12.1. Процесс: EVENT MANAGEMENT - Управление событиями

Управление событиями (Event Management) - *процесс, ответственный за управление событиями в течение жизненного цикла. Управление событиями одна из главных деятельностей операционного управления ИТ.*

Событие (Event) *- изменение состояния, которое имеет значение для управления конфигурационной единицей или услугой.*

Для того чтобы быть эффективным, операционное управление должно знать состояние инфраструктуры и ее компонентов, а также отслеживать любые отклонения от нормальной работы.

Управление событиями реализовывается с помощью систем мониторинга и контроля, которые основаны на двух типах инструментов:

- **Инструменты активного мониторинга** - опрашивают ключевые конфигурационные единицы с целью выяснения их статуса и доступности. Любое отклонение вызовет алерт (предупреждение), который перенаправляется необходимой команде или инструменту для принятия необходимых действий.

Активный мониторинг (Active Monitoring) - мониторинг конфигурационных единиц или услуг, использующий автоматизированные регулярные проверки для отслеживания текущего статуса объекта мониторинга.

- **Инструменты пассивного мониторинга** - обнаруживают и собирают алерты, без принятия каких-либо ответных действий.

Пассивный мониторинг (Passive Monitoring) - мониторинг конфигурационной единицы, услуги или процесса, который основывается на предупреждениях или уведомлениях о текущем состоянии объекта мониторинга.

Из определений, данных в глоссарии, не совсем очевидна разница двух видов мониторинга. Основным отличием является принятие ответных действий в случае алерта при активном мониторинге и их полное отсутствие при пассивном.

Необходимо сразу отметить разницу между Мониторингом и Управлением событиями. Эти процессы очень похожи, но, тем не менее, имеют отличия.

Управление событиями сфокусировано на обнаружении значимых событий, касающихся статусов услуг и инфраструктуры.

Мониторинг же следит за всеми событиями, и, по сути, имеет более широкий охват. Например, *мониторинг* может отслеживать состояние устройства, чтобы удостовериться, что оно функционирует в установленных рамках, даже если устройство не генерирует никаких событий. Таким образом, *Управление событиями* является частью системы мониторинга.

Фактически, *Управление событиями* может контролировать любой аспект сервис-менеджмента.

© YeSSoft 2015

Объектами Управления событиями могут быть:

- Конфигурационные единицы;
- Условия среды (пожар или обнаружение дыма);
- Мониторинг использования лицензий на программное обеспечение;
- Безопасность;
- Нормальная активность (например, использование приложений или производительность сервера).

Ценность Управления событиями для бизнеса косвенная. Тем не менее, можно выделить следующие преимущества, которые дает *Управление событиями* бизнесу:

- Предоставляет механизмы для раннего обнаружения инцидентов. Во многих случаях Управление событиям позволяет обнаружить инцидент и принять соответствующие действия до того, как он значительно повлияет на услугу в целом.
- При интеграции с другими процессами сервис-менеджмента может повысить их эффективность. Он может сообщить об изменениях или отклонениях статусов, что позволит соответствующей команде своевременно предпринять необходимые действия.
- Раннее оповещение о необходимости обновления процедур или ресурсов;
- Управление событиями основано на автоматизированных операциях, что увеличивает эффективность и позволяет задействовать персонал на более "креативные" работы, в частности, проектирование новых услуг и поиск путей по улучшению существующих.

***Управление событиями* работает со следующими типами событий:**

- События, сигнализирующие о регулярной операции, например:
  - Уведомления о том, что работы в соответствии с графиком выполнены;
  - Аутентификация пользователя для использования приложения;
  - E-mail достиг получателя;
- События, отмеченные как отклонения, например:
  - Попытка входа в приложение с использованием некорректного пароля;
  - Нестандартная ситуация в работе бизнес-процесса, которая может потребовать дальнейших действий;
  - Использование CPU выше установленной нормы;
  - Установка неизвестных приложений.
- События, отмеченные как нестандартные, но при этом не являющиеся отклонениями. При обнаружении подобных событий необходим более детальный мониторинг.
-

На рис. 12.1 представлена схема процесса Управления событиями.

