

Лекция 9: Управление тестированием

Организация тестирования. Независимость тестирования

Задачи тестирования могут выполняться как людьми в конкретной роли, связанной с тестированием, так и людьми в иной роли (например, заказчиками). Конкретный уровень независимости часто делает тестировщика более эффективным в поиске дефектов из-за психологических различий разработчика и тестировщика. Независимость не является, однако, альтернативой осведомленности, и разработчики могут находить много дефектов в своем собственном коде. Уровни независимости тестирования включают следующие (от низкого уровня до высокого уровня независимости):

- Нет независимых тестировщиков; разработчики тестируют собственный код, других форм тестирования нет
- Независимые разработчики или тестировщики в команде разработки или в команде проекта; это могут быть разработчики, тестирующие продукты своих коллег
- Независимая команда или группа тестирования внутри организации, отчеты перед руководством проекта или руководством организации
- Независимые тестировщики из организации заказчика или сообщества пользователей. Могут специализироваться на отдельных типах тестирования, таких как тестирование практичности, тестирование безопасности, тестирование производительности, тестирование на соответствие нормативным документам или тестирование переносимости
- Независимые тестировщики, внешние по отношению к организации, работающие либо на территории компании (инсорсинг), либо вне ее (аутсорсинг)

Для большинства проектов лучше иметь несколько уровней тестирования, часть из которых выполняется независимыми тестировщиками. Разработчики должны участвовать в тестировании, особенно на более низких уровнях, чтобы контролировать качество своей работы. Способ организации независимого тестирования зависит от модели жизненного цикла. Например, в гибкой разработке тестировщики могут быть частью команды разработчиков. В некоторых организациях, использующих гибкие методологии разработки программного обеспечения, тестировщики могут также считаться частью большой независимой группы тестирования. Кроме того, в таких организациях владельцы продуктов могут выполнять приемочные испытания для проверки пользовательских историй в конце каждой итерации. К потенциальным преимуществам независимого тестирования можно отнести:

- Эффективное распознавание различных видов отказов по сравнению с разработчиками из-за разницы подходов, технических перспектив и предубеждений

- Возможность независимого тестировщика проверять, оспаривать или опровергать допущения, сделанные заинтересованными сторонами во время проектирования и внедрения системы.

К потенциальным недостаткам независимого тестирования относятся:

- Изоляция от команды разработчиков, что приводит к отсутствию сотрудничества, задержкам с предоставлением обратной связи команде разработчиков или соперничеству с командой разработчиков

- Потеря разработчиками чувства ответственности за качество

- Восприятие независимых тестировщиков как «узкого места» и обвинения в задержках релиза.

- Недостаточность у независимых тестировщиков какой-либо важной информации (например, об объекте тестирования). Многие организации могут успешно пользоваться преимуществами независимого тестирования, избегая при этом указанных недостатков.

Задачи руководителя тестирования и тестировщика

В этой программе рассматриваются две тестовые роли - руководителя тестирования и тестировщика. Действия и задачи, выполняемые людьми в этих двух ролях, зависят от контекста проекта и продукта, навыков людей в этих ролях и организации. Руководитель тестирования отвечает за процесс тестирования и успешное руководство активностями тестирования. Роль руководителя тестирования может выполняться профессиональным руководителем тестирования, руководителем проекта, руководителем разработки, или руководителем по обеспечению качества. В крупных проектах или организациях несколько команд тестирования могут отчитываться перед одним руководителем тестирования, куратором или координатором тестирования, при этом каждая команда возглавляется лидером тестирования или ведущим тестировщиком. Типичные задачи руководителя тестирования могут включать:

- Разработку или рецензирование тестовой политики и стратегии тестирования для организации

- Планирование тестирования, с учетом контекста и понимания целей и рисков тестирования. Это может включать выбор тестовых подходов, оценку времени тестирования, трудозатрат и стоимости, привлечение ресурсов,

определение уровней тестирования и циклов тестирования и планирование управления дефектами

