LEVERAGING THE CLOUD FOR DEVELOPMENT & TESTING ENVIRONMENTS

CONTENTS

BUSINESS PROBLEM

The development and testing process is full of challenges for enterprises and federal agencies of all sizes. Traditional dev/test environments are underfunded, under resourced, extending development intervals and creating uneven transitions to production. How can we quickly deliver new or modified applications to our internal and external customers while satisfying the desired business requirements for a lower cost than traditional development environments?

The reality of today's economic environment has resulted in thinly staffed IT departments, while the demand for quick and thorough application development continues to increase. This drives the need to seriously evaluate ways to improve the development and testing lifecycle process. To improve this process, we must address the challenges we face in the dev/test environment. These include committing a significant capital outlay for development environments, developing and testing an application that truly meets end user requirements, and delivering an application quickly and efficiently while dealing with compressed delivery timelines. In this fast-moving cloud and mobile environment where development speed is everything, DevOps professionals need partners who can keep up.

The adoption of cloud technologies can help address roadblocks to satisfy functionality and human and technology resource allocation, while lowering costs.

While development and testing environments have changed little in the past twenty years, the adoption of cloud technologies for this purpose can help address the traditional roadblocks to successfully satisfy functionality and human and technology resource allocation, while lowering costs. It's rare that a combination of concepts, tools, and resources is offered to help developers overcome these challenges. Taking a holistic approach in addressing these challenges can result in significant improvements in the testing and development lifecycle process.

OVERCOMING COST OBSTACLES

Development projects often prove to be both a costly and risky endeavor, with the primary challenge being the capital outlay to mimic a production environment for dev/test efforts. Due to the current economic environment, many—if not most—development organizations, along with the businesses they represent, face the need to manage budgets even more tightly. This often means poor project execution and missed deadlines in the best cases, and in the worst cases, escalations in commitment of additional resources as a cure-all for runaway projects.[1]

To overcome the cost obstacle, many developers look to alternative deployment methods (for example, buy versus build, or SaaS versus buy). In other cases, companies are deferring a portion of new development or are making tough choices to cut services that the business might otherwise need.[2]

The process proves even harder from smaller organizations that may not have the extra IT support to do the development. For some enterprises, devoting IT personnel to application development can be a costly venture. In fact, some have devoted up to a quarter of their IT budget to application development. Often times the remainder of that budget is spent on keeping an existing infrastructure up and running. In the end, devoting a quarter of that to development is truly a significant cost.[3]

Deploying more servers and network equipment was once considered a solution to solving resource issues, but resulted in soaring costs and inefficiencies. This also held true in the traditional dev/test lifecycle that required devoting dedicated resources such has hardware, software and precious development and engineering time and effort. This leads to the inefficient use of resources and driving up costs, both Cap-ex and Op-ex.

Several factors in the traditional dev/test environment continually drive costs up:

- No sharing of resources across projects
- Constantly running, driving up power costs while utilization is low
- No self-service model and changes
- Configuration changes are time and manually intensive

1. Hillegersberg, Jos -Information Systems Management; Winter2008, Vol. 25 Issue 1, p57-70, 14p, 2 charts, 9 diagrams, 1 graph
2. Gartner: Key Issues for Application Development, 2009 20 April 2009 ID:G00166694

Developers are constantly being asked to respond to change with both flexibility and speed, which often becomes nothing more than a vague set of goals and shifting priorities. Whether you are launching a new application, or enhancing, modifying, or fixing a bug in an existing application, comprehensive testing is required prior to pushing to production to maintain proper levels of service for your customers. This often requires creating multiple environments, with each involving configuration and resource allocation that can sometimes take weeks or even months. Add testing and remediation to the equation, and you now stretch delivery timelines to months and even years. Cloud environments enable dynamic provisioning for multiple environments while utilizing a shared pool of resources. This increases the efficient utilization of resources—both technology and human—freeing up those resources to be applied to other projects or corporate objectives.

SCALABILITY

Before the adoption of the Internet and related technologies, scalability was an easier issue to manage. During that time, the computing environment was essentially a closed system as data access requirements and usage patterns of staff, customers, or business partners was predictable, with little fluctuation. Strategic planners had reviewed historical data on which to base their projections for scaling the computing environment to match customer demand and performance requirements. During this time, the application development life cycle typically spanned a year or many years. Every company's IT environment is different, but at the end of the day, it just needs to work.

Once Internet technologies were embraced, corporations viewed this new technology as an ideal, low-cost method for sharing information throughout the organization with their partners and their customers. Not only was it inexpensive, but it also made it very easy for users to do their own development and share information quickly. The number of consumers of corporate information grew exponentially and created scalability issues, as the massive amounts of data traffic resulted in system failure.

Cloud technologies provide the opportunity for developers to dynamically scale application environments to match workloads in both production and testing

environments. Virtualization enables the automation of the build and configuration process removing the hours, days, and weeks it can take to bring up an environment. This allows developers the luxury of focusing their efforts on thoroughly planning and executing on designs that can handle traffic loads, while ultimately meeting the end user's requirements and expectations for availability and performance.

PROCESS AND CONTROL

The development process is often rushed, resulting in a poorly controlled development and testing process that leads to a chaotic effort and incomplete delivery. This lack of control results in delivery date slippages, application bugs and errors, and often pushing already approved and committed features to subsequent releases.

Leveraging dispersed development teams can introduce problems in coordinating the efforts of dispersed in-house and outsourced development teams either within the home country of an organization or around the world. Dispersed development teams often use different development tools and are confronted with conflicting resource allocations, priorities, and shifting deadlines typically required for implementation and training, making each of these essential tasks challenging in the end.[3]

A well-defined and rigorous dev/test process saves time and effort. It can also help bridge communication gaps and coordinate team efforts. As such, the process needs to address:

- Assistance and oversight in coordinating and managing resource allocations—both hardware and human
- Removing administrative and bureaucratic steps in the process
- Truly duplicating production environments
- Clearly defined remediation paths
- Launch process and rollback contingency planning

Tools that provide a centralized platform to manage, update, and share this information reduces cycles spent on communicating and updating team members who can then focus on the development and testing tasks at hand. Carpathia's experts custom design solutions your way to reduce complexity and free up IT to focus on more strategic projects."

3. Challenges in Application Development http://www.allthingscrm.com/application-development/challenges-in-application-development.html

PARTNERING TO ENSURE SUCCESS

Enterprises and federal agencies are constantly looking to improve their IT processes. They are striving to improve the quality of applications delivered and decreasing time to market while reducing both hard and soft costs.

Partnering with a trusted cloud operator like Carpathia can help you meet the most stringent security requirements to ensure company data is kept safe and secure.

Carpathia serves the most mission-critical needs of commercial companies such as Altiscale, Deltek, Egnyte, Exostar, EMCOR, GILT, InMobi, Tibco, Spotify, Symantec and Viggle. We deliver agile and flexible solutions that help companies scale based on need so they can significantly reduce waste and cost.

CONCLUSION

Developing and testing applications is a lengthy, expensive, and difficult process to manage. By virtualizing these environments, it allows IT teams to address many of the challenges encountered in the traditional dev/test environment. Development teams are typically early adopters of new technologies and adopting cloud technologies for development and testing environments. As such, IT departments can quickly realize the benefits of cloud technologies.

Migrating the upstream efforts into the cloud will help these teams satisfy stakeholder requirements by quickly deploying features that improve end users performance experience or functionally through corporate finance and budgeting goals.