

Лекция 5. Основы тестирования

Что такое тестирование?

Системы с программным обеспечением являются неотъемлемой частью нашей жизни, от бизнес-приложений (таких как банковское программное обеспечение) до потребительских товаров (таких как автомобили). Многие люди имели опыт использования программного обеспечения, которое не работало так, как ожидалось. Программное обеспечение, которое не работает корректно, может привести ко многим проблемам, включая потерю денег, времени или деловой репутации, и стать причиной травмы или смерти. Тестирование программного обеспечения – это способ оценить качество программного обеспечения и снизить риск отказа программного обеспечения. Распространено заблуждение о тестировании, что оно состоит только из прогона тестов, т.е. выполнения программного обеспечения и проверки результатов. Тестирование программного обеспечения – это процесс, который состоит из множества различных активностей; и выполнение тестов (включая проверку результатов) является только одной из таких активностей. Процесс тестирования содержит также такие активности, как планирование, анализ, проектирование и реализация тестов, создание отчетов о ходе и результатах тестирования, а также оценка качества объекта тестирования. Тестирование, во время которого задействуется выполнение тестируемого компонента или системы, называется **динамическим тестированием**. Другой вид тестирования, который не предполагает выполнение тестируемого компонента или системы, называется **статическим тестированием**. Таким образом, тестирование также включает рецензирование рабочих продуктов, таких как требования, пользовательские истории и исходный код. Другое распространенное заблуждение о тестировании состоит в том, что оно полностью направлено на проверку требований, пользовательских историй или других спецификаций. Наравне с проверкой соответствия системы установленным требованиям, тестирование содержит в себе проверку, будет ли система удовлетворять потребности пользователей и других заинтересованных лиц в своей рабочей среде (средах). Тестовые активности организованы и реализованы по-разному в различных жизненных циклах.

Основные цели тестирования.

Для любого проекта цели тестирования могут включать:

- Оценку рабочих продуктов, таких как требования, пользовательские истории, проектирование и код

- Проверку, все ли указанные требования выполнены
- Проверку, завершен ли объект тестирования и работает, как ожидают пользователи и заинтересованные лица
- Создание уверенности в уровне качества объекта тестирования
- Предотвращение дефектов
- Обнаружение отказов и дефектов
- Предоставление заинтересованным лицам достаточной информации, позволяющей им принять обоснованные решения, особенно в отношении уровня качества объекта тестирования
- Снижение уровня риска ненадлежащего качества программного обеспечения (например, пропущенные сбои в работе)
- Соблюдение договорных, правовых или нормативных требований, или стандартов и/или проверка соответствия объекта тестирования таким требованиям и стандартам Цели тестирования могут варьироваться в зависимости от контекста тестируемого компонента или системы, уровня тестирования и модели жизненного цикла разработки программного обеспечения.

Эти различия могут включать:

- При компонентном тестировании одна из целей может заключаться в том, чтобы найти как можно больше сбоев, чтобы выявить и устранить основные дефекты на ранних стадиях. Другая цель может быть увеличением покрытия кода тестами компонентов.
- При приемочном тестировании одна из целей может заключаться в том, чтобы подтвердить, что система работает, как ожидалось, и удовлетворяет требованиям. Другая цель этого тестирования может быть предоставлением информации заинтересованным лицам о риске выпуска в установленный срок.

Тестирование и отладка

Стоит различать отладку и тестирование. Выполнение тестов может показать сбои, вызванные дефектами в программном обеспечении. Отладка – это деятельность разработки для нахождения, анализа и исправления таких дефектов. Последующее подтверждающее тестирование проверяет, устранены ли исправленные дефекты. В некоторых случаях тестировщики отвечают только за исходный тест и финальное подтверждающее тестирование, в то время как разработчики выполняют отладку и соответствующее компонентное тестирование. Однако в разработке по гибкой методологии и в некоторых

других жизненных циклах тестировщики могут участвовать в отладке и тестировании компонентов. Стандарт ИСО (ISO/IEC/IEEE 29119-1) содержит дополнительную информацию о концепциях тестирования программного обеспечения.

Почему тестирование необходимо?

Тщательное тестирование компонентов и систем, а также связанной с ними документации может помочь снизить риск отказов во время работы. Когда дефекты обнаружены и исправлены, это способствует качеству компонентов или систем. Также может потребоваться тестирование программного обеспечения на соответствие договорным или правовым требованиям, а также отраслевым стандартам.

