**Виртуальная машина**– это программа, которая эмулирует реальный (физический) компьютер со всеми его компонентами (жёсткий диск, привод,BIOS, сетевые адаптеры и т.д.). На такой виртуальный компьютер можно установить, например, операционную систему, драйверы, программы и т.д. Таким образом, Вы можете запустить на своем реальном компьютере еще несколько виртуальных компьютеров с такой же или другой операционной системой. Вы можете без проблем осуществить обмен данными между вашим реальным и виртуальным компьютером.

**Почему необходимо применять средства виртуализации**

Средства виртуализации позволяют решать самые разные задачи, в том числе возникающие перед специалистами по тестированию и обеспечению качества разрабатываемых программных приложений.

Кроме того, применение виртуальных машин (вместо физических) дает компаниям-разработчикам ряд существенных преимуществ.

Во-первых, с помощью виртуальных машин можно выполнять сравнительное тестирование разрабатываемых программных приложений на разных машинах с разными аппаратными конфигурациями. Например, управлять размером дискового пространства или оперативной памяти, ограничивать доступ к конкретным сетевым ресурсам. При этом материальные затраты на приобретение, обслуживание и обновление компьютерного «железа» фактически исключаются.

С другой стороны, сразу несколько тестировщиков могут получить в распоряжение уже заранее подготовленную тестовую машину с установленной ОС и настроенной программной средой (включая, например, локальныйSQL-сервер для построения и обслуживания баз данных, которые используются в работе тестируемого ПО).

Кроме того, виртуальные машины предоставляют удобные возможности по созданию конкретного специфического окружения, необходимого для исследования разрабатываемого ПО. Можно свободно варьировать специфичные региональные настройки и настраивать локализацию пользователей. И при этом исследователь может легко экспериментировать с настройками среды, без влияния на конфигурацию и работоспособность собственной физической машины.

Например, можно создать такую же среду, какая настроена и на стороне конечного пользователя (клиента), для имитации его работы с приложением. Допустим, Вы разрабатываете приложение в России, а клиент будет использовать его в другой стране. Зачастую это оказывает влияние на поведение приложения, значит, просто необходимо принимать во внимание все «национальные особенности».

Технологии виртуализации сейчас применяются во многих сферах ИТ как в производственной среде, так и энтузиастами и домашними пользователями для самых разных задач. Несколько одновременно запущенных виртуальных систем на одной физической машине существенно повышают гибкость ИТ-инфраструктуры и увеличивают эффективность использования аппаратных ресурсов. Тестирование программного обеспечения – один из самых распространенных вариантов использования для платформ виртуализации. Неудивительно, ведь виртуальные машины обладают множеством полезных свойств, благодаря которым значительно сокращается время разработки и тестирования и повышается эффективность этих процессов.

В классической модели разработки программного обеспечения программистам и инженерам по качеству ПО большую часть времени приходилось испытывать программный продукт на одной платформе на протяжении практически всего времени разработки и лишь в конце проводить его тестирование в различных ОС и пользовательских средах (так называемое тестирование конфигураций, Configurations Testing). К тому же, в распоряжении сотрудников отделов тестирования и разработчиков находилось не так много физических компьютеров, а на одной машине, в одной операционной системе нельзя, например, поставить одновременно несколько версий одного программного продукта, с которым должна взаимодействовать разрабатываемая программа. Вследствие этого, в программном обеспечении, особенно небольших команд разработчиков, часто встречались ошибки, связанные с особенностями пользовательских конфигураций, поскольку времени и ресурсов на полноценное конфигурационное тестирование не хватало. Кроме того, инженерам по качеству приходилось тратить много времени на развертывание комплекса программных продуктов на тестовых стендах и настройку их работы в сетевой инфраструктуре.

