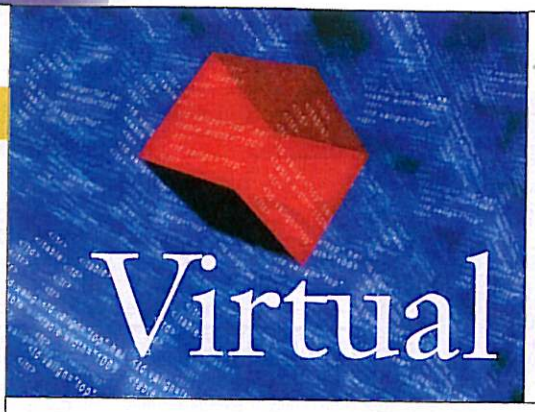
IN Focus

This month, Julian Brook discusses the virtualisation impact for the world of software testing and quality assurance.
Gary Keogh argues that we are moving slowly in the right direction on broadband provision.
Ian Master explains how to multiply the uses of your business continuity infrastructure, and Eoin Blacklock argues for constant online backup.

VALUE Points

- **Quality:** Virtualisation addresses many of the challenges encountered in providing and managing development and test environments
- **Broadband:** While the government consultation document was a good step, there is much to do to develop applications and understanding of broadband in the Irish market
- **BC:** The ability to dynamically manage applications on servers, without interruption to users, exceeds most organisations' business continuity requirements and maximises uptime
- **Back up:** Without a backup plan in place there is a risk of losing company data that cannot be retrieved. Online systems provide comprehensive protection and fast recovery



JULIAN BROOK examines virtualisation and the impact on testing and software quality

Virtual quality

There seems to be no escaping the "virtual" buzzword at the moment, where everything that was once real now appears to be virtual; software running on virtual machines with virtual memory and virtual storage is used by virtual teams in virtual meeting rooms on virtual LANs. This is largely due to the current large-scale enterprise uptake of virtual infrastructure as a result of its proven benefits, perhaps aided and abetted by the proliferation of articles on the subject.

This trend towards virtual infrastructure has important benefits, implications and opportunities for our IT projects, strategy, quality assurance and ultimately our business competitiveness. The benefits of reduced hardware, power, cooling and administration costs, the associated green credentials and increased flexibility and continuity have been discussed widely. Disadvantages have also been widely presented including increased server sprawl (exactly what we were trying to reduce), security implications and new performance and management challenges.

In this article we look at the substantial strategic opportunity presented by leveraging virtual infrastructure to improve the delivery and management of environments for software development and testing.

Infrastructure

Virtual infrastructure falls into three broad categories:

Server or platform virtualisation: Basically, this enables one physical server to appear as many servers. Ultimately, it has resulted in a paradigm shift from considering servers as a physical commodity to a logical commodity or service. This is the main type of virtualisation that is commonly referred to today. Common implementations include: Microsoft Hyper-V and Virtual Server, VMWare Server, ESX Server and Citrix Xen amongst many others.

Storage virtualisation: This enables the presentation of a single logical abstraction of a pooled set of underlying physical storage. This single view of data enables enterprise wide data de-duplication, backup and restore and management. Examples include: IBM System Storage SAN NetApp, EMC Invista, Veritas Storage Foundation and NetApp Data OnTap.

Network virtualisation: The combination or division of one or more local networks into a virtual network to provide a single logical view of a related set of resources and functionality. This can be done across separate physical resources across a wide corporate network or by grouping together a set of resources within a virtualised server\platform.

Environment to software quality

At its most abstract, testing comprises defining a set of inputs that will result in a set of expected outputs according to the requirements and specifications of the system under test. In addition to the inputs and expected outputs, testing requires an environment containing the system under test to process the inputs into output. Without this environment testing cannot take place. (See fig. 1)

The environment comprises a number of elements that define the state of the system under test and enable testing to take place. This state comprises:

Application version(s): This includes versions and patch levels across all application layers from operating system to application frameworks and business applications.

Application(s) configuration: This defines how the applications communicate and operate and can significantly affect the behaviour under test.

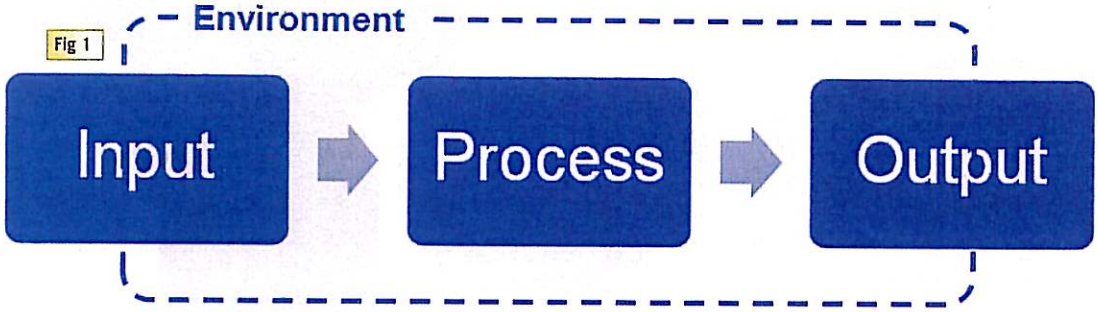
Data: The data utilised by the system under test also affects the validity of the testing. If we are considering performance testing do we have a representative volume? Does the data volume have the data variance and distribution we require? Do we have sufficient data to exercise the functionality we wish to test?