© YeSSoft 2015

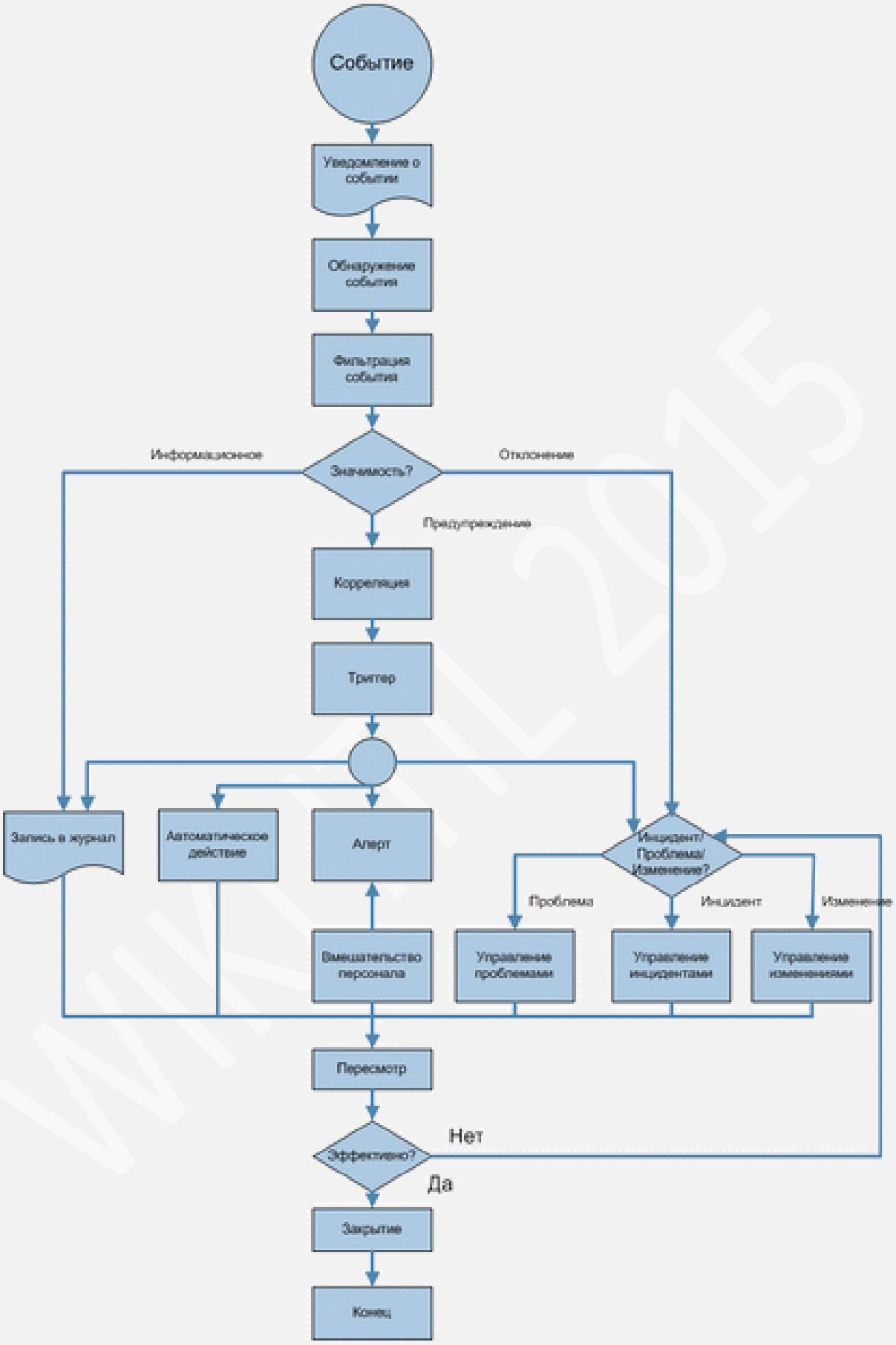

Рис. 12.1. Процесс Управления событиями

© YeSSoft 2015

Разного рода события возникают постоянно, но при этом не все события нужно регистрировать и обнаруживать. Важно, чтобы люди, участвующие в проектировании, разработке, развертывании и поддержке услуг, четко понимали, какие именно события необходимо отслеживать.

Конфигурационные единицы в большинстве случаев выдают предупреждения в случае выполнения определенных условий. Возможность формировать эти предупреждения должна быть спроектирована и встроена в конфигурационные единицы. Многие конфигурационные единицы генерируют предупреждения с использованием открытого стандарта *SNMP*.

В идеальном случае на этапе Проектирования формируется стандартный набор событий, которые необходимо отслеживать в отношении конкретных конфигурационных единиц. В рамках этапа Преобразования этот набор тестируется и настраивается.

После того, как предупреждение сформировано, специальный *агент* в системе обнаруживает его, "читает" и анализирует *значимость* события.