Безусловно, одной из самых серьезных проблем при разработке ПО является тот факт, что на ранних этапах разработки и сборки программного продукта разработчики, в процессе своей работы, могут причинить непоправимый вред системе (например, различные драйвера устройств). Поэтому приходится планировать восстановление операционных систем и их настроек после сбоев из резервных копий и тратить значительное время на их восстановление.

Надо отметить еще одну проблему, которая довольно часто встречается при разработке программного обеспечения и заставляет потратить значительное время на ее решение. Всем программистам, имеющим опыт взаимодействия с командами тестирования, известна следующая ситуация: в системе багтрекинга тестировщик фиксирует дефект, который программисту никак не удается повторить. После того, как программист ставит статус проблемы как решенный, тестировщик зовет его к своей машине и наглядно демонстрирует ошибку. В этой ситуации доходит до того, что программисту приходится установить на машину тестировщика множество отладчиков и прочей ненужной ерунды и «отлавливать» дефект, полностью останавливая работу тестировщика. Если к этому прибавить время на удаление тестировщиком программистских утилит, получаются серьезные потери времени команды.

В заключение перечисления списка проблем, возникающих в процессе разработки и тестирования, нужно привести еще одну. Зачастую складывается такая ситуация, когда требуется проверка сборки программного продукта «на дым» (так называемое дымовое тестирование, Smoke Testing), что означает быстрый прогон наиболее важных тестов. Но что если мы разрабатываем приложение, для которого требуются различные версии Internet Explorer? В этом случае будет тратиться много времени на загрузку подходящей системы, где установлена требуемая версия.

Суммируя описанные ситуации, обобщим проблемы, которые часто встречаются в процессе разработки и тестирования программных продуктов:

* Необходимость испытаний программного обеспечения в количестве пользовательских конфигураций большем, чем в распоряжении для тестирования физических компьютеров.
* Большие временные затраты на развертывание и настройку тестовых стендов, содержащих множество различных компонентов, между которыми обеспечивается сетевое взаимодействие.
* Большие временные затраты на создание резервных копий систем и их конфигураций, а также восстановление после сбоя, вследствие нестабильной работы сборок программного продукта.
* Невозможность воспроизведения дефекта, найденного тестировщиком, на машине разработчика и потеря времени на его поиск, исправление.
* Необходимость в испытаниях программы в условиях аппаратной среды, которой нет в распоряжении команды тестирования.
* Необходимость тестирования программного продукта в условиях, требующих быстрого переключения между пользовательскими конфигурациями.

Если подсчитать, сколько времени и ресурсов тратится на решение этих проблем, то получится весьма и весьма внушительная цифра, которая, будучи выраженной в денежном эквиваленте, составляет довольно большую часть бюджета проекта.

Технологии виртуализации, грамотно примененные в процессе разработки и тестирования, могут существенно снизить трудозатраты и значительно повысить эффективность процесса, что положительно скажется на качестве выпущенного программного продукта. Как конкретно виртуализация позволяет это сделать:

* Тестировщики, в процессе работы с виртуальными машинами, могут создавать неограниченное количество пользовательских конфигураций на своей физической машине запуская, по необходимости, наиболее подходящую в данный момент.
* Созданные однажды многомашинные конфигурации могут быть настроены с помощью инструментов платформ виртуализации и просто перенесены на другое оборудование, при этом их повторная настройка не требуется.
* Резервная копия виртуальной машины может быть создана путем копирования папки или создания мгновенного снимка состояния («снапшота»).
* После нахождения ошибки тестировщиком, виртуальная машина с повторяющимся дефектом может быть передана разработчику, при этом высвобождаются ресурсы на дальнейшее тестирование.
* Необходимые условия для тестирования могут быть быстро созданы за счет гибкой настройки параметров аппаратной среды виртуальной машины (объем оперативной памяти, число виртуальных процессоров, ограничение ресурсов).
* Возможность быстрого отката к сохраненному состоянию виртуальной машины с необходимой конфигурацией или переключение между несколькими одновременно работающими гостевыми системами уменьшает время на тестирование.

