**Анализ приложений с проблемами совместимости. Использование динамически загружаемых библиотек. Механизм решения проблем совместимости на основе «системных заплаток». Разработка модулей обеспечения совместимости**

**Решение проблем совместимости приложений при обслуживании компьютеров**

Обслуживание компьютеров организаций нередко подразумевает ведение подробного учета аппаратных компонентов.

* Используя эти данные, можно установить степень совместимости всех устройств. Впрочем, гораздо удобнее осуществлять подобную проверку при помощи специального инструментария.

Для этого можно, в частности, использовать Microsoft AssessmentandPlanningToolkit (MAP) – программу для оценки оборудования от корпорации Microsoft.

С помощью этой программы можно провести инвентаризацию компьютерного парка предприятия, оценить совместимость оборудования с выбранной версией операционной системы и получить подробный отчет о проведенной проверке. Программа работает со всеми актуальными версиями операционных систем Windows.

**При переходе на новую версию операционной системы неизбежно возникает проблема совместимости приложений, которая должна быть решена в ходе обслуживания компьютеров.**

Проблема совместимости чаще всего возникает, когда приложение некорректно обращается к функциям операционной системы.

Как правило, ведущие производители программного обеспечения стараются своевременно выпускать новые версии программных продуктов, совместимые с актуальными операционными системами, но нередко предприятия для решения некоторых специфических задач вынуждены использовать ПО, совместимость которого с новой версией операционной системы находится под вопросом.

С учетом возможных проблем совместимости используемых приложений и новой версии операционной системы рекомендуется в обязательном порядке включить в подготовку к обновлению ОС на компьютерах предприятия следующие этапы:

* обязательная проверка используемого ПО на совместимость с новой операционной системой и сбор информации о выявленных в процессе этого тестирования проблемах
* анализ приложений, у которых выявлены проблемы совместимости.  
  В процессе этого анализа нужно выяснить, насколько необходимы предприятию данные приложения, выпущены ли для них новые версии, в которых решена проблема совместимости, и в случае отсутствия таких версий - поиск инструментов для решения проблем совместимости
* тестирование инструментария для решения проблем совместимости, желательно не слишком усложняющего обслуживание компьютеров.

Лишь после получения удовлетворительных результатов такого тестирования, то есть стабилизации работы всех приложений, специалисты по обслуживанию компьютеров могут осуществлять переход на новую операционную систему.

***Самый распространенный метод решения проблем совместимости*** – это использование DLL (динамически загружаемых библиотек), которые перехватывают «проблемные» вызовы системных функций. Эти библиотеки нередко называют «системными заплатками». Во многих случаях этого метода вполне достаточно для решения проблемы совместимости.

**DLL – библиотека (динамически загружаемые библиотеки)**

Новый выпуск ОС Windows привел к появлению проблем, связанных с устаревшими интерфейсами API или библиотеками DLL. В ОС Windows устарели многие объекты, использовавшиеся в предыдущих версиях ОС. Устарели некоторые DLL – файлы, EXE – файлы, COM – объекты, разделы реестра, интерфейсы API и прочие файлы. Эти изменения влияют на любое приложение, что приводит к потере приложениями своих функциональных возможностей или к невозможности запуска этих приложений.

Библиотека (сборник подпрограмм или объектов используемых для разработки ПО) DLL – динамически подключаемая библиотека, заложена функция многократного применения различными приложениями. К DLL относятся элементы управления ActiveX (компонента, пригодная к использованию из программ, написанных на разных языках программирования) и драйверы (компьютерная программа с помощью, которой другие программы (обычно ОС) получают доступ к аппаратному обеспечению некоторых устройств).

DLL — это библиотека, содержащая код и данные, которые могут использоваться более чем одной программой одновременно. Например, в операционных системах Windows, библиотека Comdlg32 выполняет общие функции, связанные с диалоговыми окнами. Таким образом каждая программа может использовать функцию, которая содержится в этой библиотеке DLL для реализации диалогового окна Открыть. Это помогает повысить уровень повторного использования кода и эффективности память.

С помощью библиотеки DLL программы может быть модульности на отдельные компоненты. Например, программа учета может продаваться по модулям. Каждый модуль может быть загружен в основной программе во время выполнения при установке данного модуля. Поскольку программа разделена на модули, её загрузка, выполняется быстрее, и модули загружаются только при вызове их функций.

Кроме того обновления, проще применять для каждого модуля, не влияя на другие части программы. Например, имеется программа по зарплате и изменении ставок налога каждый год. Когда эти изменения изолированы с библиотекой DLL, можно применить обновление без необходимости создания или установить программу целиком.