Infrastructure: The network, processor, peripherals, storage and their interfaces can all affect the validity of testing. For example, do we require a 64-bit processor or 32-bit processor to ensure our compatibility testing is valid? (See fig. 3)

It is critical that this environment has a defined state that aligns with the test objectives. This environment definition then becomes another test asset alongside test documentation, test data and test scripts. Without this well-defined environment, testing is unrepeatable and unable to be audited.

Environment management is concerned with the service of delivery of environments in a defined state to support the teams and activities across the software development lifecycle.

Fig 1



Environment challenges

Many IT projects have encountered environment related issues and defects that impact the test and development teams and are the cause of significant delays to projects. It has been estimated that up to 40% of effort during the software development life-cycle is wasted due to these types of issues. The table below outlines these challenges and how virtualisation can reduce the impact of these. (See fig. 2 - table.)

Server virtualisation "snapshots" provide a full audit capability by being able to re-instantiate the exact test environment used during testing. This is important to

Time to provision and configure new test environments: The lead time to obtain new hardware, deploy into the data centre and then install and configure the necessary applications and data is so long that it affects the organisation's ability to respond to change and the time to market for change. For a not insignificant number of clients, this can be in the region of months.	Although this activity is still required, server virtualisation technical resources can configure a template "image" of a server and\or application. With storage virtualisation template datasets can be created. Any time a new instance of a server or application configuration is required, all that is required is for a new copy of the image to be created; basically a file copy operation and a new instance of the configuration is available within hours and not days or weeks. Similarly different datasets can be cloned and attached to virtual server configurations. Combining multiple server and application images and datasets through network virtualisation allows an entire complex multi-system architecture to be encapsulated as a single independent and isolated object. Through a combination of these techniques, a library of ready to use servers, applications and datasets can be built and made ready to deploy. Provisioning can be done in a matter of minutes.
Unauthorised change made to environments: Code and configuration changes are made in response to defects or other issues which are not tracked. This typically leads to defects and issues affecting production at go-live as the necessary changes are not tracked and so not applied into production.	Virtualisation does not help prevent unauthorised change to be made to environments. However, as templates can be created we can revert to known states and verify testing against these whenever we require. Additionally, we can ensure that a "snapshot" of the final state of environments used during testing is retained. This can be used to check if there are differences between the test environment and production environment if defects are found in production.
Insufficient environments or infrastructure available: Typically a compromise is established and environments are shared between testers and even across multiple projects. This results in testers affecting the validity of each other's testing by altering the state of data (unless is it is very carefully managed and coordinated - a challenge in itself).	Server virtualisation allows more servers to be offered using the same amount of physical infrastructure. By having more servers available we can provide more environments. We then start to reduce the number of issues and complications that sharing environments brings – reducing scheduling complexities, project interdependencies and the risk that testers can impact one another.
Insufficient resources available to support non-production environments: Typically technical support resources for environments, such as system and database administrators, are shared between production and non-production system support. As a result when production issues arise, non-production support activities are put on hold affecting the productivity of the entire software development life cycle.	By the creation of template "images", although issues can still occur, these can be addressed in the standard image and users of environments can update their images to obtain these "fixes" according to their own environment usage schedule. Over time this means that support resources only need to support single instances of applications in a central location.
Scheduling of test environments: The coordination and allocation of environments becomes a complex and time-consuming role. This is usually compounded when sharing of environments as dependencies become more complex to manage. For example an issue affecting one project may prevent testing continuing until a code fix is applied. However, the code fix cannot be applied as it would affect the testing for another project.	By having more environments available, the scheduling of environments tends to become simpler. Additionally, server virtualisation allows us to provision "image" templates so quickly and also ,gives the ability to "pause" an environment. We can thus avoid the scenario where environments sit unused for periods of time because the process of tearing down and provisioning another environment takes longer than the window of time for which the environment was not being utilised
Reproducing defects can be difficult and time consuming: In order for defects to be resolved, a developer generally must be able to observe them. Recreating the issue in a different environment is sometimes problematic and results in defects being unresolved. This is compounded when ineffective configuration management is in place.	Server virtualisation allows "snapshots" to be made of the environment state at the point of defect detection. These snapshots can then be transferred to developers who can redeploy the snapshot and investigate the defect in an exact copy of the environment in which the defect was found.
Integration with change control, release management and configuration management teams: Environment management teams are responsible for delivering environments to its customers. However, this delivery must be part of an integrated release management approach and strategy. These processes are the foundations and support that define the input for the environment requirements.	Server virtualisation can assist by ensuring that change is distributed using known configurations contained within defined image templates.

'A virtualised environment involves a simulation of the underlying processor to a greater or lesser extent. Does a virtualised environment provide a realistic environment for the testing being conducted?'

many organisations in industries such as the life sciences sector, where audit compliance is critical to business.