- Составление и обновление планов тестирования
- Согласование планов тестирования с руководителями проектов, владельцами продуктов и другими участниками
- Координацию активностей тестирования с другими проектами, такими как планирование интеграции
- Инициирование анализа, разработки, реализации и выполнения тестов, отслеживание прогресса и результатов тестирования, контроль выполнения критериев выхода (или критериев готовности)
- Подготовку и предоставление отчетов о статусе тестирования и сводных отчетов о тестировании на основе собранной информации
- Адаптацию планов в зависимости от прогресса и результатов тестирования (и выполнение действий, необходимых для контроля тестирования)
- Поддержку настройки системы управления дефектами и конфигурацией тестового обеспечения
- Выбор подходящих метрик для измерения результатов тестирования и оценки качества процесса тестирования и продукта
- Выбор и внедрение инструментов поддержки процесса тестирования, включая рекомендации по выбору инструментария (и, возможно, приобретение и / или поддержку), выделение времени и трудозатрат для пилотных проектов, обеспечение постоянной поддержки в использовании инструмента/инструментов
- Решения о создании тестовой среды/сред
- Демонстрацию ценности тестировщиков, группы тестирования и профессии тестировщика в организации
- Развитие навыков и карьеры тестировщиков (например, посредством планов обучения, оценки эффективности, коучинга и т.д.)

Реализация роли руководителя тестирования зависит от используемой модели жизненного цикла разработки. Например, в гибкой разработке некоторые из упомянутых выше задач выполняются командой, особенно те, что связаны с повседневным тестированием. Они выполняются внутри команды, часто с помощью тестировщиков, работающих в команде. Задачи, охватывающие несколько команд или всю организацию, или связанные с управлением персоналом, могут выполняться руководителями тестирования

(иногда их называют кураторами тестирования) за пределами команды разработки. К типичным задачам тестировщика могут относиться:

- Рецензирование и разработка планов тестирования
- Анализ, рецензирование и оценка требований, пользовательских историй и критериев приемки, спецификаций и моделей (базиса тестирования) на предмет тестируемости
- Определение и документирование тестовых условий, установление связей между тестовыми сценариями, тестовыми условиями и базисом тестирования
- Проектирование, настройка и проверка тестовой среды/сред, зачастую вместе с системным администрированием и управлением сетью
- Проектирование и разработка тестовых сценариев и процедур тестирования
- Подготовка и получение тестовых данных
- Создание подробного расписания выполнения тестов
- Выполнение тестирования, оценка результатов и документирование отклонений от ожидаемых результатов
- Использование соответствующих инструментов поддержки процесса тестирования
- Автоматизация процесса тестирования по мере необходимости (может поддерживаться разработчиком или экспертом по автоматизации тестирования)
- Оценка нефункциональных характеристик, таких как производительность, надежность, практичность, безопасность, совместимость и переносимость
- Рецензирование тестов, разработанных другими тестировщиками

Люди, занимающиеся тест-анализом, проектированием тестов, выполняющие специфические виды тестирования или занимающиеся автоматизацией, могут быть специалистами в этих ролях. В зависимости от рисков проекта и продукта, а также используемой модели жизненного цикла, роль тестировщика на различных уровнях тестирования могут выполнять разные люди. Например, на уровне компонента и интеграции компонентов роль тестировщика часто выполняют разработчики. На уровне приемки роль тестировщика могут выполнять бизнесаналитики, эксперты и пользователи. На уровне системы и интеграции систем роль тестировщика могут выполнять специалисты независимой команды тестирования. На уровне

эксплуатационной приемки роль тестировщика часто выполняют специалисты по сопровождению и/или администрированию системы.

