

2008

PACIFIC NW SOFTWARE
QUALITY
CONFERENCE



COLLABORATIVE
QUALITY

OCTOBER 13-15, 2008

CONFERENCE PAPER EXCERPT
FROM THE

CONFERENCE
PROCEEDINGS

Permission to copy, without fee, all or part of this material, except copyrighted material as noted, is granted provided that the copies are not made or distributed for commercial use.

Virtual Lab Automation: Best Practices and Common Pitfalls

Abstract

Virtualization is a ground-breaking technology that promises quantifiable benefits for application development and QA organizations: faster lab deployment, less manual set-up work, greater resource flexibility and utilization, and easier reproduction of defects.

However, adopting virtualization in a development or QA organization isn't without issues. Often it's not obvious whether to build out a custom virtualization framework or make a strategic bet to implement a full virtual lab management solution, complete with automation and a pool of centralized hardware.

This paper discusses the software quality challenges commonly faced by application development teams and how virtual lab automation can lead to a more strategic approach to QA practices. It describes the best practices for virtual lab automation adoption and also highlights the common pitfalls organizations face during implementation. Finally this paper outlines the steps to evaluate a virtualization solution for your QA organization and provides further resources to help you get started.

Ian Knox
iank@skytap.com
Skytap, Inc.

Ian is currently Director of Product Management for Skytap and is responsible for product management, market positioning, demand generation and go-to-market strategy. Ian joined Skytap from Microsoft Corporation where he was group product manager for Microsoft Visual Studio. In this role, Ian led the product management team responsible for building a new \$400M enterprise tools business in the Application Lifecycle Management (ALM) category. Prior to Microsoft, Ian was a Principal Consultant at PricewaterhouseCoopers where for 7 years he worked on global software delivery projects for Fortune 500 clients.

Virtualization Forces a Rethink of QA Practices

A manager in charge of application QA needs to determine the best way to apply limited resources to achieve the goal of delivering a reliable, performant application. Given the nature of modern distributed environments, this job is becoming more and more difficult:

- **Increasing complexity.** Building a lab environment to support testing is a painstaking task for a typical distributed architecture. Implementing test environments that mirror production as closely as possible require that machine, network and application settings are carefully configured to ensure environmental issues are found before deploying to production.
- **Resource constraints.** Budgets are shrinking and procuring physical hardware, storage and network resources for test environments is costly and often hard to justify given low utilization. Applications often require testing on multiple operating system versions and language variants, browsers and devices, so achieving the optimum balance between adequate test coverage and acceptable risk is difficult to achieve.
- **Productivity bottlenecks.** The set-up and tear-down of labs is usually a time-consuming, manual process. This IT provisioning overhead is costly and reduces the time QA teams can spend testing an application.
- **Faster cycle times.** The broad adoption of agile development practices have put pressure on QA teams to reduce test cycles and work iteratively to deliver software.
- **Communication issues.** Developers often spend an inordinate amount of time trying to reproduce and debug defects reported by the QA team. If an application state is difficult to reproduce, it can mean hours of wasted time diagnosing an issue.
- **Globally distributed teams.** The growing trend of using offshore testing resources compounds the problem of sharing consistent environments across teams and facilitating effective team collaboration.

Given the rise of virtualization, application development managers and QA professionals are rethinking tooling, practices and skills to help solve these ongoing challenges. Many have already experimented with virtualization in their lab environment and are now realizing a virtual lab automation solution is necessary to overcome new challenges that a virtualized environment brings.

Virtualization Challenges

Many QA teams are using virtualization by building a proprietary framework using scripting, such as VMWare images with Perl scripts. This can provide many benefits including the ability to snapshot and restore images, faster machine deployment times and better utilization of hardware.

This approach has proved successful in many organizations, especially with ad-hoc or simple test frameworks. However, as many teams are learning, it can soon become a significant effort to develop and maintain scripts and a library of images. Additionally, it's not easy to deploy and manage multi-machine configurations in an isolated network without implementing virtual private networking. Usually there is no user interface to manage the test lab which limits the control users have over the lab environment. The cost of administration can quickly warrant implementing a more robust solution and many organizations are investigating virtual lab automation to solve the overhead costs of a custom solution.