Вклад тестирования в успех

На протяжении всей истории компьютеризации довольно часто поставка программного обеспечения и систем в эксплуатацию из-за наличия дефектов приводила к сбоям или иным образом не отвечала потребностям заинтересованных лиц. Однако, использование соответствующих методов тестирования может снизить частоту таких проблемных поставок, если эти методы применяются с соответствующим уровнем опыта тестирования, на соответствующих уровнях тестирования и соответствующих этапах жизненного цикла разработки программного обеспечения. Например:

- Наличие тестировщиков, участвующих в рецензировании требований или уточнении пользовательских историй, может обнаружить дефекты в этих рабочих продуктах. Выявление и устранение дефектов требований снижает риск разработки неправильной или нетестируемой функциональности.
- Наличие тесного сотрудничества тестировщиков с проектировщиками системы во время проектирования системы может повысить понимание архитектуры системы и способов тестирования каждой стороной. Такое более глубокое понимание снижает риск возникновения фундаментальных дефектов проектирования и даёт возможность определить тесты на ранней стадии.
- Наличие тесного взаимодействия тестировщиков с программистами во время разработки кода может улучшить понимание кода и способов его тестирования каждой стороной. Такое более глубокое понимание снижает риск возникновения дефектов в коде и тестах.
- Наличие тестировщиков для валидации и верификации программного обеспечения до релиза может обнаружить сбои, которые в противном случае

могли быть пропущены, и поддержать процесс устранения дефектов, вызвавших отказы (т.е. отладку). Это повышает вероятность того, что программное обеспечение удовлетворяет потребностям заинтересованных лиц и соответствует требованиям. В дополнение к этим примерам достижение определенных целей тестирования способствует общей успешной разработке и сопровождению программного обеспечения.

Обеспечение качества и тестирование

Хотя люди часто используют фразу «обеспечение качества» (или просто QA) для обозначения тестирования, тем не менее, обеспечение качества и тестирование не одно и то же, но эти понятия связаны. Их объединяет более крупное понятие - управление качеством. Управление качеством включает все виды деятельности, которые направляют и контролируют организацию в отношении качества. Помимо других видов деятельности, управление качеством содержит как обеспечение качества, так и контроль качества. Обеспечение качества, как правило, сосредоточено на соблюдении соответствующих процессов для обеспечения уверенности, что будут достигнуты соответствующие уровни качества. Если процессы осуществляются должным образом, рабочие продукты, созданные этими процессами, как правило, более высокого качества, что способствует предотвращению дефектов. Кроме того, использование анализа первопричин для выявления и устранения причин дефектов вместе с надлежащим применением результатов ретроспективы для улучшения процессов имеет большое значение для эффективного обеспечения качества. Контроль качества включает различную деятельность, в том числе работы по тестированию, которые поддерживают достижение соответствующего уровня качества. Работы по тестированию являются частью общего процесса разработки и сопровождения программного обеспечения. Поскольку обеспечение качества связано с надлежащим выполнением всего процесса, обеспечение качества поддерживает надлежащее тестирование.

Ошибки, дефекты и отказы

Человек может совершить ошибку (просчет), которая может привести к появлению дефекта (недочета, помехи) в коде программы или в каком-либо другом сопутствующем продукте. Ошибка, приводящая к появлению дефекта в одном рабочем продукте, может вызвать ошибку, приводящую к появлению дефекта в связанном рабочем продукте. Например, ошибка извлечения требований может привести к дефекту требований, который затем приводит к ошибке программирования, что приводит к дефекту в коде. Если дефект в коде выполняется, это может (но не обязательно во всех ситуациях) привести к отказу. Например, для некоторых дефектов требуются очень специфические входные данные или предварительные условия, чтобы вызвать отказ, который

может произойти редко или никогда. Ошибки могут возникать по многим причинам. Например,

- Нехватка времени
- Человек может ошибаться
- Неопытные или недостаточно квалифицированные участники проекта
- Недопонимание между участниками проекта, включая недопонимание требований и проектирования
- Сложность кода, проектирования, архитектуры, основной проблемы, которую надо решить, и или используемых технологий
- Непонимание внутрисистемных и межсистемных интерфейсов, особенно когда таких внутрисистемных и межсистемных взаимодействий много
- Новые, незнакомые технологии. Кроме отказов, вызванных дефектами в коде, отказы также могут быть вызваны условиями окружающей среды. Например, радиация, электромагнитные поля и загрязнения могут вызвать отказ в программно-аппаратных средствах или повлиять на выполнение программного обеспечения, изменяя условия работы аппаратных средств. Не все неожиданные результаты теста являются отказами. Ложные срабатывания могут возникать из-за ошибок в способе выполнения тестов, из-за дефектов в тестовых данных, тестовой среде или другом тестовом окружении, а также по другим причинам. Может возникнуть и обратная ситуация, когда подобные ошибки или дефекты приводят к ошибочным негативным тестам. Ошибочные негативные тесты – это тесты, которые должны были обнаружить, но не обнаружили дефекты. Ошибочные позитивные тесты – это тесты, которые обнаружили дефекты, не являющиеся реальными дефектами.

