***Глава 16***

***Однократное устранение проблем***

Исправить что-то один раз лучше, чем исправлять это постоянно. Несмотря на

то что это кажется очевидным, иногда такой подход невозможен из-за тех или

иных ограничений, либо оказывается, что вы постоянно исправляете одну и ту

же неполадку, не понимая этого, либо решение на скорую руку просто кажется

более удобным. Учитывая это, вы можете поставить перед собой несколько це-

лей. Во-первых, вы можете лучше планировать свое время. Во-вторых, вы мо-

жете стать лучшим системным администратором. В-третьих, если это необхо-

димо, вы лучше можете объяснить пользователю, почему для исправления

какой-то проблемы вам потребовалось больше времени.

В главе 15 рассматривался системный процесс устранения проблемы. Данная

глава посвящена общей концепции текущей ситуации.

***16.1. Основы***

Один из наших любимых принципов – «исправляйте один раз». Если что-то

нарушено, то это должно быть исправлено однократно, чтобы больше к этому

не возвращаться. Если проблема может появиться на других машинах, то на

них нужно проверить ее наличие и исправить неполадку.

**16.1.1. Не тратьте время зря**

Иногда, особенно в случае с проблемами, которые кажутся тривиальными или

затрагивают вас, только когда проявляются, может показаться легче исправить

что-то на скорую руку, не устраняя проблемы окончательно. У вас может даже

не возникнуть мысли о том, что вы постоянно решаете одну и ту же проблему,

которая могла быть устранена однократно ценой чуть больших усилий.

**16.1.2. Избегайте временных решений**

Предыдущий раздел был довольно оптимистичным в плане возможности

в каждой ситуации исправить неполадку наилучшим образом. Однако это не

всегда возможно. Иногда ограничения по времени или ресурсам требуют быстро-

го решения, пока нельзя запланировать полное устранение проблемы. Иногда

полное исправление может требовать неприемлемого в определенных ситуациях

прерывания обслуживания и приходится довольствоваться временными мерами,

пока не будет назначен технологический перерыв. Иногда временные меры тре-

буются из-за причин, связанных с ресурсами. Может быть, для устранения про-

блемы нужно написать программу или установить оборудование. Это займет

опре­деленное время. Если диск заполнен логами, то постоянной мерой может стать

установка программы, обновляющей логи. Установка такой программы может

занять определенное время, но в это время можно удалить старые логи вручную.

Важно, чтобы за временными решениями следовали постоянные. Для этого

требуется определенный механизм, чтобы проблемы не оставались незамечен-

ными. Возвращаясь к нашему примеру с диском, заполненным логами, при

большой рабочей нагрузке может быть очень соблазнительно удалить старые

логи и перейти к следующей задаче, не отметив, что к вопросу нужно вернуться

для его окончательного решения. Записывать такие действия может быть труд-

но. Написанные на бумаге заметки теряются. Не у каждого всегда есть под рукой

ежедневник. Гораздо проще отправить себе напоминание по электронной почте.

Наличие приложения, позволяющего отправлять новые заявки по электронной

почте, даже лучше. Если вы можете создать новую заявку при помощи элект-

ронной почты, то будете способны создать ее где угодно, чтобы вам не нужно

было помнить о ней. Обычно электронную почту можно отправлять с сотового

телефона, двустороннего пейджера, коммуникатора или КПК.

UNIX-системы обычно можно настроить для отправки электронной почты из

командной строки. Не нужно ждать, пока запустится почтовый клиент. Просто

введите одно-два предложения для напоминания и создайте заявку позже, ког-

да будет время1. Однако нужно отметить следующее: во многих местах UNIX-

системы настраиваются так, что могут правильно отправлять электронную

почту, только если являются частью цепи доставки электронной почты. Поэтому

электронная почта, отправленная из командной строки на других машинах, не

доставляется. Очень легко создать простую конфигурацию «нуль-клиента» или

«перенаправления всей электронной почты на сервер», которая устанавливает-

ся как опция настройки по умолчанию. Другие способы являются непрофессиональными и ведут к путанице, связанной с потерей электронной почты.

Временные меры чисто психологически кажутся проще постоянных. Мы чувству-

ем, что сделали что-то значительное за небольшое время. Для нашего самомнения

это гораздо приятнее, чем начинать крупный проект для однократного устранения

проблемы или добавления нового пункта в наш бесконечный список дел.

Исправление одних и тех же мелких недостатков раз за разом входит в привыч-

ку. С точностью собаки Павлова мы выполняем одни и те же действия каждый

раз, когда наши системы мониторинга уведомляют нас о проблеме. После этого

мы убеждаем себя покончить с проблемой раз и навсегда: «В следующий раз

у меня будет время для окончательного исправления!» Со временем мы начина-

ем так хорошо делать временные «заплатки», что забываем о существовании

постоянных мер. Мы ощущаем себя постоянно занятыми, но не чувствуем, что

что-то выполняем. Мы просим начальников или коллег взглянуть на наш рабо-

чий день свежим взглядом. Сделав это, они видят, что наше время тратится на

вытирание пола, вместо того чтобы просто выключить воду.

Мы так привыкаем к временным мерам, что становимся в них экспертами.

Стыдно, но, обнаруживая, насколько мы преуспели в них, мы гордимся своими

успехами, демонстрируя написанные нами макросы и другие методы экономии

времени, которые мы нашли.

Такая ситуация является распространенной. Чтобы ее предотвратить, нужно

разорвать замкнутый круг.

Давайте более подробно рассмотрим вывод A: устраняйте проблему раз и навсег-

да. Конечно, он кажется простым, однако мы часто видим, что неполадку ис-

правляют только для того, чтобы увидеть, что при перезагрузке она появляется

снова. Иногда знание того, какие меры являются постоянными, а какие при

перезагрузке нужно повторять, как раз и отличает неопытного системного ад-

министратора от мастера.

Многие ОС запускают скрипты, или программы, при загрузке машины. Скрип-

ты, включенные в процесс загрузки машины, нужно время от времени редак-

тировать. Иногда нужно запустить новый демон, например HTTP-сервер Apache.

Порой нужно внести изменение в конфигурацию, например установить флаг на

новом сетевом интерфейсе. Вместо введения этих команд вручную при каждой

перезагрузке машины их нужно внести в скрипты автозагрузки. Будьте внима-

тельны при написании таких скриптов. Если в них вкрадется ошибка, система

может больше не загрузиться. Мы всегда перезагружаем машину вскоре после

изменения любых скриптов автозагрузки, таким способом мы выявляем про-

блему сразу, а не через несколько месяцев, когда перезагрузка машины действительно потребуется.

**16.1.3. Учитесь у плотников**

Системные администраторы могут многому научиться у плотников. Они стро-

или и ремонтировали гораздо дольше, чем мы.

Плотники говорят: «Семь раз отмерь, один отрежь». Повторное измерение пре-

дотвращает множество ошибок. Дерево стоит дорого. Небольшое дополнитель-

ное внимание стоит недорого по сравнению с потраченным зря деревом.

Кроме того, плотники понимают, как копировать предметы. Плотник, которо-

му нужно отрезать несколько деревянных брусков одного размера, отрезает

первый брусок нужной длины и использует его для измерения других. Это го-

раздо точнее, чем пользоваться вторым бруском для измерения третьего, тре-

тьим – для измерения четвертого и т. д. Последний метод легко приводит

к накоплению ошибок.

Системные администраторы могут многое почерпнуть из этих методов. Копи-

рование чего-либо – это возможность один раз сделать что-то правильно, а затем

повторить это много раз. Измерять несколько раз – хорошая привычка. Прове-

ряйте свою работу дважды, прежде чем окончательно внести какие-либо изме-

нения. Перечитайте файл конфигурации, попросите кого-нибудь еще посмотреть

на команду перед выполнением, проверьте емкость системы, прежде чем реко-

мендовать расширение. Проверяйте, проверяйте и еще раз проверяйте.

Вы никогда не услышите, чтобы плотник сказал: «Я подпилил эту доску три

раза, а она еще слишком короткая!» Подпиливание одной и той же доски не

сделает ее длиннее. Системные администраторы часто делают одно и тоже раз

за разом, и их раздражает, что они постоянно получают одинаковые неудачные

результаты. Вместо этого нужно попробовать что-то другое. Системные адми-

нистраторы жалуются на проблемы безопасности и ошибки, но доверяют про-

граммному обеспечению от компаний без адекватных систем контроля качест-

ва. Системные администраторы запускают критически важные системы без

брандмауэров для Интернета. Системные администраторы исправляют непо-

ладки путем перезагрузки вместо устранения первопричины.

***16.2. Тонкости***

Данный раздел этой главы повествует о том, как устранять проблемы не само-

стоятельно, а за счет автоматизации. Один из типов автоматизации устраняет

симптомы и сообщает системному администратору, чтобы он обеспечил посто-

янные меры. Другой тип автоматизации принимает постоянные меры самосто-

ятельно.

Автоматизация, устраняющая проблемы, может быть опасной. Мы видели

слишком много плохой фантастики, в которой робот «устраняет» проблему,

убивая невинных людей или уничтожая Землю. Следовательно, автоматизация

должна быть особенно осторожной в своих действиях и должна вести логи,

чтобы ее работу можно было проверить.

Автоматизация часто исправляет симптомы, не устраняя первопричину.

В данной ситуации очень важно, чтобы автоматизация обеспечивала сообщение

о том, что что-то было сделано, – тогда человек сможет обеспечить принятие

постоянных мер. Мы видели автоматизацию, которая решает проблему запол-

нения диска путем удаления старых файлов логов. Это работает хорошо, пока

потребители дискового пространства не потеснят чрезмерно файлы логов

и удаляемые файлы логов вдруг не станут слишком маленькими. Тогда автома-

тизация требует немедленного вмешательства человека, а человек обнаружи-

вает очень сложную проблему. Если бы автоматизация сообщила человеку, что

она приняла временные меры, то она предоставила бы человеку время для при-

нятия постоянных мер.

Однако в этом случае мы рискуем попасть в ситуацию «мальчика, который

кричал про волка». Очень легко привыкнуть игнорировать предупреждения

о том, что автоматика приняла временные меры и требуется более глобальное

решение. Если в этот раз временные меры сработали, то они сработают и в сле-

дующий раз. В первый раз игнорировать такое сообщение обычно безопасно.

Постоянные меры нужно принимать в следующий раз. Так как у системного

администратора работы обычно всегда больше, чем времени на нее, нетрудно

предсказать, что «следующий раз» может быть очень нескоро. В больших сис-

темах велика вероятность того, что предупреждения каждый раз будут видеть

разные системные администраторы. Если каждый из них посчитает, что он про-

игнорировал сообщение первым, ситуация превратится в серьезную проблему.

Устранение самой проблемы редко можно автоматизировать. Автоматизация

может помочь вам при небольшом сбое, но не способна исправить программное

обеспечение с ошибками. Например, она может отключить неуправляемый

процесс, но не может исправить ошибку в программе, которая делает его неуп-

равляемым.

Иногда автоматизация может устранить саму проблему. Крупные системы

с виртуальными машинами могут выделить дополнительные процессоры для

ресурсоемких вычислений, увеличить размер заполненного раздела или авто-

матически переместить данные на другой диск. Некоторые типы файловых

систем позволяют вам автоматически создавать виртуальную файловую систе-

му, обычно выделяя свободный диск и присоединяя его к разделу. Это не силь-

но помогает, если диск был заполнен из-за неуправляемого процесса, создаю-

щего бесконечное количество данных, потому что новый диск будет заполнен

так же быстро. Однако это устранит ежедневную рабочую проблему заполнения

дисков. Вы можете добавить в систему пустые диски – и автоматизация обеспе-

чит их присоединение к очередным практически заполненным виртуальным

разделам. Это не заменяет хорошее планирование использования дискового

пространства, но будет неплохим средством в качестве элемента вашей системы

управления дисковым пространством.