The remainder of this paper suggests five best practices for adopting a virtual lab automation solution and some common pitfalls organizations should avoid.

Best Practice #1: Understanding Virtual Lab Automation (VLA) Capabilities and Limitations

Virtual Lab Automation (VLA) is the industry term that has been coined to describe a new breed of tools and test practices to automate labs using virtualization technology. A VLA solution can include some or all of the following capabilities:

Resource Pooling and Provisioning

Resource pooling enables processing power, storage and networking infrastructure to be shared between different teams and individuals, increasing utilization and availability of resources, and reducing costs. In conjunction with resource pooling, an orchestration and provisioning process allocates and releases resources as needed.

Multi-Machine Configurations

Virtual machine images are the containers that enable operating systems and applications to be isolated from physical resources. A group of virtual images that define a complete system, including network and storage characteristics, is defined as a configuration. For instance, a configuration could consist of multiple Windows Vista client machines, an Oracle database server and a WebSphere application server. Configurations can be easily created by combining virtual machines through a user interface. A configuration is a very useful concept for QA teams because it allows a whole system to be defined and isolated in a test environment. Virtual networking

enables copies of the same environment to be run in parallel and the emulation of production environments during the test process.

Configuration Library

A configuration library allows a team to manage and organize virtual images and configurations. Standard builds and images can be created and made available to development and QA teams to save hours of set-up time and environment configuration. Additionally, the library is used to store new configurations that are cloned or created as part of a test.

Suspend, Snapshot and Restore

The ability to suspend a complete state of a multi-machine configuration and make a snapshot (a copy at a point in time) is a major benefit of virtualization. This is especially useful for application development teams because when a bug is found a configuration snapshot can be taken at the point of failure and a link to the configuration added to a defect report. Instead of a developer spending hours to reproduce the defect, he or she can restore the configuration and start debugging the issue within minutes.

Scheduling and Reservations

Many in-house virtual lab implementations have a fixed pool of resources for teams to share. Scheduling and reservation functionality allows the resolution of resource conflicts and test environments to be reserved ahead of time.

Reporting and Monitoring

Reporting modules allow users and administrators to manage usage and quotas and determine whether the system resources are being used optimally. Monitoring enables the system health to be diagnosed, including CPU utilization, storage performance and network usage.

Automation API

Automating a test lab almost always involves integrating tools and test processes. An automation API enables teams to automatically create test environments as part of the build process and initiate automated test runs once a new build has been deployed. An automation API is typically made available through a web services interface.

Administration and Security

Administration and security features often include user and quota management, project creation, permissions and authentication. Remote access to the system (for instance for an outsourced vendor) is usually managed through secure connections via encrypted protocols and virtual private networking.

These capabilities are undoubtedly attractive to the vast majority of QA teams. However, every team is different and determining whether some or all of these capabilities are beneficial is important to consider. Any QA manager who has implemented functional testing tools knows that there are some testing projects where the effort required to implement an automated testing framework can far outweigh the benefits. The same is true for virtual lab automation.

Common Pitfall: Misunderstanding test types that are suitable for a virtual lab

Almost all VLA solutions utilize standard infrastructure and support a wide range of test scenarios. Hardware and network characteristics can be easily configured through the web interface to specify number of processors, amount of memory and network settings of machines in a configuration.

Test Types	Virtual Lab Support
Unit Testing	●
Functional Testing	●
System Testing	●
Integration Testing	●
Unit Testing	●
Performance Testing	○
Regression Testing	●
Localization Testing	●
End-User Acceptance Testing	●
Appliance / Hardware-Specific Testing	
Key: ○ Partial capability ● Full capability	

Fig 1. Test Types Supported

Typical customer test scenarios include unit testing, functional testing, system testing, integration testing and load testing of applications. However, there are a few use cases where a VLA solution is not recommended. These include tests which require specific hardware access (e.g. bios driver tests) and some types of performance and stress testing (e.g. a test of application performance on a specific hardware profile).