1.2.4 Дефект, первопричина и эффект

Первопричины дефектов – это самые ранние действия или условия, которые способствовали созданию дефектов. Дефекты можно проанализировать для поиска первопричины, чтобы уменьшить возникновение подобных дефектов в будущем. Обращая внимание на наиболее существенные первопричины, анализ первопричин может привести к улучшению процессов, которые предотвратят появление значительного числа будущих дефектов. Например, предположим, что одна строка неправильного кода приводит к жалобам клиента, что процентные платежи неверны. Дефектный код был написан для пользовательской истории, которая была неоднозначна, из-за того, что владелец продукта не понимал, как рассчитать проценты. Если в расчетах процентов существует большое

количество дефектов, и первопричина этих дефектов в подобных недоразумениях, то можно обучить владельца продукта расчетам процентов, чтобы уменьшить такие дефекты в будущем. В данном примере жалобы клиентов – это эффекты. Неправильные выплаты процентов являются отказами. Неправильный расчет в коде является дефектом, и он является результатом исходного дефекта - неоднозначности пользовательской истории. Первопричиной первоначального дефекта было отсутствие знаний у владельца продукта, который в свою очередь допустил ошибку при написании пользовательской истории. Процесс анализа первопричин обсуждается в программе подготовки ISTQB Экспертного уровня Тест Менеджер и в программе подготовки ISTQB Экспертного уровня Улучшение тестового процесса.

1.3 Семь принципов тестирования

За последние пятьдесят лет был предложен ряд принципов тестирования, которые являются общим руководством для тестирования в целом.

1. Тестирование демонстрирует наличие дефектов, а не их отсутствие. Тестирование может показать, что дефекты присутствуют, но не может доказать, что их нет. Тестирование снижает вероятность наличия дефектов, находящихся в программном обеспечении, но, даже если дефекты не были обнаружены, тестирование не доказывает его корректности.

2. Исчерпывающее тестирование недостижимо. Полное тестирование с использованием всех комбинаций вводов и предусловий физически невыполнимо, за исключением тривиальных случаев. Вместо попытки исчерпывающего тестирования должны использоваться анализ рисков, методы тестирования и расстановка приоритетов, чтобы сосредоточить усилия по тестированию.

3. Раннее тестирование сохраняет время и деньги. Для нахождения дефектов на ранних стадиях, как статические, так и динамические активности по тестированию должны быть начаты как можно раньше в жизненном цикле разработки программного обеспечения. Раннее тестирование иногда называют «сдвигом влево». Тестирование на ранних этапах жизненного цикла разработки программного обеспечения помогает сократить или исключить дорогостоящие изменения.

4. Кластеризация дефектов. Обычно небольшое количество модулей содержит большинство дефектов, обнаруженных во время тестирования перед выпуском, или отвечает за большинство эксплуатационных отказов. Предсказанные кластеры дефектов и фактические наблюдаемые кластеры дефектов в ходе тестирования или эксплуатации являются важными входными

данными для анализа риска, используемого для сосредоточения усилий по тестированию.

5. Парадокс пестицида. Если одни и те же тесты будут выполняться снова и снова, в конечном счете эти тесты больше не будут находить новых дефектов. Для обнаружения новых дефектов может потребоваться изменение существующих тестов и тестовых данных, а также написание новых тестов. (Тесты больше не эффективны при обнаружении дефектов, так же как пестициды через некоторое время больше не эффективны при борьбе с вредителями). В некоторых случаях, таких как автоматизированное регрессионное тестирование, парадокс пестицидов имеет положительный результат, который является относительно низким числом регрессионных дефектов.

6. Тестирование зависит от контекста Тестирование выполняется по-разному в зависимости от контекста. Например, программное обеспечение управления производством, в котором критически важна безопасность, тестируется иначе, чем мобильное приложение электронной коммерции.

7. Заблуждение об отсутствии ошибок Некоторые организации ожидают, что тестировщики смогут выполнить все возможные тесты и найти все возможные дефекты, но принципы 2 и 1, соответственно, говорят нам, что это невозможно. Кроме того, ошибочно ожидать, что простое нахождение и исправление большого числа дефектов обеспечит успех системе. Например, тщательное тестирование всех указанных требований и исправление всех обнаруженных дефектов может привести к созданию системы, которая будет трудной в использовании, не будет соответствовать потребностям и ожиданиям пользователей или будет хуже по сравнению с другими конкурирующими системами.

Процесс тестирования

Нет универсального процесса тестирования программного обеспечения, но есть общие наборы тестовых активностей, без которых тестирование вряд ли достигнет поставленных целей. Эти наборы тестовых активностей и есть процесс тестирования. Правильный соответствующий процесс тестирования программного обеспечения в любой конкретной ситуации зависит от многих факторов. Какие тестовые активности участвуют в этом процессе и когда они происходят, можно обсудить в корпоративной стратегии тестирования.