Решением являются политика и поддержание порядка, выполнение которых,

возможно, обеспечивается программами. Устранение проблем вместо их игно-

рирования требует поддержания порядка.

Иногда создание автоматизации требует много времени. Однако порой ее мож-

но создать быстро. Основные этапы 5-минутной задачи можно интегрировать

в скрипт. Затем можно добавить другие элементы задачи. Это может выглядеть

как трата часа на автоматизацию 5-минутных задач, но в долгосрочной перспективе вы сэкономите время.

***16.3. Заключение***

Лучше исправить что-то один раз, чем исправлять постоянно. В конечном ито-

ге меры должны быть постоянными, а не временными. Вы не должны изобретать

велосипед – старайтесь по возможности копировать решения, работоспособность

которых известна. Лучше предупреждать проблемы: если вы обнаружите про-

блему в одном месте, исправьте ее на всех похожих узлах или во всех местах.

Системному администратору легко привыкнуть к ситуации и забыть, что про-

блемы нужно устранять правильно. Однако иногда ограниченные ресурсы не

позволяют системному администратору ничего другого, кроме как принять

временные меры и запланировать окончательное решение на будущее. С другой

стороны, системный администратор должен избегать привычки откладывать

такие меры и следовать по пути повторного применения небольших «заплаток»,

который является психологически комфортным. Вместо этого следует найти

время для реализации полного решения. В конце концов, лучше всего устранять

проблемы правильным методом и своевременно.

Данная глава была чуть более философской, чем остальные. В первом примере

мы увидели, как важно с самого начала усвоить основы. Если обеспечена авто-

матизация первоначальной загрузки и конфигурации ОС, большинства проблем

просто не возникло бы. Во многих случаях постоянной мерой является обеспе-

чение автоматизации. Однако у автоматизации есть свои недостатки. Создание

автоматизации решения может занять много времени, и в ожидании завершения

автоматизации у системного администратора могут появиться плохие привыч-

ки или невосприимчивость к напоминаниям о необходимости принятия посто-

янных мер. Тем не менее хорошая автоматизация может существенно снизить

вашу загруженность работой и повысить надежность ваших систем.

***Глава 17***

***Управление изменениями***

**Управление изменениями** – это процесс, который обеспечивает эффективное

планирование, реализацию и последующий анализ изменений, внесенных

в систему. Это означает, что изменения хорошо документируются, имеют план

отмены и возможность воспроизводства. Управление изменениями касается

управления риском. Изменения, которые вносят системные администраторы,

приводят к риску сбоя обслуживания их пользователей. Управление изменени-

ями означает оценку этих рисков и управление ими посредством стратегий

смягчения рисков. Оно включает заблаговременное планирование изменений,

доведение информации до сотрудников, согласование графика, план проверки,

план отмены и набор условий, определяющих, нужно ли реализовывать план

отмены и когда. В данной главе рассмотрен базовый процесс, а дальнейшие

главы показывают, как применяется управление изменениями в различных

аспектах работы системного администратора.

Управление изменениями создает отчетные данные, которые могут быть ис-

пользованы для определения, что было сделано, когда и почему. Элементом

управления изменениями является обсуждение проекта с пользователями

и другими группами системных администраторов до его реализации. Контроль

версий, другой компонент управления изменениями, является процессом низ-

кого уровня для контроля изменений в отдельных файлах конфигурации.

Управление изменениями – один из основных процессов развитой группы сис-

темного администрирования. Это механизм, посредством которого группа может

обеспечить уверенность в том, что изменения, которые могут влиять на другие

изменения, не произойдут одновременно. Это механизм снижения количества

сбоев или проблем за счет того, что системные администраторы вынуждены

продумывать различные аспекты изменения перед его реализацией. Это способ

связи, который обеспечивает уверенность в том, что при внесении изменений

все будут действовать согласованно. Другими словами, это означает снижение

уровня ошибок и возможность быстрее справиться с ошибкой, когда она про-

изойдет. Управление изменениями критически важно для компаний электрон-

ной коммерции, чьи доходы основаны на доступности в режиме 24/7.

***17.1. Основы***

В данном разделе мы рассмотрим, как управление изменениями связано с уп-

равлением риском, и покажем четыре основных компонента управления изме-

нениями для системных администраторов:

1. Общение и составление графика. Общайтесь с пользователями и другими

системными администраторами, чтобы они знали, что происходит, и со-

ставляйте график изменений, чтобы влияние было наименьшим.

2. Планирование и проверка. Планируйте, как и когда вносить изменение,

как проверить, что все работает, и как и когда отменить изменение, если

будут проблемы.

3. Процессы и документация. Изменения должны соответствовать стандарт-

ным процессам и хорошо планироваться, с рассмотрением всех возможных

последствий. Изменения должны быть документированы и одобрены пе-

ред их реализацией.

4. Контроль версий и автоматизация. Используйте контроль версий для от-

слеживания изменений и облегчения отмены проблемных обновлений. По

возможности автоматизируйте изменения, чтобы обеспечить точное и вос-

производимое выполнение всех процессов.

Мы покажем, как все вместе эти компоненты могут обеспечить плавное обнов-

ление системы с минимумом проблем.

Управление изменениями предполагает учет различных категорий изменяемых

систем, типов вносимых изменений и особых процедур для каждой комбинации.

Например, категории машин могут включать рабочие станции, серверы подраз-

делений, системы корпоративной инфраструктуры, критические для бизнеса

системы, присутствие в Интернете и рабочие системы электронной коммерции.

Категории изменений могут включать управление учетными записями и досту-

пом, обновление директорий, установку новой службы или программы, модер-

низацию существующей службы или программы, изменения оборудования,

изменения политики безопасности или конфигурации.

Небольшие изменения могут выпасть из процесса компании по управлению

изменениями и часто будут из него выпадать, наличие слишком громоздкого

процесса для небольших изменений не позволит системным администраторам

эффективно работать. Но значительные изменения должны быть предметом

процесса полного управления изменениями. Наличие такого процесса означает,

что системный администратор не может внести значительное изменение, не

соблюдая правильную процедуру, которая предполагает общение с нужными

людьми и планирование изменения на приемлемое время. В критических сис-

темах это может включать написание небольшого плана проекта с процедурами

тестирования и планом отмены, который просматривается равными по долж-

ности или старшими системными администраторами, а также может предпо-

лагать назначение помощника для наблюдения и помощи с изменением.

Компания должна определить, какой уровень процесса управления изменени-

ями должен быть у каждого элемента матрицы категорий машин/типов изме-

нений. Очевидно, что критически важные для бизнеса системы нужно жестко

контролировать. Изменения в отдельном компьютере могут не требовать тако-

го контроля. Изменения в каждом компьютере в компании, скорее всего, будут

его требовать. Наличие достаточного количества процессов для изменений

и возможностей обзора изменений в более важных системах приводит к сниже-

нию числа потенциально затратных ошибок. Библиотека инфраструктуры

информационных технологий (ITIL – Information Technology Infrastructure

Library) – ценный ресурс для дальнейшего изучения области управления изме-

нениями и рабочих процессов системных администраторов. Процессы передовых

методик ITIL становятся широко распространенными стандартами.

**17.1.1. Управление риском**

Управление риском – часть деятельности системного администратора. Основные

риски, которые нас касаются, связаны с невыполненным обслуживанием, под-

классом которого является потеря данных. Один из основных способов, при

помощи которых системные администраторы управляют риском, – это создание

резервных копий. Резервные копии являются элементом стратегии смягчения

рисков, защищающим от потери данных и невыполненного обслуживания.

В главе 25 мы более подробно рассмотрим, как различные технологии, такие

как RAID, помогают нам смягчить риск невыполненного обслуживания из-за

потери данных.

Первые шаги по управлению риском – это выявление рисков и количественная

оценка рисков. На какие системы и службы может повлиять изменение? Како-

вы наихудшие возможные сценарии развития событий? Скольких ваших поль-

зователей могут затронуть эти сценарии? Это помогает разделить машины по

профилям использования, таким как инфраструктурные машины, серверы

подразделения, критичные для бизнеса, или рабочие станции, и подсчитать

количество машин, на которые влияет изменение.

Следующий шаг после оценки риска изменения – определить, как смягчить

риск. Смягчение имеет пять основных компонентов. Первый из них – провести

консультацию по изменению: касается ли это изменение удовлетворения потребностей бизнеса, влияет ли он на другие события и изменения, когда оно

должно быть реализовано? Второй – это план тестирования: как оценить, было

ли изменение успешным? Третий – план отмены: как вернуть старую службу

или систему, если изменение не было успешным? Четвертый компонент – мо-

мент решения: как и когда нужно принимать решение о реализации плана от-

мены? Последний компонент – это подготовка: что вы можете сделать и прове-

рить заранее, чтобы убедиться, что изменение проходит плавно и в минимальные

сроки?

Важно заранее решить, при каких условиях обновление системы будет полно-

стью остановлено. Остановка должна обеспечить достаточное время для реали-

зации плана отмены до того, как служба снова должна начать работать. Время,

к которому служба снова должна начать работать, может быть основано на

обязательстве перед пользователями, потребностях бизнеса или обусловлено

тем, что данное изменение является элементом большей последовательности

изменений, которые будут затронуты, если служба не будет вовремя восстанов-

лена.

Точка принятия решения часто является самым сложным элементом для сис-

темного администратора, который вносит изменение. Мы часто считаем, что

можем потратить «еще только 5 минут», чтобы все заработало. Часто полезно,

чтобы другой системный администратор или руководитель разделял вашу от-

ветственность и обеспечивал, чтобы план отмены был реализован по графику,

если изменение было неудачным.

В идеальном случае лучше всего внести и проверить изменение заблаговремен-

но в тестовой лаборатории. Также может быть возможность заранее внести из-

менение на дополнительной машине, которой можно заменить нужную маши-

ну. Однако тестовые лаборатории и дополнительные машины – это роскошь,

которую не все компании могут себе позволить, а некоторые изменения непри-

годны для тестирования в лабораторной среде.

**17.1.2. Структура распространения информации**

Распространение информации об изменении имеет два аспекта: обеспечение

того, что вся группа системного администрирования знает о происходящем,

и того, что об этом знают ваши пользователи. Когда каждый в группе хорошо

информирован об изменениях, все системные администраторы смогут внима-

тельно следить за проблемами, которые могут быть вызваны изменениями.

Любые проблемы будут обнаружены раньше и смогут быть быстрее устранены.

Вам также потребуется разработать структуру распространения информации

для информирования ваших пользователей об изменениях, которые вы вноси-

те. Если изменения предполагают резкий переход, после которого старая служ-

ба, система или программа больше не будет доступна, вы должны обеспечить,

чтобы все ваши пользователи, работающие со старой версией, смогли продол-

жить работу с новой версией. Если предполагается плавный переход и старая

версия некоторое время будет доступна, вы должны обеспечить, чтобы все за-

ранее знали, когда это произойдет, как им воспользоваться старой версией при

необходимости и когда старая версия будет недоступна, если будет. Если вы

добавляете службу, вы должны обеспечить, чтобы все люди, которые ее запра-

шивали, а также те, кому она может быть полезна, знали, как ею пользоваться,

когда она будет доступной. Во всех трех случаях информируйте своих пользо-

вателей, когда работа будет успешно завершена и как сообщать о любых воз-

можных проблемах.