Все перечисленные решения нужно рассматривать в ракурсе того факта, что виртуальная машина (с установленным в ней программным обеспечением) представляет собой весьма гибкий объект, который может быть как быстро развернут на клиентских машинах из централизованного хранилища шаблонов пользовательских конфигураций, так и максимально гибко настроен в отношении параметров гостевой системы и ее окружения. Легкая переносимость на другое оборудование и независимость от аппаратной платформы – ключевое достоинство виртуальных машин.

**Инструменты виртуальных машин для тестирования и разработки**

Внедряя виртуальную инфраструктуру для целей разработки и тестирования ПО, необходимо, прежде всего, выбрать наиболее подходящую, надежную и эффективную платформу виртуализации, удовлетворяющую всем требованиям к процессу разработки в организации. Есть два наиболее распространенных пути использования виртуальных машин для тестирования программных продуктов:

* **Неуправляемое развертывание**виртуальных машин на клиентских машинах или серверах тестирования, при котором либо сами тестировщики создают необходимые им конфигурации, либо копируют шаблоны виртуальных систем на свои компьютеры из центральной библиотеки виртуальных шаблонов (файлового сервера). Преимущество данного подхода – дешевизна решения, можно воспользоваться одной из многих бесплатных систем виртуализации (VMware Server, Virtual PC, VirtualBox и другие). Основные недостатки – стихийное развертывание виртуальных машин порождает конфликты в сетевой инфраструктуре, отсутствие контроля над использованием лицензий на операционные системы и прикладное ПО, невозможность интеграции в существующую ИТ-среду организации.
* **Управляемое развертывание**достоинствам данного подхода надо отнести возможность разрешения конфликтов между виртуальными и физическими системами в сети, контроль использования лицензий, возможность мониторинга использования виртуальной инфраструктуры и создания общего пространства между различными участниками процесса разработки и тестирования, интеграция в реальную ИТ-инфраструктуру предприятия. Основной недостаток данного подхода – высокая стоимость решений. Примеры продуктов, обеспечивающих управляемое развертывание виртуальных машин: VMware LabManager, VMLogix LabManager, Microsoft System Center Virtual Machine Manager.

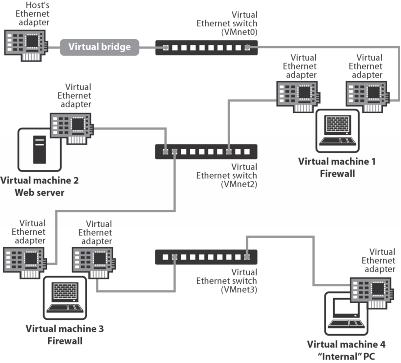
**Инструменты для разработки и тестирования при неуправляемом развертывании**

Платформы виртуализации, коих на рынке сейчас большое количество, развиваются стремительными темпами и предоставляют пользователям все больший набор инструментов для повышения эффективности процесса разработки и тестирования. Для решения каждой из перечисленных проблем платформы виртуализации различных вендоров имеют свои средства, позволяющие пользователям эффективно тестировать программные продукты в виртуальных машинах:

1. **Создание множества пользовательских конфигураций.**При наличии большого объема свободного дискового пространства на машине тестировщика с помощью платформы виртуализации можно создать неограниченное число виртуальных систем, каждая из которых может быть загружена по требованию, без остановки рабочей деятельности работника в хостовой системе. Виртуальные машины могут также применяться на специализированных серверах тестирования на основе платформ VMware ESX Server, XenEnterprise или Virtual Iron. При этом могут быть назначены определенные права пользователям виртуальных систем, а также ограничены физические ресурсы сервера, которые могут быть использованы конкретным пользователем. На файловом сервере с виртуальными шаблонами могут храниться предустановленные виртуальные машины, которые развертываются на сервера тестирования или рабочие станции. В этом случае нужно учитывать особенности использования виртуальных машин в соответствии с лицензией. В большинстве случаев каждая виртуальная машина требует отдельной лицензии, однако есть и исключения, например, лицензия Windows Server 2003 Datacenter Edition позволяет запускать неограниченное число виртуальных экземпляров ОС.