Процесс тестирования в контексте

Контекстные факторы, которые влияют на корпоративный процесс тестирования, включают, в частности:

- Используемые модель жизненного цикла разработки программного обеспечения и проектные методологии

- Рассматриваемые уровни и типы тестирования

- Продуктовые и проектные риски

- Предметную область

- Ограничения, включая, в частности: о Бюджеты и ресурсы о Сроки о Сложность о Договорные и нормативные требования

- Организационные политики и практики

- Необходимые внутренние и внешние стандарты В следующих разделах описываются общие аспекты организации процессов тестирования с точки зрения:

- Активностей и задач тестирования

- Рабочих продуктов тестирования

- Трассируемости между базисом тестирования и рабочими продуктами тестирования.

Очень полезно, если базис тестирования (для любого рассматриваемого уровня и типа тестирования) имеет конкретные измеримые критерии покрытия. Критерии покрытия могут эффективно использоваться в качестве ключевых показателей эффективности (КПЭ) для управления мероприятиями, демонстрирующими достижение целей тестирования программного обеспечения. Например, для мобильного приложения базис тестирования может состоять из списка требований и списка поддерживаемых мобильных устройств. Каждое требование является элементом базиса тестирования. Каждое поддерживаемое устройство также является элементом базиса тестирования. Для критериев покрытия может потребоваться по крайней мере один тестовый сценарий для каждого элемента базиса тестирования. После выполнения этих тестов заинтересованные лица узнают, выполнены ли указанные требования и наблюдались ли отказы на поддерживаемых устройствах. Стандарт ИСО (ISO/IEC/IEEE 29119-2) содержит дополнительную информацию о процессах тестирования.

1.4.2 Активности и задачи в тестировании Процесс тестирования состоит из следующих основных групп активностей:

- Планирование тестирования

- Мониторинг и контроль тестирования

- Анализ тестирования

- Проектирование тестов
- Реализация тестов
- Выполнение тестов
- Завершение тестирования

Каждая группа активностей состоит из отдельных активностей, описание которых приводится в подразделах ниже. Каждая активность в рамках каждой группы активностей в свою очередь может состоять из нескольких отдельных задач, которые будут варьироваться от одного проекта или релиза к другому. Более того, хотя многие из этих групп активностей могут выглядеть логически последовательными, они часто реализуются итеративно. Например, в гибкой методологии разработки задействуют небольшие итерации проектирования программного обеспечения, сборки и тестирование происходят постоянно, поддерживаемые непрерывным планированием. Поэтому в рамках этой методологии тестовые активности также повторяются непрерывно. Даже при последовательной разработке логически ступенчатая последовательность активностей будет включать перекрытие, сочетание, параллельное выполнение или пропуск этих активностей, поэтому, как правило, требуется их адаптация в контексте системы и проекта. Планирование тестирования

Планирование тестирования состоит из активностей, которые определяют цели тестирования и подход к достижению целей тестирования с ограничениями, налагаемыми контекстом (например, определение подходящих методов тестирования и задач, а также формирование графика тестирования для соблюдения крайнего срока). Планы тестирования могут быть пересмотрены на основе обратной связи от мониторинга и контроля. Планирование тестирования далее объяснено в разделе

Мониторинг и контроль тестирования

Мониторинг тестирования предполагает непрерывное сравнение фактического хода работы с планом тестирования, используя любые метрики мониторинга тестирования, определённые в плане тестирования. **Контроль тестирования** подразумевает принятие мер, необходимых для достижения целей плана тестирования (который может быть обновлен с течением времени). Мониторинг и контроль тестирования поддерживаются оценкой критериев выхода, которые в некоторых жизненных циклах называются критериями готовности.

Например, оценка критериев выхода для выполнения тестов в рамках заданного уровня тестирования может включать:

- Проверку результатов и журналов тестирования на соответствие заданным критериям покрытия

- Оценку уровня качества компонентов или систем на основе результатов и журналов тестирования

- Определение потребности в дополнительных тестах (например, потребуется написание и выполнение дополнительных тестов, если тесты, первоначально предназначенные для достижения покрытия определенного уровня риска продукта, не смогли этого сделать)

Прогресс тестирования по сравнению с планом сообщается заинтересованным лицам в отчетах о ходе тестирования, включая отклонения от плана и информацию для подтверждения какого-либо решения о прекращении тестирования. Мониторинг и контроль тестирования более подробно описаны в разделе

Анализ тестирования

В процессе анализа тестирования анализируют базис тестирования для определения тестируемых функций и установление соответствующих тестовых условий. Другими словами, анализ тестирования решает “что тестировать” с точки зрения измеримых критериев покрытия. Анализ тестирования состоит из следующих основных активностей:

- Анализ базиса тестирования, применимого к рассматриваемому уровню тестирования.