Ниже перечислены некоторые из файлов, которые реализованы в виде библиотеки DLL в операционных системах Windows.

* Файлы ActiveX Controls (.ocx) Примером элемента управления ActiveX является элемент управления календаря, который позволяет выбрать дату с помощью календаря.
* Файлы панели (.cpl) элементов управления. Пример файла .cpl — элемент, который расположен на панели управления. Каждый элемент представляет собой специализированную Библиотеку.
* Файлы драйвера (.drv) устройства. Пример драйвера устройства — драйвер принтера, который управляет печатью на принтере.

Ниже перечислены некоторые преимущества, которые предоставляются, когда программа использует библиотеку DLL.

* **Использует меньше ресурсов**. При нескольких программ с помощью той же библиотеки функций, библиотеки DLL можно уменьшить дублирование кода, который загружается на диске и в оперативной памяти. Это может значительно повлиять на производительность не только программы, работающей в фоновом режиме, но также и другие программы, выполняющиеся в операционной системе Windows.
* **Обеспечивается модульная архитектура**. Библиотека DLL способствует разработке модульных программ. Это позволяет разрабатывать крупные программы, которые требуют нескольких языковых версий или программу, которая требует модульную архитектуру. Примером модульной программы является бухгалтерская программа, которая имеет много модулей, которые могут динамически загружаться во время выполнения.
* **Облегчает развертывание и установку.** Если функции в библиотеке DLL требуется обновление или исправление, развертывание и установка библиотеки DLL не требует повторной связи программы с библиотекой DLL. Кроме того Если несколько программ используют ту же библиотеку DLL, несколько программ будут все преимущества обновления или исправления. Такое поведение чаще наблюдается при использовании DLL независимых производителей, которая регулярно обновляется или фиксированной.

Механизм решения проблем совместимости на основе ***«системных заплаток»*** реализован в операционных системах Windows.

Для запуска приложений система может эмулировать одну из предыдущих версий Windows. Для активации данного режима не требуется специализированное обслуживание компьютеров, достаточно активировать данную опцию в свойствах исполняемого файла.

* Но не всегда запуска приложения в режиме совместимости достаточно для решения проблемы. Существует и расширенный инструментарий обеспечения совместимости. Например, Microsoft Application Compatibility Toolkit – набор, который позволяет проанализировать запросы программы к ОС и на основании этого анализа выбрать и протестировать «системные заплатки».

Использование подобных утилит позволяет, с одной стороны, упростить обслуживание компьютеров, а с другой - увеличить вероятность корректного решения проблемы совместимости.

В случаях, когда использование «системных заплаток» не привело к необходимому результату, можно использовать технологии, позволяющие создать в системе виртуальную машину, в которой и будут исполняться приложения.

Виртуализация может осуществляться с помощью приложений Microsoft Application Virtualization (App-V), виртуальной машины Virtual PC или базирующейся на ней технологии Microsoft Enterprise Desktop Virtualization или программными продуктами иных производителей. Последний метод позволяет запускать практически любые приложения, но требует существенных затрат аппаратных ресурсов компьютера.

**Что такое совместимость программ?**

Совместимость программ - это режим Windows, который позволяет выполнять программы, написанные для более ранних версий Windows.

**Первоочередные действия**

Если старая версия программы работает неправильно, прежде всего рекомендуется запустить средство устранения проблем с совместимостью программ. Оно автоматически обнаруживает и устраняет распространенные проблемы, препятствующие правильной установке или запуску более старых программ.

Откройте средство устранения неполадок с совместимостью программ. Для этого нажмите кнопку ПускИзображение кнопки «Пуск» и выберите пункт Панель управления. В поле поиска введите неполадки и затем выберите пункт Устранение неполадок. В разделе Программы щелкните команду Выполнение программ, предназначенных для предыдущих версий Windows.

**Пять этапов подготовки приложений к новой Windows**

Ни одна глобальная цель не является недосягаемой, если для ее реализации выполнить ряд логически завершенных, реалистичных задач.

Проект по подготовке приложений к работе в новой ОС включает в себя следующие три этапа: сбор информации, анализ приложений и поиск решения проблем совместимости.  Однако есть еще несколько дополнительных этапов, о которых мы бы хотели рассказать.  Среди них анализ технологий виртуализации перед началом тестирования для сокращения объема тестирования, возможной оптимизации инфраструктуры настольных систем и упрощения решения проблем совместимости в будущем; очередность приложений на этапе тестирования должна соответствовать стратегии развертывания.