Если настроенное тестовое окружение уже развернуто на физической машине, его можно мигрировать на виртуальную с помощью продуктов вендоров платформ виртуализации и сторонних разработчиков. К таким решениям относятся продукты PlateSpin PowerConvert, VMware Converter, Microsoft Migration Toolkit.

1. **Создание многомашинных конфигураций на одном физическом сервере.**  
   Платформы виртуализации, ориентированные на тестирование ПО (VMware Workstation, Virtual PC, VirtualBox, Xen), позволяют создавать целые виртуальные инфраструктуры с различными типами сетевого взаимодействия в пределах одного физического хоста. Например, можно создать несколько «блоков» из виртуальных серверов (сервер баз данных, сервер приложений, окружение клиента), настроить сетевые адаптеры виртуальных машин (у одной машины их может быть несколько, в VMware Workstation – до десяти) и запустить тестируемую связку серверов. При этом платформы виртуализации позволяют подключать сетевые адаптеры виртуальных машин к различным сегментам виртуальной сети.

  
Рис. 11.1 - Пример виртуальной сети в пределах физического хоста.

После того, как тестовая виртуальная инфраструктура будет создана, можно настроить параметры каналов связи между виртуальными машинами.

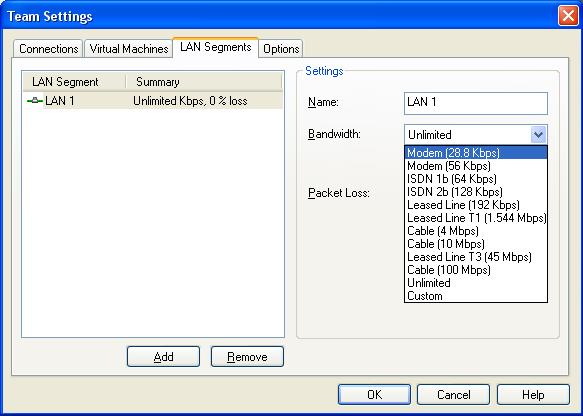


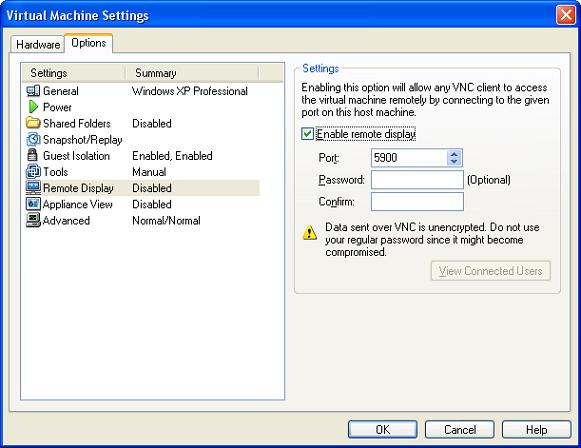
Рис.11.2. - Пример настройки пропускной способности сегмента виртуальной сети в VMware Workstation.

Нужно отметить, что виртуальные сети не на всех платформах виртуализации являются портируемыми и иногда требуется их повторная настройка при перенесении виртуальных машин на другое оборудование:

1. **Резервное копирование виртуальных машин при тестировании.**  
   Если тестировщики используют виртуальные машины на своих рабочих станциях, то они могут создавать их резервные копии путем копирования папки с файлами виртуальной машины. В случае краха системы, сохраненную копию не надо восстанавливать – она уже полностью готова к работе. К тому же, многие платформы виртуализации позволяют создавать несколько снимков состояния виртуальной машины, откат к каждому из которых может быть произведен за несколько минут.