Несмотря на то что информировать об изменениях и графике их реализации

пользователей, чья работа может быть ими затронута, необходимо и правильно,

вы должны постараться не заваливать пользователей слишком большим коли-

чеством сообщений. Если вы так поступите, пользователи их проигнорируют,

посчитав неважными. Выбор нужных групп для сообщения о каждой службе

требует понимания своей пользовательской базы и своих служб. Например,

если вы знаете, что группа A пользуется службами A–K, а группа B – службами

B, D и L–P, то вам нужно сообщить об изменениях в службе A только группе A,

но об изменениях в службе B нужно проинформировать обе группы. Эта задача

может показаться трудной, но ее правильное выполнение очень существенно

для ваших пользователей.

Наиболее эффективный метод распространения информации отличается в раз-

ных компаниях и зависит от культуры компании. Например, в некоторых

компаниях самым эффективным средством может быть новостная группа, на

которую люди подписываются, где можно быстро просматривать заголовки

сообщений на наличие важных тем. Однако другие компании могут не пользо-

ваться новостными группами, поэтому электронная почта может быть более

эффективной. В случае значительных изменений мы рекомендуем отправлять

людям сообщение («толкать»), а не требовать, чтобы ваши пользователи каждые

несколько дней проверяли определенную веб-страницу («тянуть»). Значитель-

ное изменение, как рассмотрено в разделе 17.1.3, – это крупномасштабное или

критическое изменение.

**17.1.3. Составление графика**

Распределение времени – ключевой компонент управления изменениями. Вре-

мя внесения изменения может быть значительным фактором, влияющим на

ваших пользователей. Мы кратко рассмотрим три типа изменений: штатные,

критические и крупномасштабные обновления.

Штатное обновление может произойти в любое время и в основном незаметно

для большей части пользовательской базы. Эти изменения происходят все вре-

мя: обновление содержимого сервера директорий или базы данных аутентифи-

кации, помощь отдельному пользователю в настройке его системы, устранение

проблемы с рабочей станцией или принтером либо изменение скрипта, обраба-

тывающего файлы логов для предоставления статистических данных. Вам не

нужно планировать график штатного обновления; масштаб проблемы, которую

может вызвать ошибка, очень ограничен в силу сущности задачи.

Крупномасштабные обновления затрагивают многие системы или требуют серьезного нарушения работы системы, сети либо службы. Что считается крупно-

масштабным – зависит от компании. Для большинства компаний все, что затра-

гивает 30% систем или более, является крупномасштабным обновлением.

Крупномасштабные изменения включают модернизацию системы аутентифи-

кации, изменение инфраструктуры электронной почты или печати либо модер-

низацию базовой сетевой инфраструктуры. График этих изменений должен быть

тщательно согласован с пользовательской базой при помощи механизма «тол-

кания», например электронной почты. Крупномасштабные обновления не

должны проходить все время. Если это происходит, проверьте, не стоит ли вам

изменить классификацию тех или иных обновлений. В некоторых компаниях

хотят, чтобы эти обновления происходили вне временных интервалов пиковой

нагрузки, а в других может требоваться, чтобы все крупномасштабные обновле-

ния происходили в течение одного технического перерыва (см. главу 20).

Критические обновления могут не показаться масштабными или даже особен-

но заметными для ваших пользователей, но способны вызвать серьезное нару-

шение работы системы в случае проблемы. Критические обновления включают

изменение конфигурации маршрутизаторов, глобальных политик доступа,

конфигурации межсетевых экранов или внесение изменений в критический

сервер. У вас должен быть определенный способ заблаговременного сообщения

вашим пользователям о критическом обновлении на случай проблем. Эти об-

новления будут происходить достаточно часто, и вам не требуется распростра-

нять слишком много информации, поэтому допустим механизм «вытягивания»,

например веб-страница или новостная группа. Службе поддержки нужно сооб-

щить об изменении, проблемах, которые оно может вызвать, времени начала

и окончания работы и с кем связаться в случае проблемы.

Критические обновления должны происходить вне интервалов максимальной

загрузки, чтобы предоставить вам время на обнаружение и устранение любых

проблем, прежде чем они затронут ваших пользователей. Время пиковой загруз-

ки может различаться в зависимости от того, кто ваши пользователи. Если вы

работаете в компании электронной коммерции, чей сайт используется

в основном физическими лицами по вечерам или в выходные, лучшим временем

для внесения изменений может быть 9 часов утра. Кроме того, человек, который

внес критические изменения, не должен сразу уходить с работы домой после их

внесения. Если вы делаете критическое изменение, подождите пару часов – ваше

присутствие может понадобиться для устранения возможных проблем.

Во многих компаниях предпочитают вносить изменения в течение рабочего дня

по той же причине, по которой стараются не делать это в пятницу: желательно,

чтобы после изменения люди были на месте и смогли заметить и устранить

любые проблемы. Однако во многих компаниях для внесения изменений, тре-

бующих отключения системы, предпочитают дождаться, чтобы на месте не

осталось никого или почти никого.

Вам потребуется выяснить, что правильно в вашем случае. Однако вам следует

стараться избегать ситуаций, при которых люди вне организации системных

администраторов могут внести изменения, способные негативно повлиять на

важные системы.

Разные люди имеют различные взгляды на внесение критических изменений.

Более старшие системные администраторы обычно осторожнее младших,

поскольку лучше понимают возможное влияние изменений и научены горь-

ким опытом недостаточной осторожности в прошлом. В развитой организации

системных администраторов каждый будет знать о документированных,

последовательных инструкциях, которые надо соблюдать. Эти инструкции

будут включать приемлемое время для внесения определенных типов изме-

нений.

Пытаясь классифицировать обновления как штатные, критические или круп-

номасштабные, учитывайте, что некоторые изменения могут считаться штат-

ными в одной компании и критическими в другой и даже в различных подраз-

делениях одной компании. Например, в компании электронной коммерции

подключение нового узла к корпоративной сети может считаться штатным

обновлением, а подключение нового узла к сервисной сети, видимой пользова-

телям, – критическим. Определите, как должны классифицироваться различ-

ные виды обновлений в разных областях вашей компании, и создайте методику

построения графика, которая отражает ваше решение.

Полезно определить время запрета на изменения, когда можно осуществлять

только незначительные изменения. Запрет на изменения обычно накладывает-

ся в конце квартала и в конце финансового года. На рис. 17.1 изображен пример

сообщения о запрете на изменения одной компании, которое было отправлено

всем системным администраторам и начальникам отделов.

**17.1.4. Процессы и документация**

Процессы и документация являются важными элементами управления изме-

нениями. Соблюдение процессов и создание документации заставляет системных

администраторов тщательно готовиться к значительным изменениям. Систем-

ным администраторам нужно заполнить формы контроля изменения, или

предложения изменения, где подробно описываются изменения, которые они

будут вносить, затрагиваемые системы и службы, причины изменения, риски,

процедура проверки, план отмены, время реализации изменения и время реа-

лизации плана отмены. Иногда системным администраторам требуется пере-

числить все команды, которые они будут вводить. Необходимый уровень дета-

лизации различается в разных компаниях и обычно зависит от того, насколько

важна затрагиваемая машина или служба. Для очень важных машин системный

администратор не может вводить ничего, что не перечислено в одобренной фор-

ме контроля изменения. Однако для менее важных машин требования по соб-

людению процессов и документации должны быть менее жесткими, иначе

системные администраторы будут ощущать себя связанными ограничениями

управления изменениями и не смогут эффективно работать.

Если компания может определить одну или две машины, которые являются

критически важными для ведения бизнеса, эти машины должны быть защище-

ны жесткими процессами управления изменениями. Например, в компании

электронной коммерции в эту категорию должны входить машины главных баз

данных и машины, обрабатывающие данные о кредитных картах. В фармацев-

тических компаниях строгие требования по управлению изменениями для

машин, вовлеченных в процесс разработки лекарств, часто устанавливаются

законодательством. Машины в этой категории обычно не являются серверами,

которые обеспечивают такие важные службы, как электронная почта, печать,

DNS или аутентификация. Машины, которые обеспечивают эти службы, долж-

ны быть защищены менее строгими политиками управления изменениями,

чтобы найти равновесие между пользой от управления изменениями и от спо-

собности системных администраторов быстро ответить на требования пользо-

вателей. Однако значительные изменения в этих серверах или службах должны

выполняться с соблюдением хорошего процесса управления изменениями,

чтобы они проходили плавно, с минимальным количеством неожиданностей.

В последующих главах подробно рассмотрены процессы, связанные с различны-

ми типами изменений. В частности, в главе 18 рассмотрена модернизация сер-

веров, в главе 19 – изменение служб, а в главе 20 – технические перерывы.

**17.1.5. Технические аспекты**

Вам может потребоваться документированная процедура обновления системных

файлов конфигурации, которую будет соблюдать каждый системный админис-

тратор. Эта процедура должна последовательно применяться везде, где обнов-

ляются файлы конфигурации. Она должна быть точно документирована по

шагам, включая процедуру, которую нужно соблюдать, если любой из шагов

будет неудачным, и должна выдаваться всем системным администраторам при

вступлении в группу. Эта процедура должна включать создание историй обнов-

ления, блокировку файлов конфигурации, чтобы только один человек мог од-

новременно их редактировать, запуск автоматизированных проверок формата

информации в файлах и, если это допустимо, уведомление других систем или

приложений о том, что произошло обновление. Это принципиально полезный

метод, который должен всегда применяться всеми. Он простой и на удивление

часто может сэкономить время.

**17.1.5.1. История изменений и блокировка**

История изменений позволяет любому человеку, обладающему соответствую-

щим доступом, просматривать изменения, внесенные в файл, что очень полезно,

если текущая версия становится поврежденной. Обычно в истории изменений

также записывается, кто и когда внес изменение, и можно добавлять дополни-

тельный комментарий к изменению. Кроме того, программы контроля версий

обычно предоставляют механизм блокировки, который необходимо использо-

вать для предотвращения одновременного изменения одного и того же файла

конфигурации двумя людьми.

Прикрепление идентификатора каждого человека к его изменениям полезно.

Если младший системный администратор сделает ошибку, то старший сотруд-

ник, который это обнаружит, может позвать этого человека и воспользоваться

возможностью немного обучить его в данной области системы.

Для выполнения этих функций вам нужно обратить внимание на системы кон-

троля исходного кода, используемые разработчиками программного обеспече-

ния.

В хорошем справочнике (например, Bolinger 1995) объясняются более сложные

вопросы, в частности отмена изменения. Создайте простой текстовый файл

для экспериментов, пока изучаете инструкцию. Вы очень скоро станете экс-

пертом.

**17.1.5.2. Автоматизированные проверки**

Последние этапы обновления файла или набора файлов – проверка того, что

каждый файл не содержит синтаксических ошибок, а затем обеспечение того,

чтобы все приложения, использующие этот файл, начали использовать новую

информацию. Эти этапы должны выполняться автоматизированной програм-

мой, которая также обязана сообщать различным серверам, что их файлы кон-

фигурации были изменены, или при необходимости распространять файлы по

другим местам.

Иногда вам может потребоваться разделить эти два этапа. Если для того, чтобы

приложение начало использовать новую информацию о конфигурации, потре-

буется небольшое прерывание обслуживания и если обновление может подож-

дать до момента, когда оно вызовет меньшие нарушения работы или не вызовет

их совсем, синтаксис нужно проверить немедленно, а процесс обновления пе-

ренести на более поздний срок.

