involved. Now, you might think, of course, if there's integration with other systems, there has to be an IT project. That's not as true today as you might imagine. With Web APIs and cloud computing, it is easy to picture a scenario where a third party, such as a distributor or customer, simply asks for your API information and integrates right into your PaaS application without anyone in IT even knowing about it. This is a very unwise practice.

Connecting a PaaS application to external systems exposes it, as well as any internal system it is integrated with, to many of the risks discussed in this paper. Security, performance, reliability, availability, and scalability are all at risk if the integration is not implemented correctly. The IT department will know how to handle the integration process. Integration should be treated as just another development cycle, managed by IT but subject to review and input by all participants in the development process. The pace may be slower than business stakeholders want it to be, but if the PaaS/DevOps model is in use, the integration can move a lot faster than a conventional application integration project.

CONCLUSION

PaaS technologies have the potential to accelerate software development while reinventing how IT supports the development process. Best practices for PaaS are evolving, but three major themes should emerge from this brief review: balance, people, and alignment. With PaaS, it's tempting to take shortcuts and release applications without considering important security and performance factors. There has to be a balance between the need for speed and sensible planning and controls. Many software tools are available to coordinate and control the PaaS development process. People are a big part of making PaaS work for you on your terms. PaaS success is an organizational issue because the technology throws stakeholders together in rapid development cycles. However, the technology itself can only do so much to bring about the kind of collaboration that will make PaaS an effective mode of software development. Best practices should pay attention to the human side of the equation. Finally, PaaS practices will deliver best results when they foster alignment between multiple agendas in the IT-business relationship. Strategic objectives have to align with PaaS development practices. Operational plans have to align with security and infrastructure policies, and so forth. Balance, people focus and alignment are the key ingredients to getting the results that you want from PaaS.

ABOUT THE AUTHOR

Michael Benedict is the vice president and general manager of the Progress® Pacific™ cloud platform and Progress® OpenEdge® product line and is responsible for leading the company's platform-as-a-service (PaaS) and application development businesses. He has previously held other senior leadership roles at Progress in product management, operations, strategy and sales.

Prior to joining Progress, Benedict served as director of product management and marketing for online products at Engage Technologies, where he helped grow its market capitalization to more than $3 billion and launched the first cloud-to-on-premise advertising campaign distribution network, Engage AudienceNet. Benedict was also a co-founder and operating manager of NV Company. Benedict received his Masters in Business Administration from the Kenan-Flagler Business School at the University of North Carolina and holds a Bachelor of Science in Civil Engineering from the State University of New York.

PROGRESS SOFTWARE

Progress Software Corporation (NASDAQ: PRGS) is a global software company that simplifies the development, deployment and management of business applications on-premise or in the cloud, on any platform or device, to any data source, with enhanced performance, minimal IT complexity and low total cost of ownership.

WORLDWIDE HEADQUARTERS

Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA Tel: +1 781 280-4000 Fax: +1 781 280-4095 On the Web at: www.progress.com

Find us on facebook.com/progresssw twitter.com/progresssw youtube.com/progresssw

For regional international office locations and contact information, please go to www.progress.com/worldwide

Rev. 11/13 | 131114-0067