Планирование и оценка тестирования

Цель и содержание плана тестирования. В плане тестирования перечисляются работы по тестированию для проектов разработки и сопровождения. Планирование зависит от политики тестирования и стратегии тестирования организации, используемых методов и жизненных циклов разработки, объема тестирования, целей, рисков, ограничений, критичности, тестируемости и доступности ресурсов. По ходу проекта становится доступно больше информации, и в план тестирования могут быть включены дополнительные сведения. **Планирование тестирования** - непрерывная деятельность, которая выполняется в течение всего жизненного цикла продукта. (Обратите внимание, что жизненный цикл продукта может выходить за рамки проекта и включать в себя фазу сопровождения.) Обратная связь в ходе работ по тестированию должна использоваться для идентификации изменяющихся рисков с последующей корректировкой планов. План может быть оформлен как в виде главного плана тестирования, так и в виде уровней планов (план системного тестирования, план приемочного тестирования) или планов для отдельных видов тестирования (план тестирования практичности, план тестирования производительности). Мероприятия по планированию тестирования могут включать в себя следующие работы, часть из которых может быть отражена в плане тестирования:

- Определение объема, целей и рисков тестирования
 - Определение общего подхода к тестированию
 - Координацию работ по тестированию и их совмещение с другими работами в рамках жизненного цикла программного обеспечения
 - Принятие решений о том, что тестировать, кто будет выполнять тестирование и как должны выполняться работы по тестированию
 - Планирование анализа, проектирования, реализации и выполнения тестов, оценки результатов тестирования с указанием сроков (при последовательной разработке) либо итераций (при итеративной разработке)
 - Выбор метрик для мониторинга и контроля тестирования
 - Формирование бюджета тестирования
 - Определение структуры и уровня детализации тестовой документации (например, путем предоставления шаблонов или примеров документов)
- Содержание планов тестирования различается и может выходить за пределы

указанных тем. Примеры планов тестирования можно найти в стандарте ИСО (ISO/IEC/IEEE 29119-3).

Стратегия тестирования и подходы к тестированию

Стратегия тестирования содержит верхнеуровневое описание процесса тестирования, как правило, на уровне продукта или организации. К распространенным типам стратегий тестирования относятся:

- Аналитический подход – базируется на анализе некоторого фактора (например, требования или риска). Тестирование, основанное на рисках, является примером аналитического подхода, при котором тесты разрабатываются и ранжируются по приоритетам в зависимости от уровня риска.

- Тестирование на основе моделей – подход, при котором тесты разрабатываются на основании модели некоторого аспекта продукта, такого как функция, бизнес-процесс, внутренняя структура или нефункциональная характеристика (например, надежность). Примерами являются модели бизнес-процессов, модели состояния и модели роста надежности.

- Методический подход – основан на систематическом использовании некоторого predetermined набора тестов или тестовых условий, таких как классификация общих или вероятных типов отказов, список важных характеристик качества или корпоративный стандарт дизайна мобильных приложений или веб-страниц.

- Тестирование на основе процесса (или стандарта) – подразумевает анализ, проектирование и выполнение тестов в соответствии с внешними правилами или стандартами, такими как: отраслевые стандарты, документация процесса, базис тестирования или любой другой нормативной базой, используемой в организации.

- Направленный (или консультативный) подход – определяется, прежде всего, советами, руководствами или инструкциями заинтересованных сторон, экспертов предметной области или экспертов по технологиям, которые могут находиться вне команды тестирования или организации.

- Тестирование на основе минимизации регресса – нацелено на проверку работоспособности существующих возможностей ПО. Эта стратегия тестирования подразумевает повторное использование существующего тестового обеспечения (особенно тестовых сценариев и тестовых данных), обширную автоматизацию регрессионных тестов и стандартные наборы тестов.

- **Реактивный подход** – тестирование в данном случае не является спланированным заранее, но зависит от тестируемого компонента, системы или событий, происходящих при выполнении тестов. Новые тесты проектируются, разрабатываются и выполняются, исходя из знаний, ранее полученных при тестировании. Исследовательское тестирование является распространенной методикой, применяемой в реактивных стратегиях. Подходящая стратегия тестирования может быть создана путем объединения нескольких стратегий тестирования. Например, тестирование на основе рисков (аналитическая стратегия) может сочетаться с исследовательским тестированием (реактивная стратегия); они дополняют друг друга и могут обеспечить более эффективное тестирование при совместном использовании. Стратегия тестирования дает общее описание процесса тестирования, в то время как подход к тестированию адаптирует стратегию к конкретному проекту или релизу. Подход к тестированию является отправной точкой при выборе методов тестирования, уровней и типов тестирования, а также при определении критериев входа и выхода (или критериев готовности и критериев завершения соответственно). Построение стратегии тестирования зависит от решений, принятых в отношении сложности и целей проекта, типа разрабатываемого продукта, анализа рисков продукта. Выбранный подход зависит от контекста и может учитывать такие факторы, как риски, безопасность, доступные ресурсы и навыки, технологии, характер системы (например, разработанная на заказ или готовый коммерческий продукт), цели тестирования и правила.