Конфигурации некоторых систем трудно проверить автоматизированной про-

граммой, и в идеале они должны создаваться программой, чтобы, по крайней

мере, не содержать синтаксических ошибок. Создайте какой-нибудь другой

способ тестирования этих компонентов, который будет гарантировать вам до-

статочно высокий уровень уверенности в том, что они работают правильно.

Например, под UNIX скрипты загрузки системы часто корректируются вруч-

ную, чтобы изменить набор служб, запускаемых во время загрузки, или, воз-

можно, чтобы изменить режим работы сетевых интерфейсов. Важно, чтобы эти

скрипты внимательно проверялись, потому что ошибка может не позволить

системе завершить цикл перезагрузки. В большинстве коммерческих UNIX-

систем загрузочные скрипты разделены на множество небольших элементов,

по одному на каждую службу, чтобы каждый из них можно было проверить

индивидуально и обеспечить достаточную степень уверенности в том, что изме-

нения и дополнения правильны.

Если скрипты загрузки не проверить заранее, проблемы с ними не будут обна-

ружены до следующей перезагрузки. Поэтому очень важно обеспечить полную

надежность загрузочных скриптов. Машины перезагружаются в самое неудоб-

ное время. Сбои систем происходят поздно вечером, во время вашего отпуска

и т. д. Если вы только при следующей перезагрузке обнаружите, что скрипт,

написанный вами, содержит опечатку, это случится в очень неудобное время.

Что еще хуже, хорошие системы могут работать без перезагрузки в течение

нескольких месяцев. Очень трудно запомнить, какие изменения были внесены

после последней перезагрузки, особенно если она была несколько месяцев назад.

Даже если в компании есть политика записи изменений в журнале событий или

системе заявок, может быть очень трудно найти нужное изменение, если прошло

несколько месяцев.

Если ваши пользователи предупреждены о том, что конкретный узел будет

недоступен в определенное время, используйте эту возможность для дополни-

тельной перезагрузки, когда работа будет закончена.

***17.2. Тонкости***

Как только у вас будет базовая структура управления изменениями, которая

описывает процесс обновления конфигурации, используемые методы распро-

странения информации и составление графика изменений, вы сможете восполь-

зоваться некоторыми другими методами управления изменениями для повы-

шения стабильности вашей компании. В частности, вы можете создать автома-

тизированные интерфейсы для распространенных изменений конфигурации,

которые осуществляют всю блокировку, ведение истории изменений, проверку

и обновление за системных администраторов. Вам также нужно устраивать

формальные собрания по управлению изменениями с многосторонними кон-

сультациями, чтобы рассматривать предложения об изменениях.

**17.2.1. Автоматизированные интерфейсы**

Автоматическая проверка системных файлов на ошибки формата и синтаксиса

перед утверждением изменения обеспечивает для ваших систем более серьезную

стабильность. Следующий шаг по этому пути – создать для этих системных

файлов оконечный интерфейс, который задает соответствующие вопросы, про-

веряет ответы на ошибки, ищет пропуски и затем правильно обновляет файл,

используя предоставленную информацию. Если каждый станет пользоваться

этим интерфейсом, будет только одно место, требующее проверки ошибок.

**17.2.2. Собрания по вопросам управления изменениями**

Официально одобренные собрания по вопросам управления изменениями для

рассмотрения, обсуждения и составления графика предлагаемых изменений

являются ценным средством повышения стабильности систем. Это формальный

процесс отчета системных администраторов, что и когда они планируют делать,

сколько времени это займет, что может не получиться, как это будет проверять-

ся, как отменить изменение и сколько времени займет отмена. Он заставляет

системных администраторов думать о последствиях того, что они делают,

а также приготовиться к возможным проблемам.

Он также предупреждает об изменениях других людей, чтобы они могли быть

в курсе потенциального источника проблем. В число людей, которые одобряют,

отклоняют или переносят предлагаемые изменения, должны входить сотруд-

ники из всех подразделений компании, чтобы представители каждой области,

которая может быть затронута, могли предупредить свои группы и подготовить

их к грядущим изменениям. Участники таких собраний называются заинтере-

сованными лицами.

Эти собрания предоставляют заинтересованным лицам общий обзор того, что

происходит в компании. Они обеспечивают старшим системным администра-

торам и руководителям возможность заметить изменение, которое вызовет

проблемы, до того, как оно будет внесено, и не позволить этому случиться. Они

снижают неупорядоченность и обеспечивают более стабильную среду. Обычно

собрания по управлению изменениями проходят раз в неделю или раз в месяц

в соответствии с частотой изменений в компании.

Одобрение каждого изменения всеми заинтересованными лицами не только

повышает стабильность системы, но и предоставляет системным администра-

тором своего рода «прикрытие». Мы не считаем, что это цинично. Если группы

пользователей постоянно жалуются на то, что процесс выходит из-под контро-

ля и сбои происходят в неудобное время, хорошим решением является прове-

дение консультации по пересмотру изменения, чтобы вовлечь в это самих

пользователей и получить их одобрение на изменения.

Развитый процесс управления изменениями также может приобрести черты

управления проектами, при этом предлагаемое изменение будет внимательно

изучаться в плане влияния не только на другие системы, но и на сроки выпол-

нения других задач группы. Если внесение изменения вызовет задержку вы-

полнения других, более важных проектов, оно будет отклонено.

**17.2.3. Упрощение процесса**

В конце концов, когда вы посчитаете, что все на месте, вам стоит оценить свой

процесс с точки зрения возможности упростить его. Есть ли неиспользуемые

вопросы в вашей форме предложения изменения? Можно ли более эффектив-

но выполнять элементы процесса, связанные с бумажной работой? Если

формы сетевые, есть ли возможность сохранения каждым пользователем

значений по умолчанию в некоторых полях, например имени и контактной

информации? Какие проблемы есть у людей, использующих систему в ее

нынешнем состоянии?

***17.3. Заключение***

Управление изменениями – это ценное средство, которое используется в разви-

тых учреждениях для повышения надежности системы за счет как ограничения

на внесение определенных изменений, так и наличия процесса заблаговремен-

ного рассмотрения изменений, позволяющего выявить любое негативное вли-

яние, которое мог упустить системный администратор, или взаимодействие,

о котором он мог не знать. Кроме того, управление изменениями помогает

с устранением проблем, потому что изменения отслеживаются и в случае появ-

ления проблем могут быть пересмотрены.

Частота, с которой нужно устраивать собрания по управлению изменениями,

зависит от их охвата и частоты изменений среды, которую они затрагивают.

Создание механизма, при помощи которого системные администраторы могут

проверить, работает ли система нормально, прежде чем вносить свои изменения,

снижает риск того, что изменение, внесенное в процессе устранения существу-

ющей проблемы, усложнит процесс отладки или сделает систему даже менее

стабильной.

***Глава 18***

***Обновления серверов***

Данная глава имеет очень специфичную тему: обновление операционной систе-

мы отдельного узла. Эта задача, хотя и выглядит обманчиво простой, на самом

деле требует большого объема предварительной подготовки и последующего

тестирования. Само по себе обновление может осуществляться различными

способами. Чем более критичен узел, тем важнее правильность выполнения

обновления. Этот метод является строительным блоком. После его детального

изучения вы можете перейти к более крупным проектам по обновлению, рас-

смотренным в главе 20.

Для успешного выполнения этой задачи требуется единственное средство, вне

зависимости от типа операционной системы. Это средство – лист бумаги, который

будет использоваться для создания контрольного списка. Использование этого

средства обязательно.

Некоторые люди предпочитают имитировать лист бумаги при помощи веб-стра-

ницы, википедии или электронной таблицы. Такие высокотехнологичные ре-

шения имеют преимущества, которые будут рассмотрены позднее. Однако

принцип один: обновлять сервер без контрольного списка недопустимо. Возь-

мите карандаш, давайте начнем.

***18.1. Основы***

Принципиальная задача любого обновления ОС состоит в том, чтобы как мини-

мум все службы, которые предоставлялись до обновления, работали после об-

новления. Обновление может проводиться для того, чтобы расширить функциональность или надежность, но оно не должно их снижать. Учитывая это,

процесс имеет следующую структуру.

1. Составьте контрольный список служб:

a. Какие службы предоставлялись сервером?

b. Кто является пользователем каждой службы?

c. Какие программы предоставляли каждую службу?

2. Проверьте, чтобы каждая программа работала с новой ОС, или заплани-

руйте путь обновления программного обеспечения.

3. Для каждой службы разработайте тест на проверку работоспособности.

4. Напишите план отмены с конкретными условиями.

5. Выберите технический перерыв.

6. Объявите об обновлении в необходимом порядке.

7. Выполните ранее разработанные тесты, чтобы убедиться, что они действенны.

8. Заблокируйте пользователей.

9. Проведите обновление с наблюдением/помощью (или под руководством)

другого человека.

10. Повторите все ранее разработанные тесты. Соблюдайте стандартный про-

цесс отладки.

11. Если тесты будут неудачными или произойдут другие события, которые

являются условиями для выполнения плана отмены, выполните план от-

мены.

12. Разблокируйте пользователей.

13. Сообщите пользователям о завершении/отмене обновления.

14. Проанализируйте, что прошло правильно, а что нет, измените контроль-

ный список в соответствии с приобретенным опытом.

А вы думали, что нужно всего лишь воспользоваться установочным диском, не

так ли? Давайте более подробно рассмотрим каждый этап.

18.1.1. Этап 1: составьте контрольный список служб

Контрольный список служб – это средство, которым вы будете пользоваться для

проведения всей процедуры. Список должен отражать, какие службы предо-

ставляются узлом, кто пользуется каждой службой и какая программа предо-

ставляет каждую службу.

Электронные таблицы – отличный способ представления такой информации.

Самым большим преимуществом представления этой информации в электрон-

ном виде является то, что ее смогут легко совместно использовать персонал

и пользователи. Лучше предоставить доступ к файлу через Интернет, чем от-

правлять его каждому человеку, потому что версию в Сети можно быстро обно-

вить. Люди всегда будут видеть самые последние обновления1. Однако Сеть

предполагает наличие механизмов активной доставки. Люди не будут искать

файл сами. Вы можете включать URL в каждое электронное письмо, касающе-

еся проекта, но это не даст гарантии, что его прочтут. Правильная идея – объ-

являть о любых значительных обновлениях.

Дважды проверьте свои планы, устройте собрание ключевых представителей

затрагиваемого сообщества. Покажите им план, шаг за шагом, попросите их

проверить ваши предположения. Наиболее эффективно начать процесс с собра-

ния, а затем пользоваться для обновления электронной почтой, устраивая

другие личные встречи, возможно, только в ключевые моменты процесса.

1 Явно укажите в контрольном списке номер версии и дату, чтобы каждый мог

легко проверить, является ли его версия последней.

Проверка зависимости пользователей

Один системный администратор устроил собрание десяти опытных сис-

темных администраторов, каждый из которых ознакомился с планом

и сразу подтвердил, что он не возражает. Когда системный администратор

начал разбирать его поэтапно, задавая конкретные вопросы, например:

«Что будет, если мы это выключим?», они начали говорить: «Ой, нет,

если вы это сделаете, биллинговая система не будет работать. Я думаю,

нам нужно добавить этап, на котором мы перенесем биллинговую инфор-

мацию». В результате получился совершенно другой план, в котором

было в три раза больше этапов. Если бы не было личной встречи систем-

ных администраторов, на которой был рассмотрен каждый этап в отде-

льности, первоначальный план вызвал бы масштабную катастрофу.