Best Practice #2: Determine the right implementation approach for your organization

There are essentially two approaches for adopting a virtual lab automation solution, an on-site package and a hosted virtual lab service, both with different advantages and disadvantages

There are a number of vendors now offering packages for Virtual Lab Automation, including VMware. Many of these solutions offer most or all of the capabilities discussed above. They have typically been adopted by large enterprise organizations where the expense, time and organizational changes required to build a centralized lab are worth the effort. These organizations have seen dramatic Return on Investment (ROI) from these new labs. Voke, an analyst firm, estimates a virtual lab automation solution can deliver a 25-50% reduction in hardware needs and an average time saving of 3 days to deploy a lab environment.

For QA organizations that are not part of a large enterprise organization, there are some considerations that may lead them to a hosted alternative (such as the solution offered by Skytap). First, implementing an automated test lab requires a large upfront investment in lab hardware and software. It also requires significant effort to implement, configure and train lab personnel. This type of expense is especially hard to justify for dynamic projects and if departmental QA budgets are under pressure.

Second, implementing an in-house virtual lab solution requires skilled IT resources to be assigned for administration and virtual image population and maintenance. Unless a lab reaches a critical mass to cover multiple development and test organizations, this administration overhead can be prohibitive.

Finally, even though an automated virtual lab improves resource utilization there are still going to be resource conflicts between teams unless an expensive pool of infrastructure is procured that covers peak demand. This means some test groups will still need to wait for resources, reducing their effectiveness and increasing delivery risk.

An alternative to an in-house virtual lab package is using a hosted lab or 'Virtual Lab-as-a-Service'. In the same way Salesforce.com offers a CRM package as a service, a virtual lab as a service solves many of the issues and risks associated with an in-house implementation. For example, there are no upfront investment costs and infrastructure can be scaled up and down according to testing needs. Furthermore, the administrative costs associated with running the lab are handled by the service provider. A hosted lab can be easily integrated back to on-site assets using a Virtual Private Networking (VPN) connection and typically a customer will only pay for the hourly usage of the lab machines when in use, eliminating the expense of test machines sitting idle.

Common Pitfall: Ignoring Indirect Costs

When determining the Total Cost of Ownership (TCO) for each approach, it's important not to forget to include the costs for internal IT support, as well as the more obvious

hardware and software capital expenditures. Often, much of the ongoing costs for a lab involves IT administration. Also, indirect costs should be included such as time savings for development and test team members gained by using the capabilities of a VLA solution.

Best Practice #3: Automate Your Test Lab Operations

One of the major benefits of a VLA solution is the ability to integrate with a build process and testing framework to enable an automated workflow. Typically, this workflow is enabled through scripting or tooling support in the build server and/or automated testing tools. Almost all VLA solutions (both in-house and hosted) offer an API to enable integration.