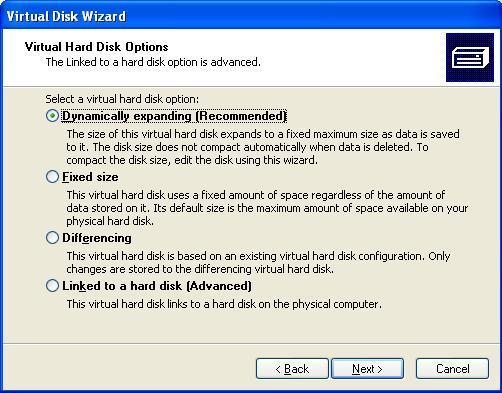
Если тестирование производится на выделенных серверах тестирования, то для создания резервных копий виртуальных машин могут применяться специализированные решения вендоров платформ виртуализации, такие как vRanger Pro компании Vizioncore, VMware Consolidated Backup (VCB) или еще не выпущенное решение Microsoft Data Protection Manager для Virtual Server 2005 R2, которые позволяют создавать резервные копии виртуальных машин без их остановки.

1. **Демонстрация дефектов разработчикам.**  
   При нахождении дефекта тестировщик может просто сохранить состояние системы, в котором проявляется ошибка, в снапшоте и продолжить тестирование системы. При необходимости демонстрации дефекта, виртуальная машина может быть передана разработчику, который сможет работать с ней, не боясь повредить окружение тестировщика. Кроме того, на платформе VMware Workstation виртуальные машины могут действовать как VNC-серверы, без необходимости установки дополнительного ПО для удаленного доступа к рабочему столу.

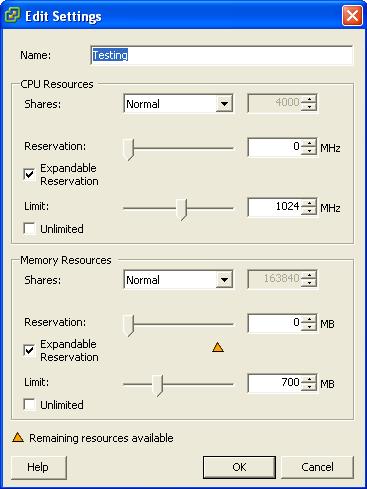
  
Рис.11.3. - Настройка доступа VNC-сервера виртуальной машины VMware Workstation.

1. **Гибкая настройка аппаратной среды.**  
   Зачастую при тестировании программного обеспечения требуется большая гибкость в отношении настройки аппаратных компонентов. Например, при стрессовом тестировании (Stress Testing) требуется проверка работы программного продукта в экстремальных или ограниченных условиях (нехватка дискового пространства, обрыв сетевого соединения). В этом случае, с помощью платформы виртуализации виртуальной машине можно добавить новые виртуальные устройства или ограничить выделяемые ей ресурсы.

При этом, если мы добавляем виртуальный диск в гостевой системе, можно создать его как динамически расширяемый, что позволяет экономить свободное место на диске, а также создать так называемые Undo-диски, изменения которых действуют только во время работы с виртуальной машиной и при завершении сеанса могут быть отменены, что очень удобно для тестирования.

  
Рис. 11.4. - Добавление виртуального диска на платформе Virtual PC 2007.

Что касается контроля ресурсов, многие платформы виртуализации позволяют ограничивать ресурсы виртуальной машины или пула ресурсов виртуальных машин, что позволяет моделировать реальные условия пользовательских окружений.

  
Рис. 11.5. - Ограничение ресурсов пула виртуальных машин на платформе VMware ESX Server.