Критерии входа и выхода (критерии готовности и критерии завершения)

Для обеспечения эффективного управления тестированием и качеством программного обеспечения рекомендуется иметь критерии, которые определяют, когда начинается и завершается каждая из работ по тестированию. Критерии входа (или критерии готовности в гибкой разработке) определяют условия, которые должны быть выполнены до начала работ. Если критерии входа не выполнены, вполне вероятно, что выполняемая задача окажется более сложной, более трудоемкой, более дорогостоящей и более рискованной. Критерии выхода (или критерии завершения в гибкой разработке) определяют, какие условия должны быть выполнены, чтобы завершить уровень тестирования или набор тестов. Критерии входа и выхода должны быть определены для каждого уровня и типа тестирования и могут отличаться в зависимости от целей тестирования. Типичные критерии входа включают:

- Доступность тестируемых требований, пользовательских историй и/или моделей (например, при использовании стратегии тестирования на основе моделей)

- Наличие элементов тестирования, которые удовлетворяют критериям выхода для предыдущих уровней тестирования

- Доступность тестовой среды

- Наличие необходимых инструментов тестирования

- Наличие тестовых данных и других необходимых ресурсов

Типичные критерии выхода включают:

- Выполнение запланированных тестов

- Достижение определенного уровня покрытия (например, требований, пользовательских историй, критериев приемки, рисков, кода)

- Количество открытых дефектов ниже оговоренного порогового значения

- Низкая оценка количества еще не обнаруженных дефектов

- Соответствие требуемым значениям оценок надежности, производительности, практичности, безопасности и других характеристик качества.

Даже если критерии завершения не выполняются, тестирование может быть сокращено из-за превышения бюджета, истечения запланированного времени и/или необходимости вывода продукта на рынок. Завершение тестирования может быть приемлемым, если заинтересованные лица со стороны проекта и бизнеса рассмотрели и приняли риск вывода продукта в промышленную эксплуатацию без дальнейшего тестирования.

Расписание выполнения тестов

После того, как тестовые сценарии и процедуры (в том числе автоматизированные) разработаны, их объединяют в наборы тестов. Эти наборы располагаются в соответствии с расписанием тестирования, которое задает последовательность их выполнения. Расписание должно учитывать такие факторы как: приоритет, зависимости между тестами и/или тестируемыми функциями, необходимость выполнения подтверждающих тестов и регрессионных тестов и наиболее эффективную последовательность выполнения тестов. В идеальном случае тестовые сценарии упорядочиваются на основе их приоритетов, при этом сначала выполняются тестовые сценарии с наивысшим приоритетом. Однако эта практика может не работать, если тесты или тестируемые функции имеют зависимости. Если тестовый сценарий

с более высоким приоритетом зависит от тестового сценария с более низким приоритетом, то сначала выполняется тестовый сценарий с более низким приоритетом. Аналогичным образом, если в тестовых сценариях есть зависимости, они должны быть упорядочены соответствующим образом, без учета относительных приоритетов. Подтверждающие и регрессионные тесты тоже должны иметь приоритет, исходя из важности обратной связи об изменениях, но здесь также могут применяться зависимости. Иногда могут использоваться разные последовательности тестов, имеющие разные уровни эффективности. В таких случаях должен быть достигнут компромисс между эффективностью выполнения тестов и соблюдением приоритетов.