Other organisations utilise storage virtualisation and clone differencing technology to provide complete copies of very large datasets to more test environments while utilising less storage. Clone differencing is where multiple instances of a dataset are delivered by referring to a static base copy of the data. Each instance maintains only the changes made to the original data set.

New challenges

Where virtualisation has been introduced, inevitably new challenges have arisen. This has included the following.

It has introduced a need to manage and contain the proliferation of image templates. Also the cloning of images tends to require more than just a file copy. How do we deal with IP address conflicts and naming conflicts? If we change IP addresses how do we reconfigure the applications in the templates to point to the systems they need to?

Virtualisation has sometimes been introduced to deliver pure data centre cost savings by the data centre and infrastructure teams without including the users and customers of the infrastructure in the design of the solution. This approach can lead to an ineffective implementation resulting in new issues. For example, with server virtualisation "guest" servers run on underlying shared infrastructure. Issues to single infrastructure components that previously typically would have only affected a single environment can now impact every environment.

Consider if virtualised environments are suitable for the testing being conducted. For example on performance testing, how does the workload of other environments on the same shared host physical infrastructure impact an environment? Does the virtualised infrastructure provide guaranteed computing resources to the environment? How does it compare to the go live scenario? More and more often the go live scenario is into a virtualised environment itself.

A virtualised environment involves a simulation of the underlying processor to a greater or lesser extent. Does a virtualised environment provide a realistic environment for the testing being conducted?

What are the license implications for the applications and systems within the virtualised environments?

These challenges have resulted in the creation of markets for new tools. The biggest challenge is the management of virtualised environments and there are now many solutions available in this space. Some of these are geared towards data centre management goals and others towards environment management service delivery. Examples of these virtualised lab management tools for environment management include Surgient's VQMS or VMware's Lab Manager and these provide functionality such as:

- Managing and organising a library of application configuration and versions
- Overcoming the challenges around cloning application configurations (i.e. IP address\naming conflicts, network isolation)
- Providing environment scheduling capabilities and guaranteed resource availability for reservations
- Allowing end users to self serve their environment needs, further reducing environment management support overhead
- Facilitating the sharing of environments between users
- Allowing for test automation tools to schedule and provision environments for testing at the point of execution and tearing down and releasing resources once testing has completed
- User access control to application configurations

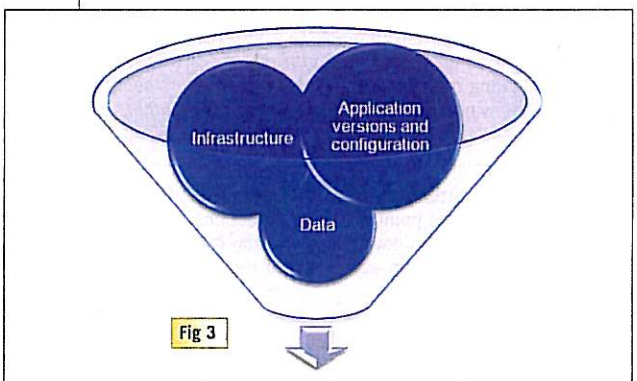


Fig 3

'Virtualisation has sometimes been introduced to deliver pure data centre cost savings by the data centre and infrastructure teams without including the users and customers of the infrastructure in the design of the solution'

Magic bullet

It is clear however, that virtualisation is not a magic bullet. There are challenges that are not addressed by virtualisation.

One issue that repeatedly arises with environment management is the difficulty and effort required to install and configure software within environments. More often than not, this is typically a manual process that has grown organically as systems and applications have been developed over time; the pace of development is so fast and no value is attached to the delivery of this information or process. In some cases the knowledge no longer exists in the organisation as to how a system is built; the knowledge has disappeared with members of staff that have long since gone. This short term speed gain does not yield long or even medium term results and quality in production soon becomes affected.

Access control can be enhanced by the use of virtualisation and other virtualisation lab management tools but are no substitute for policy and full audit capability. The amount of problems that result from quick fixes introduced to environments through the incorrect route is substantial and does not yield any long term gain.

Finally, virtualisation does not detract from the need of defining a process for environment management that aligns with an organisation's software development lifecycle, business objectives and teams. This process needs to be an integrated and coherent integration with change control, configuration management processes and the overall release management. Without good process, just as with automation tools, we end up being able to operate the same bad process only quicker.

Conclusion

Virtualisation addresses many of the challenges encountered in providing and managing development and test environments. Theresa Lanowitz, Voke Research states that development and QA time can be reduced by as much as fifty percent. However, virtualisation is only one component of an overall test environment strategy. There are challenges that are not addressed by virtualisation. Further, virtualisation does not help the environment management team integrate with other release management activities such as change control and configuration management and with the other teams that the environment management team is dependent on.

An environment management strategy must consider the tools available, including virtualisation, and ensure that process is adjusted to maximise the benefits and minimise the impact of the new challenges that the tools and virtualisation brings. This will ensure that environments of a known state are provided when they are required by the development and test teams.

Julian Brook is principal consultant with SQS Group Limited.