Современные платформы виртуализации поддерживают стандарт USB 2.0, большое количество виртуальных сетевых адаптеров в виртуальной машине, эмуляцию SCSI-дисков, а также представление различного числа процессоров в гостевой системе посредством виртуального SMP (Symmetric Multi Processing):

1. **Работа с несколькими виртуальными системами одновременно.**  
   Эта возможность позволяет тестировщикам не только использовать экземпляры различных гостевых систем при тестировании, но и осуществлять простой обмен файлами как между хостом и гостевой ОС, так и между гостевыми ОС с помощью механизма Drag&Drop. При этом некоторые платформы виртуализации позволяют производить простой обмен файлами либо через общие папки хостовой системы (Shared Folders), либо «перетаскивать» файлы между гостевыми системами (VMware Workstation).

**Инструменты для разработки и тестирования при неуправляемом развертывании**

Основой при управляемом развертывании виртуальных машин являются специализированные решения для создания и обслуживания тестовых виртуальных лабораторий (Virtual Labs), в пределах которых осуществляется контроль над использованием виртуальных систем различными группами пользователей, автоматизированное развертывание виртуальных машин на компьютеры участников проекта и создание общей рабочей среды. Эти решения довольно дорогостоящие (например, инфраструктура VMware LabManager обойдется не менее чем в $15 000), однако оправдывают себя в больших масштабах использования. Крупные компании, такие как Dell, очень широко используют виртуальные машины в пределах виртуальных лабораторий для испытаний программных продуктов. Такие системы используют сети хранения данных SAN, где в общей библиотеке виртуальных шаблонов располагаются шаблоны виртуальных машин, которые развертываются по требованию на свободных серверах тестирования на основе платформ виртуализации. Использование виртуальных лабораторий дает следующие преимущества:

* Работа с многомашинными конфигурациями как с единым модулем, возможность публикации этих модулей.
* Снижение временных затрат на конфигурирование систем.
* Разграничение доступа к виртуальным машинам и демонстрация дефектов путем передачи ссылок на проблемную ситуацию, сохраненную в виде снимка состояния гостевой системы.
* Общая библиотека шаблонов с возможностью разрешения сетевых конфликтов при развертывании (идентификаторы безопасности SID, назначение уникальных MAC-адресов виртуальным сетевым интерфейсам).
* Централизованное обслуживание и развертывание обновлений в тестовых окружениях.
* Мониторинг загрузки серверов тестирования.

В данный момент, наиболее популярными решениями для управляемого развертывания виртуальной тестовой инфраструктуры являются продукты VMware LabManager (для платформы ESX Server) и VMLogix LabManager (для платформ Microsoft, VMware и XenSource).

**Тестовые лаборатории VMware LabManager**

Компания VMware предлагает крупным компаниям использовать виртуальную тестовую инфраструктуру на основе решения [LabManager](http://www.vmware.com/products/labmanager/" \t "_blank) (бывшая разработка поглощенной компании Akimbi), которое позволяет максимально быстро развертывать виртуальные машины на серверах тестирования и контролировать использование виртуальных систем, при этом процесс выглядит так, будто пользователь работает с физическим компьютером. Модель работы виртуальной лаборатории представлена на Рис. 11.6:

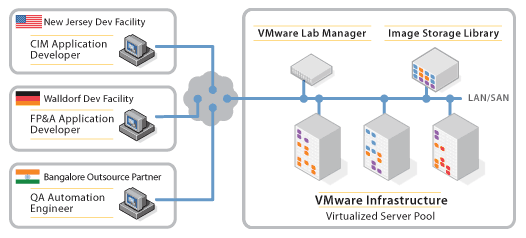


Рис. 11.6. - Архитектура решения VMware LabManager.

Помимо всех перечисленных достоинств систем управления виртуальными лабораториями, VMware LabManager предоставляет интеграцию с популярными средствами тестирования [Borland SilkCentral Test Manager](http://www.borland.com/us/products/silk/silkcentral_test/" \t "_blank) и [HP Quality Center](http://www.genilogix.com/solutions/vmware.aspx#labmanager), имеет возможности для развертывания шаблонов в несколько кликов мыши, поддерживает протокол LDAP, легко интегрируется с другими решениями для виртуальной инфраструктуры VMware и имеет возможности для автоматизации операций посредством собственного API (Application Program Interface). Основной недостаток этого продукта – возможность обслуживания виртуальных серверов только на платформах VMware.