Следующий шаг - фильтрация событий. На этом этапе выносится решение о том, будет ли данное событие проигнорировано или его необходимо передать менеджменту для осуществления необходимых ответных действий. Если событие игнорируется, оно просто записывается в журнал событий (лог). Никаких других действий не выполняется. Фильтрация является первым шагом к классификации событий. Этап фильтрации, по сути, является необязательным.

Каждая организация имеет свои критерии для оценки значимости события, но *ITIL* рекомендует использовать как *минимум* три категории событий:

- **Информационное событие** - относится к событию, которое не требует никаких действий и не является отклонением. Такие события просто записываются в логи и используются для слежения за работой системы и ее компонентов, или контроля выполнения каких-то операций. Они также могут использоваться для сбора статистики и дальнейших исследований. Примерами информационных событий могут быть вход пользователя в систему или успешное завершение транзакции.
- **Предупреждение** - этот тип события формируется тогда, когда услуга или устройство приближается к пороговым значениям. Предупреждения предназначены для того, чтобы соответствующий сотрудник, процесс или инструмент проверили ситуацию более детально и приняли необходимые меры для предотвращения отклонения. Примерами предупреждений могут быть использование памяти сервера более 75% или увеличение количества коллизий в сети.
- **Отклонение** - этот тип событий сигнализирует о том, что услуга или устройство функционируют неправильно (за пределами нормы). Это значит, что нарушаются SLA и OLA, что, как следствие, приводит к негативному влиянию на бизнес в целом. Примерами отклонений могут послужить:
  - выход из строя сервера;
  - больше, чем n пользователей подключились одновременно к приложению N;
  - сегмент сети не отвечает на запросы.

Если событие отмечено как значительное, необходимо определить точно его *значимость* и необходимые ответные действия - это этап корреляции. Корреляция обычно выполняется частью средства управления под названием "*Correlation Engine*", которая применяет к событию

© YeSSoft 2015

набор правил и критериев в определенном порядке. Основная идея в том, что событие может повлиять на бизнес, а правила помогут определить степень и тип этого влияния. В механизме корреляции событий прописывается способ реагирования на событие, например: что делаем при первом, втором и последующих проявлениях данного предупреждающего события, при сочетании или последовательности ряда событий - отклонений, одиночном, но имеющем очень серьезные для заказчика последствия, отклонении.

Примеры того, что может использовать *Correlation Engine* для оценки:

- Количество похожих событий (например, большое количество попыток неправильного ввода пароля может свидетельствовать о попытке взлома устройства);
- Количество конфигурационных единиц, генерирующих похожие события;
- Сопровождают ли событие какие-либо специфичные действия с данными или кодом;
- Является ли событие отклонением;
- Категория события;
- Назначение приоритета событию и т.п.

Механизм, который инициирует ответные действия, называется **триггером**.

Существует множество *типов триггеров*, каждый из которых спроектирован для инициализации конкретных действий. Например:

- Триггеры инцидентов, которые формируют запись в Системе управления инцидентами и, соответственно, запускают процесс Управления инцидентами;
- Триггеры изменений, которые формируют RFC и инициируют процесс Управления изменениями;
- Скрипты, которые выполняют определенные действия, например, перезагрузку устройства;
- Триггеры баз данных, которые ограничивают доступ пользователей к определенным областям базы или удаляют/создают записи в ней.

Следующий этап - выбор ответных действий. Существует множество вариантов ответных действий, которые при этом могут комбинироваться.

Ниже приведены примеры вариантов ответных действий:

- **Запись события в лог.** Вне зависимости от того, какое ответное действие будет выбрано, хорошей практикой является формирование записи о событии в логе. Должна быть стандартная процедура для операционного персонала, предусматривающая периодический анализ логов, а также четкие инструкции о том, как использовать конкретный лог. Также необходимо помнить о том, что информация в логах может не иметь значения до возникновения инцидента. В рамках Управления событиями нужно определить период хранения логов перед тем, как они будут заархивированы или удалены.
- **Автоматические ответные действия.** Для регулярных и "понятных" событий можно разработать автоматические ответные действия. Триггер запустит их и затем проверит результат выполнения. Если что-то пошло не так, будет сформирована запись о проблеме или инциденте. Примерами автоматических ответных действий могут быть:
  1. Перезагрузка устройства;
  2. Повторный запуск услуги;
  3. Изменение параметра устройства;

© YeSSoft 2015

4. Блокировка приложения для предотвращения несанкционированного доступа и т.п.