Включение пользователей в процессы принятия решений и планирования дает

им ощущение контроля и участия. Пользователи вкладываются в результат

и становятся частью команды, что обычно ведет к более позитивным впечатле-

ниям и лучшим отношениям между системными администраторами и подраз-

делениями бизнеса. Совместный с пользователями доступ к информации о за-

висимости и состоянии через Сеть и по электронной почте позволяет поддержи-

вать рабочие связи.

Машина может быть выделена для предоставления одной службы или предо-

ставлять много служб. В любом случае предоставление службы в целом может

обеспечиваться несколькими программами.

Что находится на машине?

Иногда вы точно знаете, для чего используется машина, и первоначаль-

ный контрольный список обновления создать легко. Однако со временем

на машину добавляются дополнительные службы, функции и программы

(Evard 1997). Мы можем дополнительно проконтролировать себя, если

проверим сам узел. Вы можете просмотреть программы, установленные

под UNIX, в директориях /opt, /usr/local и других местах, общих для

таких систем. Операционные системы Майкрософт обычно размещают

программы в папке под названием Program Files, хотя некоторые исполь-

зуют собственные правила, например устанавливают по умолчанию

папку C:\apps. Вы можете посмотреть, какие процессы запущены в сис-

теме. UNIX- и NT-системы выводят все прослушиваемые порты TCP/IP

и UDP/IP по команде netstat –an. В UNIX есть различные загрузочные

скрипты, которые можно проанализировать. В NT есть консоль Службы.

В UNIX есть файлы crontab, которые можно просмотреть. В каждой ОС

есть по крайней мере один способ перечислить все установленные про-

граммы. Некоторые примеры таких средств – это pkginfo (Solaris

и SVR4), swlist (HP-UX 10 и выше) и ‘rpmqa’ (Linux).

Глава 18. Обновления серверов

445

Обычно каждая служба напрямую связана с одной программой. Иногда служба

связана с несколькими программами, например сервер календаря, который

использует сервер LDAP. Зафиксируйте все такие взаимозависимости в конт-

рольном списке.

Кроме того, важно определить ключевых пользователей различных служб.

Пользователи могут обращаться к службе напрямую или косвенно, взаимодействуя со службами, которые используют другие службы для получения или ввода

данных. Людей нужно привлечь к процессу или, по крайней мере, уведомить

о том, что происходит обновление. Если от служб зависят другие машины, надо

включить в процесс пользователей этих машин.

Часто вы будете находить службы без прямых или косвенных пользователей,

и такие службы можно отключить. Это всегда приятно, но будьте осторожны:

вы можете найти зависимость тогда, когда службы уже не будет. Оставьте служ-

бу в готовом к запуску, но приостановленном состоянии, чтобы при необходи-

мости ее можно было восстановить. Убедитесь, что вы отметили, почему служ-

ба существует, но не работает, чтобы в следующий раз ее можно было удалить,

если к тому времени она не будет вновь включена. Лучшее место для такой до-

кументации – один из файлов конфигурации, который будет редактироваться

при возобновлении служб.

**18.1.2. Этап 2: проверьте совместимость программ**

Следующий этап – убедиться, что каждая программа сможет работать с новой

ОС, и запланировать способ обновления для тех программ, которые не будут

работать. Используя список, созданный ранее, свяжитесь с разработчиками

и узнайте, будет ли работать версия программы после обновления. Часто разра-

ботчики предоставляют такую информацию на своих веб-сайтах.

Вы можете захотеть сами проверить достоверность информации или найти

другого пользователя, который уже сделал обновление. Представления разра-

ботчиков о том, что значит «версия будет работать», часто не включают функ-

ции, которые нужны в вашей компании, или точную конфигурацию, которую

вы определите. Самостоятельное тестирование может быть дорогим, но, скорее

всего, оно окажется дешевле, чем неудачное обновление, и при этом снижается

риск неудачи. Целесообразность определяется из соображений управления

риском. Если обновляется только одна система и приложение не является кри-

тически важным, его персональная проверка может быть пустой тратой време-

ни. Если обновление автоматизировано и будет повторяться тысячи раз и по-

тенциальная ошибка будет сильно заметна, тестирование необходимо.

Если используемая версия программы будет работать с новой версией ОС, за-

фиксируйте, где вы нашли эту информацию, для справки в будущем. Если

программа не поддерживается новой ОС, у вас есть несколько вариантов.

Обновление до версии, которая поддерживается обеими ОС. Если вам пове-

зет, то программу можно обновить до версии, которая работает как в ны-

нешней, так и в будущей системе. Если это возможно, запланируйте обнов-

ление до этой версии до обновления ОС. Здесь могут быть полезны тесты,

разработанные на этапе 3.

Обновление возможно, но работает только в новой ОС. В данном случае вы

должны запланировать обновление программы после завершения обновле-

•

•

ния ОС. В зависимости от требований пользователей, это обновление либо

может быть элементом обновления ОС, либо можно договориться о переры-

ве в работе службы, если пользователям не нужен непрерывный доступ. На-

пример, если узел – загруженный веб-сервер, пользователи могут потребо-

вать немедленной установки нового программного обеспечения, потому что

это основная функция узла. Однако, если требуется обновить редко исполь-

зуемый компилятор, пользователи могут просто попросить, чтобы он был

обновлен на следующей неделе или перед завершением определенного цик-

ла разработки. Это особенно справедливо, если пока для компиляции мож-

но пользоваться другим узлом.

Продукт больше не поддерживается. Иногда мы только при обновлении

операционной системы узнаем, что продукт больше не поддерживается раз-

работчиком. Это может заблокировать обновление, либо пользователи мо-

гут пожелать сменить поставщиков или отказаться от этого продукта.

**18.1.3. Этап 3: тесты для проверки**

После того как будет определена каждая служба, нужно разработать тесты,

которые будут использоваться для проверки того, что служба правильно рабо-

тает после обновления. Лучший сценарий – записать все тесты в виде скриптов,

которые могут быть запущены автоматически. Можно создать общий скрипт,

который выводит сообщение «OK» или «FAIL» (Неудачное завершение) для

каждого теста. Затем можно запускать тесты отдельно по мере устранения кон-

кретных проблем. Для более сложных служб пользователи могут писать тесты

или, по крайней мере, просматривать их либо предложить, чтобы их вызвали

для выполнения вручную их собственных тестов. Некоторые программы имеют

средства тестирования установки, которые могут быть запущены для проверки.

Иногда эти средства проверки недоступны пользователям, но их можно получить

через представителя разработчика.

Процедуры проверки программ

Все программные пакеты должны иметь процедуры проверки, но так

редко бывает на самом деле. Иногда такую процедуру лучше написать

самому. Тесты могут простыми, такими как проверка компилятора при

помощи компиляции программы Hello, World. Один тест гораздо лучше,

чем их полное отсутствие.

Иногда процедура проверки предоставляется, но на самом деле не рабо-

тает. Один из поставщиков суперкомпьютеров был известен наличием

плохих проверочных баз данных, особенно в бета-версиях ОС.

В обществе программистов для описания определенного способа проверки ис-

пользуется термин регрессивное тестирование. Вы сохраняете выходные данные

старой системы, вносите изменение, а затем сохраняете выходные данные новой

системы. Результаты должны точно соответствовать. Если ожидается, что новый

результат будет немного отличаться, вы можете отредактировать базовый ва-

риант вручную, чтобы он отражал ожидаемые изменения, либо воспользовать-

ся алгоритмом нечеткого соответствия. Для сравнения результатов можно

воспользоваться простыми средствами. Например, UNIX-программа diff – это

очень полезное средство, которое сравнивает два текстовых файла

и указывает на различия между ними1. Программа diff имеет ограниченные

возможности по оценке нечеткого соответствия, опция –w делает одинаковым

все незаполненное пространство. Более сложные средства регрессивного тести-

рования могут программироваться на игнорирование конкретных изменений,

обычно основанных на системе стандартных выражений. Однако такая слож-

ность необязательна. Вы можете вручную изменить старый результат – сначала

сделайте резервную копию! – чтобы отразить отличия, ожидаемые в новом ре-

зультате. Например, вы можете изменять номера версий, чтобы они соответствовали новым программам. Прекрасные примеры регрессивного тестирования

приведены в книге Кернигана и Пайка «The Practice of Programming» (Kernighan

and Pike 1999), как и процедура установки perl (посмотрите, как реализованы

тесты make tests).

Иногда тесты могут быть простыми, как при компиляции и запуске программы

Hello, world! для проверки работы компилятора. Это может быть определенная

последовательность команд или щелчков мышью, после которой можно посмот-

реть, отображается ли ожидаемый результат. Однако будьте внимательны,

чтобы такие тесты не оказались поверхностными.

Hello, World!

Однажды Том отвечал за поддержку большого количества компиляторов

для нескольких операционных систем. Он создал библиотеку простых

программ, большинство из них только выводило Hello, World! и заверша-

лось. Он всегда мог проверить, что новый установленный компилятор

был, по крайней мере, принципиально правильным, если компилирова-

лись и запускались соответствующие программы. При добавлении новых

языков программирования Том часто просил программистов написать

тестовую программу. Программистам нравилось, что их просят помочь!

Вы должны проверять тесты так же подробно, как любую другую службу. Вы

ведь не захотите, чтобы при запуске тестов были какие-то сомнения в том, за-

кончился ли тест неудачно из-за обновления или из-за ошибки в самом тесте.

Заманчиво выполнить эти тесты вручную. Однако помните, что каждый тест

будет выполнен как минимум три раза и даже больше, если возникнут пробле-

мы. В этом преимущество автоматизации тестов. Если тесты достаточно общие,

ими можно будет снова воспользоваться при дальнейших обновлениях. В конце

концов, ими можно пользоваться регулярно просто для устранения проблем

или в качестве средств мониторинга для обнаружения сбоев прежде, чем их

обнаружат ваши пользователи.

Автоматизированные тесты хорошо подходят для программ, которые выводят

предсказуемые текстовые данные, но их гораздо сложнее использовать для

графических программ, сетевых служб, например NFS, или для таких физиче1 Несмотря на то что diff впервые появилась в UNIX, есть порты практически на

каждую существующую ОС.

ских действий, как печать. В случае с NFS вы можете попытаться осуществить

доступ к файлу, а не проверять сам протокол. Тестирование сетевых служб,

имеющих простые текстовые протоколы, таких как электронная почта (SMTP,

POP, IMAP) или веб-службы (HTTP), может быть автоматизировано при помо-

щи простых скриптов, использующих средства типа netcat для отправки и по-

лучения текста протокола на соответствующий сетевой порт.

Для других программ и служб вы можете найти специализированные тестовые

системы, но они обычно очень дорогие. В таких случаях вам придется тестиро-

вать их вручную, документируя несколько ключевых функций для проверки

или последовательность операций для выполнения. Рано или поздно каждый

оказывается в такой ситуации, и все мы должны жаловаться разработчикам,

пока они не обеспечат возможность автоматизированного тестирования своих

продуктов.

Несмотря на то что автоматизировать все тесты предпочтительно, это не всегда

возможно. Некоторые тесты слишком трудно автоматизировать, либо они тре-

буют физического наблюдения. Даже если вы автоматизировали все тесты, то,

если вам покажется, что требуется дополнительное тестирование вручную,

выполните его. Иногда человеческий глаз видит то, что не может обнаружить

лучшая автоматизация.

Разработка через тестирование

TDD (Test-Driven Development) – это относительно новая тенденция

в отрасли. Раньше разработчики писали код, а затем создавали тесты,

чтобы его проверить (ну, на самом деле это было не так, редко у кого-то

было время создавать тесты). TDD – это обратный процесс. Сначала пи-

шутся тесты, а затем – код. Это обеспечивает создание тестов для всего

нового кода. Так как тесты выполняются автоматически, вы строите

структуру тестов, которая сохраняется с проектом. По мере развития

кода снижается риск того, что изменение нарушит функциональность

и это не будет замечено. Разработчики свободно могут переписывать, или

перестраивать, большие либо маленькие элементы кода, зная, что, если

они что-то нарушат, это будет сразу замечено. В результате программы

содержат меньше ошибок.