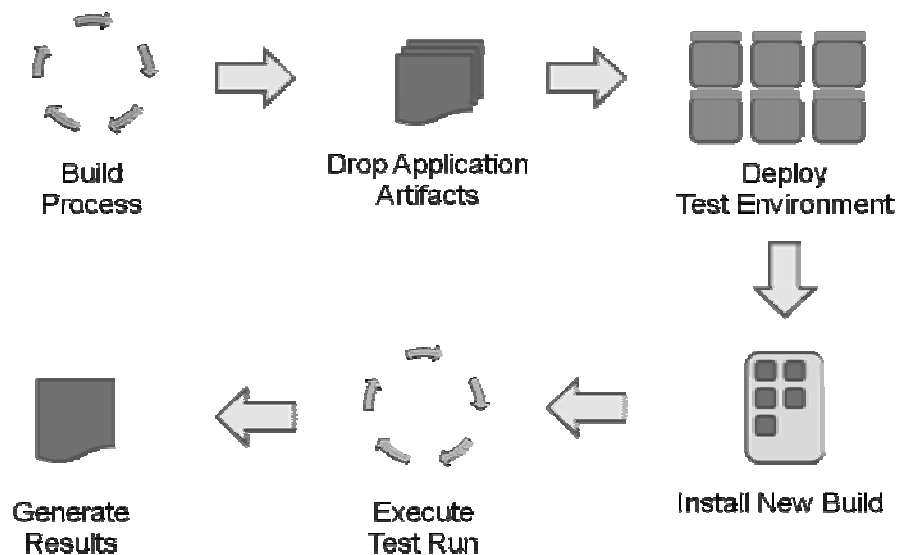


Figure 2, Typical Lab Automation Process

Investing in populating an asset and configuration library will enable standard environments to be deployed quickly. Many environment set-up tasks can be automated to avoid manual user intervention. For instance, as part of a nightly build, new virtual machines configurations can be automatically deployed and software builds and patches installed in preparation for a test run. In addition, using the virtual private networking functionality found in a typical VLA solution, a test environment can be deployed to mimic a production environment as part of the automation process.

Best Practice #4: Enable Team Collaboration with User Permissions and Projects

Every VLA solution offers the ability to specify user access levels and permissions. Typically an administrator will have access to the entire lab, team leads have the ability to create new projects and environments and individual testers work as part of a project and have access to only the resources they need.

Once user access control has been specified, administrators and team leads can enable or restrict access to resources through the virtual lab user interface. This is especially useful for projects where outsourced vendors are utilized. A project can be created with the environments to be tested and outsourced testing professionals can be given access only to the resources required for a given test run. If an organization is using a hosted lab, it's easy to ensure the test environment is isolated from the corporate network.

Now environments are stored in a configuration library, replication of defects becomes much easier as both development and test teams are sharing the same environments on the same virtual infrastructure.

Common Pitfall: Not Restricting Access to Master Configuration Images

There will be a set of master (or 'Gold') configuration images that are commonly used to create environments. These include standard desktop images, application builds and server images. It's important to ensure these images can only be accessed by a lab administrator and not accidentally modified by a developer or tester.

Best Practice #5: Obtain Team Buy-In With High Impact, Low Effort Changes

Once implemented, training your staff about the use of a VLA solution is the first step to encourage adoption. However, demonstrating how it will help make their jobs easier is equally important. By choosing a few high impact areas to focus on and securing some quick wins to improve productivity, your team is much more likely to adopt the new solution.

One of the most obvious areas to focus on is populating the configuration library. If you have chosen a hosted VLA solution, this will be pre-populated and you will only need to update base virtual machine images to match corporate images. This will enable IT operations and QA leads to very quickly assemble and deploy new environments, dramatically cutting down the 'time-to-test' and simplifying environment configuration.

Another high impact, low effort change is to ensure snapshots of virtual machine configurations are captured for new defects. This will enable testers to more easily communicate defects (some have called it a 'screenshot on steroids') and developers to isolate problems in a fraction of the time it previously took.

Common Pitfall: Inadequate Training of Outsourced Test Teams

Organizations will typically roll-out training programs to their staff, but may forget to do this for outsourced test teams, especially when new service providers are added after a VLA implementation. Conducting short training sessions using a Webinar solution will ensure outsourced teams are equally productive with a VLA solution.

Summary

As part of evaluating approaches for adopting virtualization in your development and testing organization, we recommend the following steps:

1. Determine the Capabilities Your Organization Needs

Build a requirements matrix and determine the types of testing and usage patterns typically seen in your organization. Consider operating platforms and hypervisor vendors used in your environment as well as the current skills of your IT operations and QA team.

2. Explore the Potential Solutions

Evaluate an in-house package and using a hosted virtual lab service.

3. Evaluate Total Cost of Ownership (TCO)

Build a Total Cost of Ownership model. Be sure to include software, hardware, implementation, and administration costs. Also try to estimate the indirect costs associated with each option.

4. Conduct a Trial Project

Conduct a proof-of-concept or trial project using your short-list of solutions.

5. Implement Solution and Refine Your QA Process

Once you've tested and implemented your solution, we recommend evaluating your current test practices, updating these to reflect the new virtual lab capabilities and investing in training your team before rolling out the solution broadly.