- **Алерт (предупреждение) и вмешательство людей.** Алерт служит для *уведомления о событии* людей, которые имеют необходимые навыки и знания для его разрешения. При этом алерт должен содержать как можно более полную информацию о событии, на основании которой человек сможет принять правильное решение.
- **Создание Запроса на изменение (RFC).** В Управлении событиями есть две точки, где могут быть созданы RFC:
  - При возникновении отклонения. Например, проверка компьютера показала, что на нем установлено три неавторизованных приложения. В этом случае необходимо сформировать RFC, который поможет процессу управления изменениями устранить отклонение.
  - На этапе корреляции была определена необходимость изменения. В данном случае на этапе корреляции определяется, что наиболее подходящим ответным действием будет изменение чего-то.
- **Создание записи об инциденте.** Если *Correlation* Engine определяет то, что определенный набор событий является инцидентом, создается запись об инциденте. Запись об инциденте должна быть максимально полной и отражать связи со всеми событиями, относящимися к инциденту.
- **Создание записи о Проблеме или формирование связи с уже имеющейся записью.** Инциденты обычно связаны с определенными записями о проблемах. При возникновении инцидента важно связать его с соответствующей записью о проблеме, а если такой нет - создать ее.
- **Особые типы инцидентов.** В некоторых случаях событие может являться отклонением, но при этом не влиять непосредственно на услуги. Такие инциденты не включаются в расчет времени простоя и относятся скорее с проблемам операционного характера. Информация о них должна быть записана в соответствующий лог и передана персоналу, который разбирается с инцидентами этого типа.

Каждый день может происходить десятки и сотни событий и зачастую невозможно детально рассматривать каждое событие. Обзорные действия предназначены для проверки того, как были отработаны инциденты, не пропущены ли какие-то события, сбора статистических данных и т.п. При этом обзорные действия не должны повторять то, что было сделано до этого.

Метрики, которые можно использовать для измерения эффективности Управления событиями:

- Количество событий по категориям;
- Количество событий по значимости;
- Количество событий, которые потребовали участия персонала;
- Количество инцидентов, вызванных известными ошибками и проблемами;
- Количество одинаковых инцидентов (или повторяющихся);
- Количество инцидентов, связанных с проблемами производительности;
- Количество инцидентов, свидетельствующих о наличии потенциальных проблем с доступностью и т.п.

Основными сложностями и рисками для Управления событиями являются недостаточное финансирование, выбор оптимального уровня фильтрации событий и упущение момента для своевременного развертывания агентов в рамках инфраструктуры.

© YeSSoft 2015

Для того чтобы *Управление событиями* было эффективным, его *механизмы* должны быть разработаны на этапе Проектирования услуг в рамках процессов Управления доступностью и Управления мощностями.

Но при этом *Управление событиями* не является статичным - в ходе эксплуатации услуг могут появляться новые требования и события, которые необходимо отслеживать.

Проектирование Управления событиями должно включать следующее:

1. Инструментарий - что может быть отслежено в отношении конфигурационных единиц и как можно воздействовать на них. Другими словами это точное определение и проектирование того, как контролировать и мониторить инфраструктуру и услуги. В рамках определения инструментария необходимо ответить на следующие вопросы:
   - Что необходимо мониторить?
   - Какой тип мониторинга необходим?
   - Когда необходимо формировать событие?
   - Какая информация должна содержаться в событии?
   - Для кого предназначены сообщения о событиях?
2. Сообщения об ошибках должны отображать критичные ошибки, свидетельствующие о сбое или вероятности его возникновения.
3. Механизмы обнаружения событий и формирования алертов. Проектирование этих механизмов требует:
   - Знания взаимосвязей всех бизнес-процессов, которые контролируются с помощью Управления событиями;
   - Знания SLA для каждой услуги, поддерживаемой конфигурационной единицей;
   - Знания того, кто поддерживает конфигурационную единицу;
   - Знания того, какие значения параметров конфигурационной единицы являются нормальными, а какие нет;
   - Понимание того, что именно нужно знать для эффективного управления конфигурационной единицей;
   - Знания информации, которая может помочь эффективной поддержке конфигурационной единицы;
   - Осознания важности совокупности одинаковых или похожих событий;
   - Понимание взаимосвязей конфигурационных единиц;
   - Доступности информации об известных ошибках, полученной от вендоров или из предыдущего опыта.
4. Определение пороговых значений для каждой конфигурационной единицы. При этом значения могут изменяться в зависимости от многих обстоятельств. Например, максимальное количество пользователей, получающих доступ к серверу, зависит от того, какие именно работы они на нем выполняют.

© YeSSoft 2015