Базисом тестирования могут быть, например:

- о Спецификации требований, такие как бизнес-требования, функциональные требования, системные требования, пользовательские истории, бизнеспотребности, сценарии использования системы, или аналогичные рабочие продукты, которые описывают функциональные и нефункциональные компоненты или поведение системы

- о Информация о проектировании и реализации, такая как диаграммы или документы архитектуры системы или программного обеспечения, спецификации проектирования, потоки вызовов, диаграммы моделирования (например, UML или диаграммы «сущность-связь»), спецификации интерфейсов или аналогичные рабочие продукты, которые определяют структуру компонента или системы

- о Реализация самого компонента или системы, включая код, метаданные и запросы базы данных, а также интерфейсы

- о Отчеты анализа рисков, в которых могут быть рассмотрены функциональные, нефункциональные и структурные аспекты компонента или системы

- Оценка базиса тестирования и элементов тестирования для выявления дефектов различных типов, таких как:

- Неоднозначность
- Пропуски
- Несоответствие
- Неточность
- Противоречивость
- Избыточные утверждения

- Определение свойств и совокупность свойств для тестирования

- Установление и приоритизация тестовых условий для каждого свойства на основе анализа базиса тестирования, с учетом функциональных, нефункциональных и структурных характеристик, других бизнес и технических факторов, а также уровней рисков

- Обеспечение двунаправленной трассируемости между каждым элементом базиса тестирования и соответствующими тестовыми условиями

Применение методов тестирования на основе черного ящика, белого ящика и на основе опыта может быть полезно в процессе анализа тестирования для уменьшения вероятности пропуска важных и определения более ясных и точных тестовых условий. Выявление дефектов в ходе анализа тестирования является важным потенциальным преимуществом, особенно если процесс рецензирования не используется и/или если процесс тестирования тесно связан с процессом рецензирования. Такие активности по анализу тестирования не только проверяют, являются ли требования согласованными, сформулированными должным образом и полными, но также проверяют, правильно ли требования отражают потребности клиентов, пользователей и других заинтересованных лиц. Например, такие методы как разработка, основанная на описании поведения (BDD) и разработка через приемочные тесты (ATDD), которые затрагивают формирование тестовых условий и тестовых сценариев из пользовательских историй и критериев приемки перед кодированием, проверяют, корректируют и выявляют дефекты в пользовательских историях и критериях приемки (см. программу ISTQB Базовый уровень. Тестировщик в сфере Гибких методологий). Проектирование тестов. Во время проектирования тестов тестовые условия воплощаются в высокоуровневые тестовые сценарии, наборы высокоуровневых тестовых сценариев и другое тестовое обеспечение. Так, анализ тестирования отвечает на вопрос «что тестировать?», а проектирование тестов отвечает на вопрос «как тестировать?».

Проектирование тестов состоит из следующих основных активностей:

- Проектирования и приоритизации тестовых сценариев и наборов тестовых сценариев
- Определения необходимых тестовых данных для поддержки тестовых условий и тестовых сценариев
- Проектирования тестового окружения и определения необходимой инфраструктуры и инструментов
- Отражения двунаправленной трассируемости между базисом тестирования, тестовыми условиями, тестовыми сценариями и процедурами тестирования

Воплощение тестовых условий в тестовые сценарии и в наборы тестовых сценариев во время проектирования тестов часто включает использование методов тестирования. Так же, как и при анализе тестирования, проектирование тестов может привести к выявлению аналогичных типов дефектов в базисе тестирования. Кроме того, как и при анализе тестирования, выявление дефектов при проектировании тестов является важным потенциальным преимуществом. Реализация тестов. Во время реализации тестов создается и/или подготавливается необходимое тестовое обеспечение для выполнения тестов, включая упорядочивание тестовых сценариев в процедурах тестирования. Таким образом, проектирование тестов отвечает на вопрос «как проверить?», в то время как реализация тестов отвечает на вопрос: «у нас теперь есть все для запуска тестов?». Реализация тестов – это активность, во время которой процедуры или сценарии тестирования выстраиваются в определенном порядке, чтобы облегчить выполнение тестов.

Реализация тестов состоит из следующих основных активностей:

- Разработка и расстановка приоритетов процедур тестирования и, возможно, создание автоматизированных сценариев тестирования
- Создание наборов тестов из процедур тестирования и (при наличии) автоматизированных сценариев тестирования
- Организация наборов тестов с расписанием выполнения тестов таким образом, чтобы тесты выполнялись эффективно
- Построение тестового окружения (в том числе, возможно, тестовые стенды, сервисы виртуализации, симуляторы и другие элементы инфраструктуры) и проверка правильности настройки всего необходимого

- Подготовка тестовых данных и правильная загрузка их в тестовое окружение

- Проверка и обновление двунаправленной трассируемости между базисом тестирования, тестовыми условиями, тестовыми сценариями, процедурами тестирования и наборами тестов. Часто задачи по проектированию тестов и реализации тестов объединяют. При исследовательском тестировании и других типах тестирования, основанных на опыте, разработка и реализация тестов могут выполняться и документироваться как часть выполнения тестов. Исследовательское тестирование может основываться на концепциях тестирования (составленных в рамках анализа тестирования), и исследовательские тесты выполняются немедленно по мере их разработки и реализации.