Рассмотрим эти этапы подробнее.

Этап 1:  Сбор сведений о приложениях

На первом этапе необходимо собрать сведения о приложениях, чтобы оценить объем работ. В это время, как правило, приходит осознание, что проблема намного серьезнее, чем предполагалось ранее.  Но намного важнее, что вы «узнали врага в лицо», а это позволяет лучше оценить объем работы по тестированию и подготовке миграции и понять проблемы, которые предстоит решить.

К счастью, существует ряд средств, позволяющих автоматизировать этот процесс.  Они могут быть встроены в программное обеспечение для управления клиентом. В противном случае можно бесплатно загрузить и использовать набор средств [Application Compatibility Toolkit](http://go.microsoft.com/?linkid=9659775). Если у вас есть другой механизм инвентаризации, например SystemCenterConfigurationManager, AssetInventoryService или какой-либо другой, на начальном этапе можно использовать его.

Чтобы максимально эффективно провести инвентаризацию, собирайте не только список приложений, но и более подробные сведения о пользователях приложений, имеющихся у них разрешениях и важности приложений для пользователей.  Эти сведения позволяют назначить критически важным приложениям более высокий приоритет и отказаться от использования ненужных или дублирующих приложений (более подробные сведения см. в описании следующего этапа).

Такой подход имеет еще одно неявное преимущество — выявление наиболее часто используемых приложений, которыми вы не управляете в настоящее время.  Сведения о таких приложениях необходимы для эффективного управления ими, обеспечения наличия одобренной версии и необходимых обновлений.

Этап 2: Анализ приложений

Сколько поддерживаемых в настоящее время приложений было заменено другими? А сколько используется сотрудниками предприятия менее активно?  Для большинства организаций ответ на эти вопросы— «много», а иногда — «большинство».   Как только оценка будет завершена, а «почва — подготовлена», можно переходить к следующему этапу — анализу списка поддерживаемых приложений и их приоритизации, после чего последует трудоемкий (и затратный) процесс регрессивного тестирования.

Определите цели для вашего портфеля приложений.  Поддержку какого числа приложений необходимо обеспечить?  На каком этапе приложение получает статус «управляемого»?

После определения целей необходимо решить самую очевидную задачу и сузить список приложений, которые необходимо протестировать.

* Исключите ненужные и неиспользуемые приложения. С большой вероятностью можно утверждать, что обнаружится несколько приложений, выполняющих одну и ту же функцию. Самое время исключить приложения, выполняющие одинаковые функции, и оставить по одному приложению на каждую выполняемую функцию. Рекомендуется сопоставить зависимости приложения, поскольку для использования решения стороннего поставщика программного обеспечения может потребоваться поддержка более ранних версий приложения. И, конечно же, необходимо отказаться от приложений, которые не используются совсем или используются редко. Это не только упростит тестирование, но также позволит сократить расходы на лицензирование.
* Удалите все версии одного и того же приложения, кроме последней. Практически во всех случаях последняя версия характеризуется наиболее высокой производительностью, безопасностью и надежностью. По-прежнему следует обращать внимание на возможные зависимости между приложениями.

Соберите данные от корпоративных пользователей, чтобы определить, какие из приложений являются наиболее важными и какие подразделения используют те или иные приложения.  Эти сведения будут полезны при определении последовательности при тестировании; очередность развертывания приложений должна соответствовать стратегии развертывания нового образа рабочей среды.

Этап 3:  Оценка проблем совместимости и способов их решения

Несомненно, вы найдете несколько приложений, требующих дополнительной подготовки для работы в Windows.  На этом этапе доступны следующие варианты:

1. Можно заменить несовместимое приложение его новой версией. Это самый надежный способ, но, к сожалению, и самый дорогой. Если приложение является критически важным или имеет стратегическое значение для выполнения операций, следует отдать предпочтение этому способу.
2. Создание оболочек совместимости для существующих приложений. Оболочки совместимости — это небольшие блоки кода, которые вставляются между приложением и ОС Windows для изменения вызовов к базовой операционной системе. Например, приложение можно «перехитрить», заставив его думать, что пользователь обладает правами администратора, в то время как используется режим обычного пользователя. Этот подход приведет к некоторым дополнительным расходам на управление, поскольку потребуется поддерживать базу данных оболочек совместимости, но вместе с тем он позволит решить многие проблемы совместимости приложений. Этот метод является более экономичным и может быть единственным выходом из ситуации в случае, когда производитель приложения недоступен. Обратите внимание, что многие поставщики не предоставляют техническую поддержку приложений с оболочками совместимости.
3. Использование групповой политики для изменения нежелательного поведения приложения. Как и в случае с использованием оболочек совместимости, этот подход решает проблему совместимости, однако он имеет свои недостатки. По сути, этот подход использует политику для отключения определенной функции, которая приводит к сбою приложения. К сожалению, часто в их число попадают функции безопасности базовой системы, поэтому негативный эффект очень значительный. Кроме того, эта управляемость должна обеспечиваться за счет поддержки параметров групповой политики в приложении.