Тесты лучше комментариев в коде (документации), потому что коммен-

тарии часто устаревают и никто этого не замечает. Тесты, которые охва-

тывают все граничные условия, гораздо более подробны, чем любая до-

кументация. Тесты не устаревают, потому что они могут быть включены

как элемент процесса разработки, чтобы предупреждать разработчиков

об ошибках, которые те внесли в код.

Нам бы хотелось, чтобы и в области системного администрирования изу-

чали TDD и применяли такие методы.

Сохранение тестов для дальнейшего использования

Владелец крупного бизнеса хотел проверить 400 UNIX-серверов сразу

после полуночи 1 января 2000 года, чтобы убедиться в том, что основные

функции операционной системы и связанная с ними инфраструктура

работают правильно. Был создан ряд бесконтактных тестов, каждый

выводил ответ PASS/FAIL: работает ли система, можем ли мы войти

в систему, может ли она видеть NIS-серверы, правильно ли время, может

ли система разрешать DNS, может ли она подключаться к NFS-серверам

и читать файлы, работает ли автоматическое подключение разделов

и т. д. При помощи центральной системы администрирования тесты мог-

ли одновременно запускаться на нескольких системах, а результаты – цен-

трализованно собираться. Все 400 серверов были проверены в течение

20 мин, и персонал смог сообщить о прохождении тестов группе отсле-

живания проблемы Y2K раньше других, меньших по размеру подразде-

лений. Тесты приобрели такую популярность в группе системных адми-

нистраторов, что стали элементом ежедневного мониторинга среды.

Тестам нашлось другое применение. Скрытая ошибка в автоматическом

подключении разделов в Solaris 2.5.1 и 2.6 могла проявиться после серь-

езного сбоя сети, но лишь на нескольких случайных машинах. Запуск

этого средства тестирования определял затронутые машины после любо-

го сбоя.

**18.1.4. Этап 4: напишите план отмены**

Как вы вернетесь к предыдущему состоянию, если в ходе обновления что-то идет

не так? Как вы сможете «отменить» его? Сколько времени это займет? Очевидно,

мелкую неполадку можно попытаться исправить при помощи обычного процес-

са отладки. Однако вы можете потратить весь технологический перерыв – время,

выделенное на отключение системы, – пытаясь сделать что-нибудь, чтобы об-

новление заработало. Поэтому важно иметь конкретное время, в которое будет

задействован план отмены. Возьмите согласованное время окончания и вычти-

те время, которое потребуется на отмену, а также время, которое потребуется

для проверки того, что отмена завершена. Когда вы исчерпаете это время, вы

должны либо признать обновление успешным, либо начать свой план отмены.

Полезно, чтобы за временем следил кто-то, не входящий в группу, непосредственно выполняющую обновление, например руководитель. План отмены также

может быть задействован в случае неудачного выполнения одного или более

ключевых тестов либо непредвиденных ситуаций, связанных с обновлением.

В малых или средних системах перед началом обновления можно создать полную

резервную копию. Может быть, даже проще сделать точные копии дисков

и выполнять обновление на копиях. В случае серьезных проблем можно вновь

установить первоначальные диски. Крупные системы гораздо сложнее воспро-

извести. В данном случае может быть достаточно создать копии системных

дисков и постоянно делать резервные копии.

Обновление копии

Вопрос: Если вы собираетесь сделать точную копию жесткого диска перед

обновлением сервера, где нужно выполнять обновление – на копии или

на оригинале?

Ответ: Обновляйте копию. Если обновление пройдет неудачно, то вряд

ли вам будет приятно обнаружить, что копия была сделана неправильно.

Вы просто уничтожите оригинал. Мы видели это много раз.

Точное копирование дисков легко выполнить неправильно. Иногда дан-

ные копируются, а с загрузочным сектором возникает проблема и диск

не загружается, иногда данные копируются не полностью либо не копи-

руются совсем.

Чтобы избежать такой ситуации, загрузитесь с копии. Убедитесь, что

копия работает. Затем выполните обновление на копии.

**18.1.5. Этап 5: выберите технический перерыв**

Следующий этап – это проверка ваших технических и нетехнических навыков.

Вы должны согласовать со своими пользователями технический перерыв, то

есть время, когда будет проходить обновление. Для этого вы должны знать,

сколько времени займет процесс, и иметь план на случай неудачного обновления.

Это больше касается технических вопросов.

Когда? В ваше SLA должны быть включены положения о том, когда можно

осуществлять техническое обслуживание. Обычно пользователи хорошо

представляют, когда отключение пройдет для них безболезненно. Боль-

шинство систем бизнеса не нужны ночью или в выходные. Однако систем-

ные администраторы могут не захотеть работать в это время, а в определен-

ное время может быть недоступна поддержка разработчиков. Нужно найти

компромисс. Системы, которые должны работать в режиме 24/7, имеют

предусмотренный режимом план обслуживания, возможно, содержащий

резервные системы.

Сколько времени? Продолжительность технического перерыва равна вре-

мени, которое потребуется на обновление, сложенному с интервалами вре-

мени, необходимого для устранения проблем, для выполнения плана отме-

ны и для проверки того, что отмена сработала. Изначально лучше умно-

жить ваши оценки на два или на три, чтобы не переоценить свои возмож-

ности. Со временем ваши оценки станут более точными.

Вне зависимости от того, какую продолжительность вы подсчитали, объяви-

те, что перерыв будет гораздо дольше. Иногда вы можете начать работу

позже. Иногда она может потребовать больше времени, чем вы предполага-

ли, по техническим (оборудование, программы или несвязанные либо не-

предвиденные события) или нетехническим (погода либо автомобильные

пробки) причинам. Обратная сторона объявления большего перерыва заклю-

чается в том, что, если вы раньше закончите обновление и проверку, вам

всегда следует сообщать об этом пользователям.

Когда нужно задействовать план отмены? Хорошая идея – в явном виде

указать точное время, в которое будет задействован план отмены, в силу

причин, рассмотренных на этапе 4.

Скотти всегда преувеличивает

В серии «Relics» сериала Star Trek: Next Generation Джеймс Дуэн (James

Doohan) появляется в эпизодах в роли Скотти (Scotty) из оригинального

сериала. Одной из интересных находок Скотти было то, что он всегда

преувеличивал, когда сообщал свои предположения капитану Джеймсу

Т. Керку (Captain James T. Kirk). Таким образом, он всегда выглядел

чудесным работником, когда проблемы решались быстрее, чем предпо-

лагалось. Теперь мы знаем, почему гиперпространственный привод

всегда начинал работать раньше, чем предполагалось, а системы жизне-

обеспечения функционировали дольше, чем прогнозировалось. Посту-

пайте, как советует Скотти! Преувеличивайте свои оценки! Но не забы-

вайте соблюдать другой принцип Скотти – сразу же сообщать людям,

когда работа будет проверена и завершена.

Пример: карт-бланш в понедельник вечером

Когда Том работал в отделении Mentor Graphics, у системных админист-

раторов была такая роскошь, как еженедельный технический перерыв.

Вечер понедельника был карт-бланшем системных администраторов.

Ожидалось, что пользователи выйдут из системы к 18 часам и системные

администраторы смогут использовать этот вечер для выполнения любых

видов серьезных обновлений, которые потребуют отключения служб.

Каждый понедельник в 16 часов пользователям сообщали о том, какие

изменения произойдут и когда системами снова можно будет пользовать-

ся. В конце концов, пользователи выработали привычку планировать на

вечер понедельника дела, не связанные с работой. Ходили слухи, что

некоторые из них проводили время со своими семьями.

Несмотря на то что для одобрения такой практики руководством требо-

вались серьезные политические вложения, она была важным фактором

обеспечения высокой надежности сети отделения. Редко были причины

отменять своевременные обновления системы. О проблемах в течение

недели можно было позаботиться при помощи временных мер, но долго-

срочные меры эффективно принимались вечером в понедельник. В отли-

чие от некоторых организаций, где долгосрочные меры не принимались

никогда, в данном случае они выполнялись относительно быстро.

Когда работы было немного, один администратор настаивал на перезагрузке некоторых критически важных серверов в 18 часов, чтобы «под-

толкнуть» пользователей пойти вечером домой. Он верил, что это помо-

гало пользователям поддерживать привычку не планировать ничего

критически важного на вечер понедельника. Конечно, системные адми-

нистраторы придерживались гибкого подхода. Когда приближался срок

сдачи важного проекта и сотрудники работали круглосуточно, системные

администраторы отменяли технический перерыв вечером в понедельник

или согласовывали с пользователями, что можно отключить, не мешая

их работе.

**18.1.6. Этап 6: сообщите об обновлении**

в соответствии с установленным порядком

Теперь сообщите об обновлении пользователям. Используйте одинаковый фор-

мат для всех объявлений, чтобы пользователи к ним привыкли. В зависимости

от культуры вашей организации, сообщение может распространяться по элек-

тронной или голосовой почте, в виде бумажной записки, записи новостной

группы, веб-страницы, объявления на двери или дымовых сигналов. Вне зави-

симости от формата, сообщение должно быть кратким и по теме. Многие люди

читают только строку «Тема», поэтому составьте текст грамотно, как показано

на рис. 18.1.

Кому: Всем пользователям

Тема: ПЕРЕЗАГРУЗКА СЕРВЕРА СЕГОДНЯ В 18.00

От: Группа системного администрирования

Обратный адрес: tom@example.com

Дата: Четверг, 16 июня 2001

КОГО ЭТО ЗАТРОНЕТ:

Все узлы на DEVELOPER-NET, TOWNVILLE-NET и BROCCOLI-NET.

ЧТО ПРОИЗОЙДЕТ:

Все серверы будут перезагружены.

КОГДА?

Сегодня с 18 до 20 часов (должно занять 1 час).

ЗАЧЕМ?

Мы распространяем новые параметры настройки ядра по всем серверам.

Это требует перезагрузки. Риск минимален. Более подробную информацию можно найти на веб-странице

http://portal.example.com/sa/news0005.

Я ПРОТИВ!

Отправьте сообщение в службу поддержки, и мы попытаемся изменить

время. Пожалуйста, назовите имя сервера, который вы хотите нас

попросить не перезагружать сегодня.

Рис. 18.1. Образец сообщения об обновлении

Лучше иметь пустой шаблон, который нужно будет каждый раз заполнять, чем

редактировать предыдущие объявления для включения новой информации. Это

предотвращает форму от изменения со временем. Это также предотвращает

распространенную проблему того, что некоторые части текста забывают заме-

нить. Например, при подготовке рис. 18.1 мы первоначально использовали

настоящее объявление о перезагрузке маршрутизатора. Мы изменили его на

сообщение о серверах, но забыли отредактировать строку «Тема». Пример про-

Глава 18. Обновления серверов

453

шел через четыре корректуры, прежде чем кто-то это заметил. Если бы мы на-

чали с пустого шаблона, то этого бы не произошло.

18.1.7. Этап 7: выполните тесты

Прямо перед началом обновления выполните тесты. Такая проверка в последний

момент позволит вам убедиться, что вы не будете после обновления отслеживать

проблемы, которые существовали еще до него. Представьте ужас от выполнения

плана отмены только для того, чтобы узнать, что неудачный тест все равно не

выполняется.