Выполнение тестов.

Во время выполнения тестов, наборы тестов запускаются в соответствии с расписанием выполнения тестов. Выполнение тестов состоит из следующих основных активностей:

- Запись идентификаторов и версий элемента (-ов) тестирования или объекта тестирования, инструмента (-ов) тестирования, и тестового обеспечения

- Выполнение тестов вручную или с помощью инструментов выполнения тестов

- Сравнение фактических и ожидаемых результатов

- Анализ отклонений для установления их вероятных причин (например, отказы могут произойти из-за дефектов в коде, но также могут возникнуть ложные срабатывания)

- Составление отчетов о дефектах на основе наблюдаемых отказов

- Протоколирование результатов выполнения тестов (например, пройден, не пройден, блокировка)

- Повторение тестовых действий, результаты которых привели к каждому из отклонений, либо в рамках запланированного тестирования (например, выполнение исправленного теста, подтверждающее тестирование и/или регрессионное тестирование)

- Проверка и обновление двунаправленной трассируемости между базисом тестирования, тестовыми условиями, тестовыми сценариями, процедурами тестирования и результатами тестирования
Завершение тестирования

Активности по завершению тестирования собирают данные из выполненных активностей тестирования для обобщения опыта, тестового обеспечения и любой другой соответствующей информации. Активности по завершению тестирования происходят на вехах проекта, например, по завершению выпуска релиза программного обеспечения системы, завершению (отмене) проекта тестирования, окончанию итерации проекта с гибкой методологией (например, как часть итогового митинга), завершению уровня тестирования, или завершению сопровождения релиза.

Завершение тестирования состоит из следующих основных активностей:

- Проверка закрытия всех отчетов о дефектах, входящих запросов на изменение или набора задач продукта для любых дефектов, которые остаются не реализованными в конце выполнения тестирования
- Создание сводного отчета по тестированию для передачи заинтересованным лицам
- Завершение и архивирование тестового окружения, тестовых данных, инфраструктуры тестирования и другого тестового обеспечения для последующего использования
- Передача тестового обеспечения команде сопровождения, другим проектным командам, и/или другим заинтересованным лицам, которые могут извлечь выгоду из его использования
- Анализ полученных уроков для определения изменений, необходимых для будущих итераций, релизов и проектов
- Использование собранной информации для повышения зрелости процесса тестирования

Рабочие продукты тестирования

Создание рабочих продуктов тестирования — это часть процесса тестирования. Поскольку существует значительное различие в том, как организации реализовывают процесс тестирования, есть существенные различия и в типах рабочих продуктах, созданных в ходе этого процесса - как эти рабочие продукты организованы и управляются, и в названиях, используемых для этих рабочих продуктов. Эта программа соответствует описанному выше процессу тестирования, а также рабочим продуктам, описанным в этой программе и в Стандартном глоссарии терминов ISTQB. Стандарт ИСО (ISO/IEC/IEEE 29119-3) может также служить руководством для создания рабочих продуктов тестирования. Используя средства управления тестированием и управления дефектами, можно отслеживать и

управлять многими рабочими продуктами тестирования, описанными в этом разделе.

Рабочие продукты планирования тестирования.

Рабочие продукты планирования тестирования – это один или несколько планов тестирования. План тестирования содержит информацию о базисе тестирования, с которым другие рабочие продукты тестирования будут связаны через информацию о трассируемости, а также критерии выхода (или определение готовности), которые будут использоваться во время мониторинга и контроля тестирования.

Рабочие продукты мониторинга и контроля тестирования

Рабочие продукты мониторинга и контроля тестирования – это обычно различные типы отчетов о тестировании, включая отчеты о ходе тестирования (выпускаемые оперативно и/или на регулярной основе) и сводные отчеты о тестировании (выпускаемые на различных этапах завершения). Все отчеты о тестировании должны обеспечить соответствующей аудитории подробные сведения о ходе тестирования на дату отчета, включая обобщение результатов выполнения тестирования после их получения. Рабочие продукты мониторинга и контроля тестирования также должны учитывать интересы управления проектами, такие как завершение задач, распределение и использование ресурсов и трудозатрат. Мониторинг и контроль тестирования, а также рабочие продукты, созданные в ходе этих активностей.

Рабочие продукты анализа тестирования

Рабочие продукты анализа тестирования состоят из определенных и приоритизированных тестовых условий, каждое из которых в идеале двунаправленно прослеживается до конкретного элемента (элементов) базиса тестирования, который они покрывают. Для исследовательского тестирования анализ тестирования может включать и создание концепций тестирования. Анализ тестирования может также привести к обнаружению и составлению отчетов о дефектах в базисе тестирования.