В приложениях, разработанных на предприятии или адаптированных пользователями, *можно изменить код*.  Не всегда можно использовать этот вариант, однако, если такая возможность есть, то можно воспользоваться великолепным набором средств. [Настольная книга по совместимости приложений](http://msdn.microsoft.com/library/dd371778%28VS.85%29.aspx), например, отмечает изменения, необходимые для перехода от WindowsVista® к Windows® 7. Данный набор является бесплатными руководством, позволяющим разработчикам изменить код приложения для обеспечения полной совместимости.

Этап 4: Подготовка развертывания новой операционной системы и варианты предоставления приложений

Начало проекта по миграции на новую операционную систему — это самое удобное время для переосмысления способа предоставления приложений пользователям.  Технологии виртуализации открывают возможности, которые не были доступны во время последней значительной миграции на новую операционную систему. Следует рассмотреть различные модели образа рабочей среды и способы предоставления приложений пользователям перед началом процесса тестирования.  Вы можете заметить, что стоимость развертывания виртуальной среды намного меньше той экономии, которая достигается за ее счет при тестировании и подготовке приложений; к тому же эти технологии предоставляют более гибкую и управляемую среду для будущей работы.

Существуют две формы виртуализации, которые помогут упростить решение проблем совместимости приложений — виртуализация приложений и виртуализация операционной системы.  Виртуализация приложений отделяет уровень приложения от операционной системы, включая файлы приложения и параметры реестра, и упаковывает приложение для выполнения. Виртуализация операционной системы принимает разные формы, но, по сути, создает образ операционной системы, независимый от образа, установленного на компьютере.

Виртуализация портфеля приложений предоставляет ряд преимуществ в отношении управляемости и гибкости, однако главным плюсом является сокращение числа конфликтов между приложениями.  Такие конфликты возникают, например, при необходимости запустить две версии одного и того же приложения одновременно — типичная ситуация при обучении, когда необходимо сравнить процесс выполнения конкретной задачи в старой и новой версиях приложения, или когда при миграции финансового отдела на новую версию программы для бухгалтерского учета требуется доступ к предыдущей версии, чтобы закрыть предыдущий финансовый год.

Более универсальный способ решения проблемы совместимости приложений с помощью технологии виртуализации — это создание виртуального образа, содержащего критически важное приложение и операционную систему, для которой оно разработано.

Конечно, использование альтернативной модели вычислений на клиентском ПК — это большая проблема, однако оно позволит сопоставить преимущества для организации (большая гибкость и управляемость) и дополнительные затраты, связанные с переходом на эту модель подготовки ПК.

Этап 5: Планирование последовательности тестирования, экспериментального и окончательного развертывания

Составьте последовательность проведения тестирования с помощью приоритетов, определенных на этапе 2. Они позволяют постепенно осуществлять развертывание и параллельно выполнять тестирование в соответствии с определенной последовательностью.

Существуют два подхода к проведению тестирования: статический и динамический анализ. Несмотря на то что статический анализ появился относительно недавно, тщательное тестирование предполагает сочетание обоих подходов.

* В статическом анализе рассматривается структура приложения, и определяются проблемы, которые непременно возникнут во время установки или запуска приложения. Для автоматизации данного процесса и быстрого выявления очевидных проблем используются различные средства и службы.
* В динамическом анализе рассматривается поведение приложения во время выполнения; этот подход традиционно используется при регрессивном тестировании. Проверка приложения осуществляется в «боевых условиях»: в рабочей среде моделируются ситуации, которые могут возникнуть с оборудованием или другими важными приложениями и драйверами при использовании данного приложения.
* Наконец, приложения должны пройти проверку реальными пользователями. Они запускают приложение, работают в нем и наблюдают за любыми отклонениями от нормального поведения, которые не были обнаружены во время структурированного тестирования. Обещание оставить пользователю новый компьютер после участия в экспериментальном развертывании может быть отличным мотивирующим фактором.