18.1.8. Этап 8: заблокируйте пользователей

Обычно лучше позволить пользователям выйти из системы самостоятельно,

чем выбросить их путем перезагрузки или отключения службы. В различных

службах для этого есть разные способы. Используйте доступные в ОС средства,

чтобы предотвратить вход в систему во время технического перерыва. Многие

пользователи делают попытку входа в систему или доступа к ресурсу в целях

собственной проверки обновления. После успешной попытки пользователь

считает, что система доступна для нормальной работы, даже если об этом не

было объявлено. Поэтому важно заблокировать пользователей на время тех-

нического перерыва.

18.1.9. Этап 9: выполните обновление

под чьимнибудь наблюдением

С этого начинается большинство книг для системных администраторов. Разве

вы не рады, что купили именно эту книгу?

Теперь пришел момент, которого вы все ждали: выполните обновление в соот-

ветствии с вашими местными процедурами. Вставьте DVD, перезагрузитесь или

сделайте что-то еще.

Обновления системы слишком важны, чтобы выполнять их в одиночку. Во-

первых, все мы делаем ошибки и вторая пара глаз всегда полезна. Обновления

выполняются не каждый день, поэтому кому угодно может не хватать опыта.

Во-вторых, когда два человека вместе обновляют систему, происходит уникаль-

ное обучение. Системные обновления часто требуют максимального использо-

вания наших технических знаний. Мы используем команды, знания и, возмож-

но, даже части нашего мозга, которые не задействованы в другое время. Вы

можете многому научиться, если посмотрите и поймете приемы, которые кто-то

будет применять в это время. Метод совместной разработки, или так называе-

мого парного программирования, становится все более популярным и предпо-

лагает, что разработчики работают в парах и набирают код по очереди. Это еще

один метод разработки, использование которого может принести пользу сис-

темным администраторам.

Если обновление окажется неудачным, то никогда нелишне обратиться за помо-

щью к коллеге или к старшему сотруднику вашего подразделения. Вторая пара

глаз часто творит чудеса, и никому не должно быть стыдно просить помощи.

Основы

454

18.1.10. Этап 10: проверьте свою работу

Теперь повторите все ранее созданные тесты. В случае их невыполнения действуйте в соответствии с обычным процессом отладки. Тесты можно повторять

снова и снова по мере проведения отладки. Вполне естественно снова запускать

невыполненный тест после каждой попытки устранить проблему. Однако убе-

дитесь, что вы запустили все тесты, прежде чем объявить об успехе обновления,

так как многие процессы серверов взаимосвязаны. Исправление неполадки,

которая вызывала невыполнение теста, может привести к невыполнению дру-

гого теста, который раньше выполнялся.

На данном этапе нужно привлекать пользователей. Как и в случае с моделью

службы поддержки в главе 14, работа не может считаться выполненной, пока

пользователи не проверили, что все завершено. Это может означать, что нужно

позвать пользователей в заранее назначенное время либо пользователи могут

согласиться сообщить вам на следующий день после завершения технического

перерыва. В этом случае правильная работа автоматизированных тестов даже

более важна.

18.1.11. Этап 11: если ничего не получилось,

выполните план отмены

Если человек, который следит за временем, сообщает, что наступило время

реализации плана отмены, вы должны начать его выполнение. Это может про-

изойти, если обновление занимает больше времени, чем ожидалось, или если

оно завершено, но тесты все еще не выполняются. Решение полностью опреде-

ляется временем, а не вами или вашей группой. Отмена сложного обновления

может разочаровывать или раздражать, но поддержание целостности сервера

важнее.

Возвращение системы к предыдущему состоянию не должно быть единственным

компонентом плана отмены. Пользователи могут согласиться, что, если не будут

выполняться только определенные тесты, они смогут прожить 1–2 дня без

службы, пока она восстанавливается. Заранее определите план действий на

случай каждой потенциальной ошибки.

После выполнения плана отмены службы снова нужно проверить. На этом эта-

пе важно зафиксировать в вашем контрольном списке результаты изменений.

Это полезно для сообщения о состоянии дел руководству, усовершенствования

процесса в следующий раз или восстановления цепи событий во время разбора.

Записывайте такие особенности, как «реализовано по плану», «реализовано, но

превысило заданное время», «частично реализовано, требуется больше работы»,

«не реализовано, изменение отменено», «не реализовано, служба недоступна,

ожидается конец света». По возможности зафиксируйте результаты тестов

и храните их вместе с информацией о состоянии. Это очень здорово поможет,

если вы попытаетесь вспомнить, что произошло, на следующей неделе, в следу-

ющем месяце или в следующем году.

18.1.12. Этап 12: восстановите доступ пользователей

Теперь можно снова позволить пользователям начать работать с системой.

У различных служб есть разные способы это разрешить. Однако часто сложно

осуществить проверку, не разрешив всем пользователям доступ.

Глава 18. Обновления серверов

455

Впрочем, есть ряд способов это сделать. Например, при обновлении сервера

электронной почты вы можете настроить другие серверы электронной почты не

отправлять сообщения на обновляемый сервер. Пока эти серверы удерживают

электронную почту, вы можете вручную проверить обновленный сервер, а затем

сразу разрешить окружающим серверам отправку, внимательно наблюдая за

только что обновленным сервером.

18.1.13. Этап 13: сообщите о завершении/отмене

На этом этапе сотрудникам сообщается о том, что обновление завершено или,

если был задействован план отмены, что было выполнено, что не было выполне-

но и что системами снова можно пользоваться. Здесь выполняются три задачи.

Во-первых, людям сообщают, что службы, к которым у них не было доступа,

снова работают. Во-вторых, пользователям напоминают, что изменилось. Нако-

нец, если они обнаружат проблемы, которые не были найдены в ходе вашего

тестирования, им дают знать, как сообщить об этих проблемах. Если был задействован план отмены, пользователей следует проинформировать о том, что сис-

тема должна работать идентично тому, как это было до попытки обновления.

Точно так же, как есть много способов объявить о техническом перерыве, существует много способов сообщить о завершении. Здесь возможна противоречивая

ситуация. Пользователи не смогут прочесть сообщение электронной почты, если

отключение затронуло службу электронной почты. Однако, если вы будете держаться в рамках своего технического перерыва, после него электронная почта

возобновит свою работу и пользователи смогут прочесть отправленное по ней

объявление. Если пользователи ничего не услышат, они посчитают, что после

окончания объявленного технического перерыва все сделано.

Объявления должны быть краткими. Просто укажите, какие системы или

службы снова работают, и предоставьте ссылку, по которой люди могут обра-

титься для получения более подробной информации, и номер телефона, по ко-

торому можно позвонить, если служба не заработает и будет невозможно отправ-

лять электронную почту. Одного или двух предложений будет достаточно.

Самый простой способ сделать сообщение коротким – переслать первоначальное

сообщение, которое указывало, какие службы будут отключены, и добавить

в начале предложение о том, что службы снова включены, и о том, как инфор-

мировать о проблемах. Это очень эффективно предоставляет людям контекст

того, о чем сообщается.

Большие красные знаки

Пользователи склонны игнорировать сообщения от системных админис-

траторов. Джош Саймон (Josh Simon) рассказал о том, что в одной ком-

пании, которую он обслуживал, он пытался оставлять записки, прикле-

енные к мониторам, – черный текст на ярко-красной бумаге, – где круп-

ным шрифтом было написано «НЕ ВХОДИТЕ В СИСТЕМУ – СНАЧАЛА

СВЯЖИТЕСЬ СО СВОИМ СИСТЕМНЫМ АДМИНИСТРАТОРОМ ПО

ТЕЛЕФОНУ [номер телефона]!». Более 75% пользователей срывали бу-

магу и пытались входить в систему, а не звонили по указанному номеру.

Из этого следует, что часто лучше действительно отключить службу, чем

просить сотрудников не пользоваться ею.

Основы

456

18.2. Тонкости

Что вы можете сделать для развития процесса после того, как овладеете осно-

вами обновления сервера?

18.2.1. Добавляйте и удаляйте службы одновременно

В ходе обновления вы иногда должны одновременно добавлять и удалять служ-

бы. Это усложняет дело, потому что в один момент времени вносится более од-

ного изменения. Отладка системы с двумя изменениями гораздо труднее, пото-

му что требует особых тестов. Для добавления служб характерны те же самые

проблемы, что и для установки новой службы на новом узле, но в данном случае

вы будете в новой, возможно нестабильной, среде и не сможете подготовиться,

написав соответствующие тесты. Однако, если новая служба также доступна на

другом узле, можно разработать тесты и запустить их на нем.

Удаление службы может одновременно быть простым и сложным. Оно может

быть простым по той же причине, по которой разрушить здание проще, чем его

простроить. Однако сначала вы должны убедиться, что в здании нет людей.

Иногда мы устанавливаем анализатор трафика для отслеживания пакетов,

который показывает, что кто-то пытается воспользоваться службой на узле. Эта

информация может быть полезна для поиска отставших.

Мы предпочитаем отключать службу так, чтобы ее можно было быстро снова

активировать, если в дальнейшем будут обнаружены забытые зависимости.

Обычно справедливо допущение, что можно удалить программу, если в течение

следующего месяца или года не будут обнаружены забытые зависимости. Не-

которые службы могут использоваться только раз в квартал или раз в год,

особенно те или иные средства финансовой отчетности. Не забывайте вернуться,

чтобы их убрать! Создайте заявку в вашей системе службы поддержки, отправь-

те себе сообщение электронной почты или создайте задание для at, которое от-

правит вам по электронной почте напоминание через некоторое время. Если

доступ к узлу есть у нескольких групп системных администраторов или приви-

легированных пользователей, может быть полезным внести в файл конфигура-

ции комментарий или переименовать его в OFF или DISABLED (ОТКЛЮЧЕНО).

Иначе другой системный администратор может предположить, что служба

должна работать, и снова ее включит.

18.2.2. Полная установка

Иногда гораздо лучше переустановить систему полностью, чем обновить ее.

Выполнение одного обновления за другим может привести к тому, что система

будет сильно повреждена. При этом могут остаться файлы от предыдущих об-

новлений, фрагментированные файловые системы и «богатое наследие» эпохи

беспорядка.

Ранее мы рассматривали такую роскошь, как точное копирование определенных

дисков и выполнение обновления на копии. Осуществить обновление как полную

установку на другой системе – еще большая роскошь, потому что это не требует

отключения старой системы. Вы можете выполнить полную установку на вре-

менной машине в спокойном темпе, убедиться, что все службы работают, а затем

Глава 18. Обновления серверов

457

подключить диски к обновляемой машине и соответствующим образом настро-

ить конфигурацию сети. Имейте в виду, что машина, на которой осуществля-

ется установка, должна быть практически идентична обновляемой машине,

чтобы на дисках с новой ОС обеспечивалась вся необходимая поддержка и кон-

фигурация оборудования.

18.2.3. Повторное использование тестов

Если тесты хорошо написаны, их можно интегрировать в систему мониторинга

реального времени. На самом деле, если ваша система мониторинга уже выпол-

няет все необходимые тесты, то при обновление вам ничего больше не потребу-

ется (см. в главе 22 более подробное описание мониторинга служб).

Редко все тесты можно автоматизировать и внести в систему мониторинга. На-

пример, тестирование нагрузки – определение того, как работает система под

нагрузкой определенного объема работы, – часто нельзя выполнить на работа-

ющей системе. Однако возможность запустить эти тесты во время низкой ин-

тенсивности использования или по требованию при отладке может упростить

отслеживание проблем.