Рабочие продукты проектирования тестов

Результатом проектирования тестов являются тестовые сценарии и наборы тестовых сценариев для выполнения тестовых условий, определенных в анализе тестирования. Часто хорошей практикой считается проектирование высокоуровневых тестовых сценариев без конкретных значений для входных данных и ожидаемых результатов. Такие высокоуровневые тестовые сценарии многократно используются для нескольких циклов тестирования с различными конкретными данными, но при этом все еще достаточно документируют область применения тестового сценария. В идеале, каждый

тестовый сценарий двунаправленно прослеживается к тестовым условиям, которые он покрывает. Проектирование тестов также приводит к разработке и/или установлению необходимых тестовых данных, к проектированию тестового окружения и определению инфраструктуры и инструментов, хотя объем документирования этих результатов может существенно различаться. Тестовые условия, определенные в анализе тестирования, могут быть дополнительно уточнены при проектировании тестов.

Рабочие продукты реализации тестов

Рабочие продукты реализации тестов включают:

- Процедуры тестирования и последовательности этих процедур тестирования
- Наборы тестов
- Расписания выполнения тестов

В идеале после завершения реализации тестов достижение критериев покрытия, установленных в плане тестирования, может быть продемонстрировано посредством двунаправленной трассируемости между процедурами тестирования и конкретными элементами базиса тестирования, с помощью тестовых сценариев и тестовых условий. В некоторых случаях реализация тестов включает создание рабочих продуктов, использующих или используемых инструментами, такими как службы виртуализации и автоматизированные тестовые сценарии. Реализация тестов также может привести к созданию и проверке тестовых данных и тестового окружения. Полнота документирования результатов проверки данных и/или окружения может существенно различаться. Тестовые данные служат для присвоения конкретных значений входным данным и ожидаемым результатам тестовых сценариев. Такие конкретные значения вместе с точными указаниями об использовании конкретных значений превращают высокоуровневые тестовые сценарии в исполняемые низкоуровневые тестовые сценарии. Один и тот же высокоуровневый тестовый сценарий может использовать разные тестовые данные при выполнении в разных релизах объекта тестирования. Конкретные ожидаемые результаты, связанные с конкретными тестовыми данными, определяются с помощью тестового предсказателя. При исследовательском тестировании во время выполнения тестов могут создаваться некоторые рабочие продукты для разработки и реализации тестов, хотя степень документирования исследовательских тестов (и их трассируемость до конкретных элементов базиса тестирования) может существенно различаться. Тестовые условия, определенные в анализе тестирования, могут быть дополнительно уточнены при реализации тестов.

Рабочие продукты выполнения тестов

Рабочие продукты выполнения тестов включают:

- Документацию о состоянии отдельных тестовых сценариев или процедур тестирования (например, готов к запуску, пройден, не пройден, блокирован, осознанный пропуск и т. д.)
- Отчеты о дефектах (см. раздел 5.6)
- Документацию о том, какие элемент(ы) теста, объект(ы) тестирования, инструменты тестирования, и тестовое обеспечение были задействованы в тестировании.

В идеале, как только выполнение тестов завершено, состояние каждого элемента базиса тестирования можно определить и сообщить через двунаправленную трассируемость к соответствующим процедурам тестирования. Например, можно указать, для каких требований успешно выполнены все запланированные тесты, для каких требований не прошли тесты и/или имеют связанные с ними дефекты, для каких требований запланировали тесты, которые ожидают выполнения. Это позволяет проверить, что критерии покрытия были достигнуты, и позволяет в понятных формулировках сообщать о результатах тестирования заинтересованным лицам.

Рабочие продукты завершения тестирования

Рабочие продукты завершения тестирования состоят из сводных отчетов тестирования, мероприятий по улучшению последующих проектов или итераций (например, ретроспектива после проекта с гибкой методологией), запросов на изменение или набора задач продукта, и окончательного тестового обеспечения.

Трассируемость между базисом тестирования и рабочими продуктами тестирования

Рабочие продукты тестирования и названия этих рабочих продуктов существенно различаются. Независимо от этих различий для осуществления эффективного мониторинга и контроля тестирования важно установить и поддерживать трассируемость на протяжении всего процесса тестирования между каждым элементом базиса тестирования и различными рабочими продуктами тестирования, связанных с этим элементом. В дополнение к оценке покрытия тестирования, хорошая трассируемость поддерживает:

- Анализ влияния изменений
- Проведение тестирования, поддающееся аудиту

- Достижение критериев управления
- Улучшение понятности отчетов о ходе тестирования и сводных отчетов, включая статус элементов базиса тестирования (например, требования, для которых прошли тесты, требования, для которых тесты провалились, и требования, ожидающие тестирования)
- Согласование технических аспектов тестирования с заинтересованными лицами в терминах, которые они могут понять
- Предоставление информации для оценки качества продукции, потенциала процессов и развития проекта в соответствии с бизнес-целями. Некоторые инструменты управления тестированием предоставляют модели рабочих продуктов тестирования, которые соответствуют части или всем рабочим продуктам тестирования, изложенным в этом разделе. Некоторые организации создают собственные системы управления для организации рабочих продуктов и обеспечения требуемой трассируемости информации.