Как только вы будете готовы приступить к развертыванию в производственной среде, определите пользователей, которых нужно перевести на новую систему в первую очередь, на основе необходимых им функций или из соображений обеспечения непрерывности бизнес-процессов.  Осуществить миграцию группы опытных пользователей проще, чем решать проблемы сотрудников, выполняющих рутинную работу, которые запустили незнакомую систему и не знают, что с ней делать. Далее следует определить, какие приложения потребуются данной группе для выполнения работы. Начать следует с групп, которых либо не коснутся, либо незначительно коснутся проблемы совместимости приложений. Определить это можно на основе анализа используемых приложений. Благодаря этому вы сможете подтвердить процесс развертывания и операционную систему. По мере обработки портфеля приложений и их подготовки длявсе большего числа пользователей, следует сосредоточиться на новых группах.

Еще одно предупреждение. Помните, что данный процесс относится только к обеспечению совместимости приложений.  Если выйти за рамки проекта и превратить его в полноценный проект по повышению качества приложений, велика вероятность никогда не завершить его.  Ограничьтесь исправлением только тех ошибок, которые препятствуют выполнению работы, избегайте стремления устранить все существующие ошибки. Вы можете более продуктивно распорядиться своим временем.

***Развертывание и внедрение***

Этап развертывания системы является одним из наиболее важных с точки зрения распределения ответственности за работы между заказчиком и подрядчиком. Исполнитель выполняет основную задачу – инсталляцию модуля на рабочий сервер с перенесением на него итоговую конфигурацию с тестового сервера. Однако при верном последовательном обучении и вовлечении специалистов заказчика в процесс создания / внедрения системы остальные операции могут быть осуществлены ими:

1. Предшествующее работам обеспечение охвата всех автоматизируемых рабочих мест пользователей функционирующей ЛВС.
2. Настройка рабочего сервера.
3. Определение перечня / количества рабочих мест, которые необходимо развернуть для ОПЭ.
4. Развертывание рабочих мест пользователей с определением прав доступа.
5. Подготовка наиболее актуальной информации по каждому модулю для ввода в систему (включая подготовку, дополнение и выверку справочников, которые могли потерять актуальность за время между тестовой эксплуатацией и развертыванием системы).

Организация поддержания данных системы в актуальном состоянии (в том числе при интеграции с другими системами).

Целью данного этапа является подготовка модуля системы к опытно-промышленной эксплуатации – подготовка конечных пользователей, ввод начальных данных, подготовка приказов по предприятию заказчика о передаче модуля системы в опытно-промышленную эксплуатацию.

Среди основных работ этапа:

* 1. Обеспечить безусловное выполнение условий готовности модулей системы к сдаче в опытно-промышленную эксплуатацию (закрепленных в отдельном документе).
  2. Ввести и выверить начальные данные (например, начальные остатки) по каждому модулю для ввода информации в систему.
  3. Отработать на рабочих местах пользователей модуля системы практические действия пользователей в параллельном режиме с имеющимися приложениями.
  4. Разработать, согласовать с подрядчиком и утвердить регламент взаимодействия подразделений заказчика, участвующих в опытно-промышленной эксплуатации модуля системы.
  5. Осуществить интеграцию модуля с другими модулями или внешними системами, внедренными ранее.
  6. Подготовить и издать приказ по предприятию заказчика о передаче модуля системы в опытно-промышленную эксплуатацию, который должен содержать:
     1. наименование модуля системы, проходящего опытно-промышленную эксплуатацию;
     2. наименование компании-исполнителя;
     3. сроки проведения опытно-промышленной эксплуатации;
     4. список должностных лиц со стороны заказчика и исполнителя, ответственных за проведение опытно-промышленной эксплуатации модуля;
     5. перечень подразделений предприятия заказчика, участвующих в проведении опытно-промышленной эксплуатации;
     6. порядок и сроки перевода персонала заказчика на работу в условиях функционирования модуля системы.

моменту ОПЭ уже должны быть проведены следующие работы:

* 1. Модули системы успешно перенесены и функционируют на рабочем сервере и автоматизированных рабочих местах пользователей (с разделением прав доступа).
  2. Подготовлены, дополнены и введены недостающие справочники по каждому модулю системы, проведена выверка введенных в систему недостающих справочников.
  3. Заказчиком утверждены итоговые документы.

**Задание:**

1. Подготовить 3 тестовых вопроса по данному материалу (не менее 5-ти вариантов ответов)

2. Подготовить презентацию