18.2.4. Запись изменений системы

Построение контрольного списка служб гораздо проще, если вы ведете лог того,

что было добавлено на машину. Например, в UNIX-системе просто записывай-

те изменения в файле под названием /var/adm/CHANGES. Чем проще редакти-

ровать файл, тем более вероятно, что люди будут его обновлять, поэтому создай-

те псевдоним оболочки или короткий скрипт, который просто открывает этот

файл в текстовом редакторе.

Конечно, если машина не будет работать, логи изменений могут быть недоступ-

ны. Хранение лога изменений в википедии или на общем файловом сервере

решает эту проблему, но может привести к путанице, если кто-то попытается

начать новый лог изменений для узла. Установите правила, где должны хра-

ниться логи изменений, и соблюдайте их.

18.2.5. Генеральная репетиция

Учитесь у мира театра: повторение – мать учения. Почему бы перед выполне-

нием обновления не провести генеральную репетицию на другой машине. Это

может раскрыть неожиданные препятствия, а также показать вам, сколько

займет процесс. Генеральная репетиция требует множества ресурсов. Однако,

если вы хотите выполнить первое обновление из многих, она может стать ценным

средством для оценки времени, которое для этого понадобится. Полностью за-

вершенная генеральная репетиция приводит к появлению новой машины, ко-

торая может просто заменить старую машину. Если у вас есть эти ресурсы, по-

чему бы это не сделать?

В театре также бывают так называемые технические репетиции, которые боль-

ше касаются людей, ответственных за свет и звук, чем актеров. Актеры читают

свои роли в нужном порядке, а параллельно их словам делаются указания по

свету и звуку. Эквивалент для системных администраторов – рассмотрение

задачи всеми заинтересованными сторонами.

Тонкости

458

Кроме того, мы заимствуем у театра искусство мимики и жеста. Иногда серьез-

ное изменение в системе предполагает физическую замену большого числа ка-

белей. Почему бы не рассмотреть все этапы, обращая внимание на такие про-

блемы, как длина кабелей, несоответствие прямых и перекрестных обжатий,

несоответствие коннекторов «папа/мама», неправильные коннекторы и конф-

ликтующие планы? Имитируйте замену в точности так, как она должна быть

сделана. Лучше, если рядом с вами будет еще один человек, который станет

объяснять задания по мере того, как вы имитируете их выполнение. Дайте

другому человеку проверить, что каждый коннектор исправен и т. д. Сначала

это может показаться глупым и трудоемким, но проблемы, которые вы предотвратите, того стоят.

18.2.6. Установка старых и новых версий

на одной машине

Иногда на машине обновляется одна служба, а не вся ОС. В этой ситуации удоб-

но, если разработчик разрешает, чтобы старые версии оставались на машине

в отключенном состоянии, пока устанавливаются и сертифицируются новые

программы.

Веб-сервер Apache под UNIX – один из таких продуктов. Мы обычно устанав-

ливаем его в директорию /opt/apache-x.y.x, где x.y.z – это номер версии, но

символическую ссылку из /opt/apache мы помещаем в ту версию, которой хотим

пользоваться. При загрузке новой версии ссылка /opt/apache меняется, чтобы

указывать на новую версию. В случае проблем с новой версией мы восстанавли-

ваем символическую ссылку и перезапускаем демон. Это очень простой план

отмены (применение символических ссылок в базе программного обеспечения

рассмотрено в разделе 28.1.6).

В некоторых ситуациях старая и новая версии программы могут работать одно-

временно. Если требуется серьезная отладка, мы можем запустить новую версию

Apache на другом порте, не трогая старую версию.

18.2.7. Минимальные изменения

первоначальной версии

Обновления становятся проще, если сделать нужно мало. При помощи неболь-

шого планирования все пакеты обновлений UNIX можно загружать в отдельный

раздел, таким образом оставляя системные разделы в минимально измененном

виде. Такие дополнения системы могут документироваться в файле CHANGELOG.

Большинство изменений будет располагаться в директории /etc, которая доста-

точно мала, чтобы можно было скопировать ее перед началом любых обновлений

и пользоваться ею для справки. Это предпочтительнее трудоемкого процесса

восстановления файлов с магнитной ленты.

В UNIX-среде без данных на всех машинах есть локальная ОС, но остальные

данные загружаются с сервера – обычно требуется сохранить между обновле-

ниями только /var, а затем только задачи crontabs и at, данные электронной

почты и, в таких системах как Solaris, файлы менеджера календаря. Для отсле-

живания изменений в файлах конфигурации удобно пользоваться системами

контроля версий, например RCS.

Глава 18. Обновления серверов

459

Пример: обновление критического DNS-сервера

В данном примере объединены многие приемы, рассмотренные в этой

главе. В спешке исправляя все ошибки Y2K перед 1 января 2000 года,

Том нашел критический DNS-сервер, который работал на оборудовании,

не защищенном от ошибки Y2K, и поставщик объявил, что не будет его

исправлять. Кроме того, ОС не поддерживала Y2K. Это была прекрасная

возможность поставить полностью новую ОС на совершенно новом обо-

рудовании.

Том создал контрольный список служб. Хотя он думал, что узел предо-

ставлял только две службы, при помощи netstat –a и перечисления всех

запущенных процессов он обнаружил на машине много других служб.

Он обнаружил, что многие из этих дополнительных служб больше не

использовались, и нашел одну службу, которую никто не смог иденти-

фицировать.

Люди знали, что большинство используемых программ будут работать

на новой ОС, потому что они функционировали на других машинах

с более новой ОС. Однако многие службы были внутренними разработка-

ми, и началась паника, когда не смогли найти исходный код из-за того,

что автор, написавший эту программу, больше не работал в компании.

К счастью, код был найден.

Том собрал новую машину и воспроизвел на ней все службы. На ориги-

нальном узле было много файлов конфигурации, которые регулярно ре-

дактировались. Тому требовалось переписать эти файлы на новую систему,

чтобы проверить, будут ли скрипты, которые их обрабатывают, работать

на новой машине правильно. Однако из-за того, что обновление должно

было занять пару недель, эти файлы могли подвергнуться многократному

изменению, прежде чем новый узел будет готов. Тесты были выполнены

на устаревших данных. Когда новая система была готова, Том остановил

все изменения на старом узле, заново скопировал все файлы на новую

систему и проверил, принимает ли новая система новые файлы.

Разработанные тесты запускались не один раз перед переходом, а много-

кратно, по мере того как появлялась возможность пользоваться служба-

ми на новой системе. Однако Том оставлял большинство служб отклю-

ченными, когда они не тестировались, потому что старые и новые маши-

ны могли конфликтовать друг с другом.

Переход был организован следующим образом: старая машина была от-

ключена от сети, но оставалась включенной. IP-адрес новой машины был

изменен на адрес старой. Через пять минут срок действия ARP-кэшей

локальной сети кончился и новый узел был определен. При возникнове-

нии проблемы Том мог отключить новую машину от сети и подключить

старую. Старая машина осталась включенной, поэтому для ее возвраще-

ния в эксплуатацию не требовалось даже перезагрузки, нужно было

только остановить новый сервер и подключить сетевой кабель старого.

Фактический технический перерыв мог быть довольно коротким – хвати-

ло бы и пяти минут, если бы все прошло хорошо и машину можно было бы

сразу подключить. Однако был объявлен 30-минутный перерыв.

Тонкости

460

Том решил попросить двух человек посмотреть за его работой во время

обновления, потому что он не так хорошо знал эту версию UNIX, как

другие, и мало спал предыдущей ночью. Оказалось, что дополнительная

пара рук помогла в отключении и подключении проводов.

Группа отрепетировала обновление за несколько часов до технического

перерыва. Ничего не меняя, они повторили то, что точно было заплани-

ровано. Они убедились, что длина каждого кабеля будет достаточной

и что все коннекторы были нужного типа. Этот процесс устранил всю

потенциальную путаницу, которая могла бы возникнуть.

Обновление прошло хорошо. Некоторые тесты не были выполнены, но

группа в короткие сроки смогла устранить проблемы. Неожиданной

проблемой стала невозможность выполнения некоторых обновлений баз

данных, пока не был исправлен скрипт. Пользователи, которые зависели

от этих обновляемых данных, согласились работать с несколько устарев-

шими данными, пока на следующий день не исправили скрипт.

18.3. Заключение

Мы описали достаточно полный процесс обновления операционной системы

компьютера, не упоминая при этом ОС конкретных разработчиков, конкретных

команд, которые надо ввести, или кнопок, по которым нужно щелкнуть. Важ-

нейшие элементы процесса связаны не с технологией (это вопрос чтения руко-

водств), а с распространением информации, вниманием к деталям и тестирова-

нием.

Основное средство, которым мы пользовались, – это контрольный список. Мы

начали с составления контрольного списка, которым затем пользовались для

определения того, какие службы требовали обновления, сколько времени займет

обновление и когда мы сможем его выполнить. Контрольный список определя-

ет, какие тесты мы разрабатываем, и эти тесты могут использоваться снова

и снова. Мы пользуемся этими тестами до и после обновления, чтобы обеспечить

должное качество. Если обновление проходит неудачно, мы задействуем планы

отмены, включенные в контрольный список. Когда процесс завершен, мы объ-

являем это заинтересованным пользователям, перечисленным в контрольном

списке.

Контрольный список – это простое средство. Это единственное место, где собра-

на вся информация. Вне зависимости от того, используете ли вы бумагу, элек-

тронную таблицу или веб-страницу, контрольный список является точкой

сбора. Он, фигурально выражаясь, поддерживает единство группы, позволяет

не упускать из вида детали, помогает пользователям понимать процесс, а руко-

водству – следить за происходящим и позволяет новым сотрудникам быстро

входить в курс дела.

Как и многие процессы системного администрирования, обновление требует

навыков общения. Переговоры – это процесс общения, и мы используем его,

когда определяем, когда произойдет обновление, что должно произойти и ка-

ковы приоритеты, если что-то пойдет не так. Мы предоставляем пользователям

ощущение завершенности, сообщая им об окончании работы. Это улучшает

Глава 18. Обновления серверов

461

отношения пользователей и системных администраторов. Мы не можем пере-

оценить важность размещения контрольного списка на веб-странице. Чем

больше людей могут ознакомиться с информацией, тем лучше.

Когда тесты автоматизированы, мы можем точно повторять их и обеспечить

полноту их выполнения. Эти тесты должны быть достаточно общими, чтобы

ими можно было пользоваться для дальнейших обновлений не только на том

же узле, но и на похожих. Фактически тесты должны быть интегрированы

в ваши системы мониторинга реального времени. Зачем выполнять эти тесты

только после обновлений?

Этот простой процесс легко понять и отработать. Это один из основных процес-

сов, которыми системный администратор должен овладеть, прежде чем пере-

ходить к более сложным обновлениям. Все примеры из реальной жизни, которые

мы показали, требовали некоторого отклонения от основного процесса, по все

же содержали главные элементы.

Некоторые дистрибутивы ОС делают обновление практически безопасным

и безболезненным, но другие гораздо более рискованны. Хотя и нет полной га-

рантии успеха, гораздо лучше, когда в операционной системе можно осущест-

влять обновления надежно, с возможностью повторения и простого возвращения

к предыдущему состоянию. Минимальное количество команд или щелчков

мышью снижает вероятность человеческой ошибки. Возможность обновить

много машин воспроизводимым методом имеет массу преимуществ; особенно

важно то, что она помогает поддерживать целостность систем. Любая возмож-

ность вернуться к предыдущему состоянию предоставляет уровень отмены,

аналогичный политике страхования: вы надеетесь, что это никогда вам не по-

надобится, ну а если понадобится, вы будете рады, что обеспечили такую воз-

можность.