Психология тестирования

Люди принимают участие в разработке программного обеспечения, в том числе и тестировании программного обеспечения. Таким образом, психология человека оказывает важное влияние на тестирование программного обеспечения.

Психология человека и тестирование

Выявление дефектов во время статического тестирования, такого как рецензирование требований или сессий уточнения пользовательских историй, или выявление отказов во время выполнения динамического теста может быть воспринято как критика продукта и его автора. Элемент человеческой психологии, называемый предвзятостью подтверждения, может затруднить принятие информации, которая не совпадает с текущими убеждениями. Например, поскольку разработчики ожидают, что их код будет правильным, у них есть предвзятость подтверждения, которое затрудняет принятие того, что код неправильный. В дополнение к предвзятости подтверждения, другие стереотипы людей могут затруднить понимание или принятие информации, полученной в результате тестирования. Кроме того, общепринятой человеческой чертой является то, что виноват носитель плохих новостей, а информация, полученная в результате тестирования, часто содержит плохие новости. В результате этих психологических факторов некоторые люди могут воспринимать тестирование как разрушительную деятельность, даже если она в значительной степени способствует развитию проекта и качеству продукта. Чтобы попытаться уменьшить эти восприятия, информацию о дефектах и отказах следует сообщать конструктивным образом. Таким образом,

напряженность между тестировщиками и аналитиками, владельцами продуктов, дизайнерами и разработчиками можно уменьшить. Это относится как к статическому, так и к динамическому тестированию. Тестировщики и руководители по тестированию должны иметь хорошие коммуникационные навыки, чтобы иметь возможность эффективно сообщать о дефектах, отказах, результатах тестирования, ходе тестирования, и рисках, а также строить позитивные отношения с коллегами. Следующие примеры хорошо передают способы общения:

- Начните с сотрудничества, а не сражения. Напомните каждому об общей цели улучшения качества системы.

- Подчеркните преимущества тестирования. Например, информация о дефектах может помочь авторам улучшить свои рабочие продукты и навыки. Для организации, дефекты, найденные и зафиксированные во время тестирования, сохраняют время и деньги и уменьшают общий риск качества продукции.

- Сообщайте результаты тестирования и другие выводы нейтральным, сфокусированным на фактах, способом, без критики автора. Пишите объективные и фактические отчеты о дефектах и выводах рецензирования.

- Попытайтесь понять, что другие люди чувствуют, и причины их негативной реакции на информацию.

- Убедитесь, что другой человек понял, что вы сказали, и наоборот. Четкое определение правильного набора целей тестирования имеет важные психологические последствия. Большинство людей склонно корректировать свои планы согласно целям, установленным командой, руководством и другими заинтересованными лицами. Также важно, чтобы тестировщики придерживались этих целей с минимальными личными предубеждениями.

Мышление тестировщика и разработчика

Разработчики и тестировщики часто думают по-разному. Основной целью разработчика является проектирование и создание продукта. Как обсуждалось ранее, цели тестирования включают верификацию и валидацию продукта, поиск дефектов до релиза и т. д. Это разные наборы целей, которые требуют разного образа мышления. Объединение этих образов мышления вместе помогает достичь более высокого уровня качества продукции. Образ мышления отражает предположения человека и предпочтительные методы как для принятия решений, так и для решения проблем. Образ мышления тестировщика должен включать любопытство, профессиональный пессимизм, критический взгляд, внимание к деталям и мотивацию хороших и позитивных коммуникаций и отношений. Образ мышления тестировщика имеет

тенденцию к развитию по мере приобретения опыта тестировщиком. Образ мышления разработчика может включать в себя некоторые элементы образа мышления тестировщика, но успешные разработчики чаще заинтересованы в разработке и создании решений, чем в рассмотрении того, что может быть неправильно в этих решениях. Кроме того, подтверждение предвзятости затрудняет поиск ошибок в собственной работе. При правильном образе мышления разработчики могут протестировать свой собственный код. Различные модели жизненного цикла разработки программного обеспечения часто имеют разные способы организации тестировщиков и мероприятий по тестированию. Выполнение некоторых действий независимыми тестировщиками повышает эффективность обнаружения дефектов, что особенно важно для больших, сложных или критически важных для безопасности систем. Независимые тестировщики приносят перспективу, которая отличается от перспективы авторов продукта работы (т.е. бизнес-аналитиков, владельцев продукта, дизайнеров и программистов), поскольку их предубеждения отличаются от авторских