**Тестовые лаборатории VMLogix LabManager**

В отличие от решения компании VMware, продукт [VMLogix LabManager](http://www.vmlogix.com/labmanager/" \t "_blank) поддерживает платформы виртуализации различных вендоров. В качестве основы виртуальной тестовой лаборатории можно использовать платформы Microsoft (Virtual Server), Xen (XenEnterprise) и VMware (ESX Server и Server). Кроме того, LabManager компании VMLogix поддерживает обслуживание физических серверов организации. Архитектура решения VMLogix LabManager представлена на Рис. 11.7.:

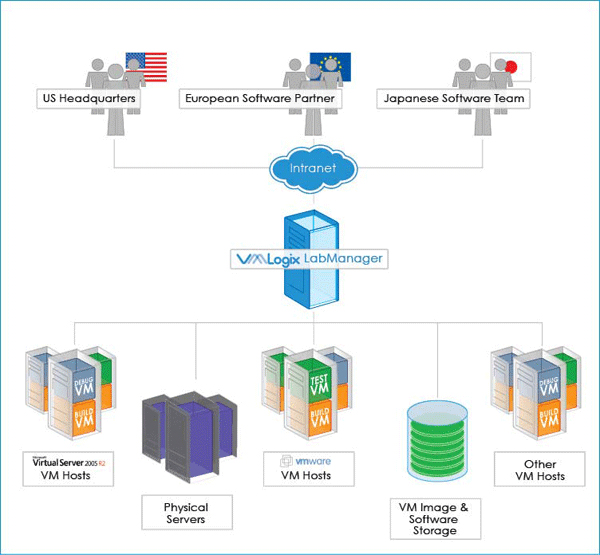


Рис. 11.7. - Архитектура решения VMLogix LabManager.

LabManager предоставляет пользователям портал самообслуживания, с помощью которого пользователи могут развертывать виртуальные машины из централизованной библиотеки шаблонов и ISO-образов операционных систем, а также имеет возможности управления лицензиями, настройки зон назначаемых IP-адресов и возможности аудита безопасности виртуальной тестовой инфраструктуры. Кроме того, VMLogix LabManager также имеет средства автоматизации операций посредством программного интерфейса API, возможности по развертыванию и обслуживанию многомашинных конфигураций и функции для демонстрации проблемных ситуаций путем предоставления общего доступа к снимкам состояния виртуальных машин.

**Заключение**

Модель организации процесса разработки и тестирования с помощью виртуальных машин позволяет существенно снизить затраты на развертывание тестовых пользовательских окружений и конфигурацию тестовых сред. По статистике, при тестировании программных продуктов на физических серверах и рабочих станциях, на эти задачи уходит до 50 процентов времени команды тестирования. Виртуальные машины на платформах различных вендоров позволяют сократить это время в несколько раз, до 5% от общих затрат на тестирование. Повышенная гибкость виртуальных систем и их независимость от оборудования позволяют работать с ними, как с некими блоками, из которых строится виртуальная тестовая инфраструктура компании. Возможность предоставления общего доступа к найденным дефектам участникам команды разработки и пользователям продукта позволяет существенно ускорить поиск и исправление ошибок. Во многих компаниях уже стало стандартом де-факто тестирование с помощью виртуальных машин.

Однако процесс тестирования на виртуальных системах имеет некоторые ограничения: например, виртуальные системы не рекомендуется применять при тестировании производительности (Performance Testing), а также - тестировании приложений, предъявляющих высокие требования к физическим ресурсам компьютера. Для других видов тестирования виртуальная тестовая инфраструктура является одним из лучших решений для повышения эффективности процесса разработки, а также упрощения взаимодействия между участниками команды.