Факторы, влияющие на затраты на тестирование. Оценка затрат на тестирование подразумевает прогнозирование объема связанной с тестированием работы, которая необходима для достижения целей тестирования проекта, релиза или итерации. Факторы, влияющие на затраты, могут включать характеристики продукта, характеристики процесса разработки, характеристики людей и результаты тестирования.

Характеристики продукта включают:

- Риски, связанные с продуктом
- Качество базиса тестирования
- Размер продукта
- Сложность предметной области продукта
- Требования к характеристикам качества (например, безопасности, надежности)
- Требуемый уровень детализации тестовой документации
- Требования к юридическому и нормативному соответствию

Характеристики процесса разработки включают:

- Стабильность и зрелость организации;
 - Используемую модель разработки
 - Подход к тестированию
 - Используемые инструменты
 - Процесс тестирования
 - Сжатость сроков
- Характеристики людей включают:

- Навыки и опыт, особенно в аналогичных проектах и продуктах (например, знание предметной области)
- Командная сплоченность и лидерство

Результаты тестирования включают:

- Количество и критичность выявленных дефектов
- Объем требуемых доработок

Методы оценки затрат на тестирование.

Существует несколько методов, используемых для оценки затрат на адекватное тестирование. Наиболее популярные методы — это:

- Метод, основанный на метриках - оценка затрат, использующая метрики ранее выполненных проектов или типовые значения
- Метод экспертной оценки - оценка затрат на основе опыта владельцев задач тестирования или экспертов

Например, в гибкой разработке диаграммы сгорания являются примерами метода, основанного на метриках: трудозатраты фиксируются и отслеживаются, а затем используются для оценки скорости работы команды и определения объема работы, которую команда может выполнить в следующей итерации. Poker планирования является примером метода, основанного на экспертизе, поскольку члены команды оценивают трудозатраты на основе своего опыта (ISTQBAT Базовый уровень. Тестировщик в сфере Гибких методологий). В итерационных методологиях примером метода, основанного на метриках, является модель устранения дефектов: сбор информации о количестве дефектов и времени на их исправление дает базис для оценки схожих проектов в будущем. Метод «Дельфи», с другой стороны, является экспертным методом, в котором группа экспертов дает оценки, исходя из своего опыта.

Контроль и мониторинг тестирования

Цель мониторинга тестирования заключается в сборе информации и обеспечении обратной связи о состоянии тестирования. Требуемая информация может собираться вручную или автоматически и использоваться для отслеживания прогресса тестирования, оценки выполнения критериев выхода или критериев готовности (в случае гибкой разработки), таких, как обеспечение требуемого покрытия рисков продукта, требований или критериев приемки. Контроль тестирования представляет собой любые

корректирующие действия, предпринятые на основании полученной информации или метрик. Действия могут охватывать любую активность тестирования и влиять на любую активность жизненного цикла программного обеспечения. Примеры действий по контролю тестирования:

- Повторная приоритизация тестов при реализации риска (например, нарушения сроков поставки программного обеспечения)
- Изменение графика тестирования из-за доступности или недоступности тестовой среды или других ресурсов
- Повторная проверка выполнения критериев входа или выхода для элемента тестирования, который дорабатывался

Контроль и мониторинг тестирования

Метрики могут собираться вовремя и по завершении тестирования, чтобы оценить:

- Прогресс относительно запланированного графика и бюджета
- Текущее качество объекта тестирования
- Адекватность подхода к тестированию
- Эффективность активностей тестирования по достижению целей тестирования Типичные метрики тестирования включают:
- Процент выполненных работ по подготовке тестовых сценариев (или процент разработанных тестовых сценариев)
- Процент выполненных работ по подготовке тестовой среды
- Метрики выполнения тестов: количество выполненных/невыполненных тестовых сценариев, количество тестовых условий или сценариев, выполненных успешно/неуспешно
- Информацию о дефектах: плотность дефектов, количество обнаруженных и исправленных дефектов, частоту отказов и результаты подтверждающих тестов
- Покрытие тестами требований, пользовательских историй, критериев приемки, рисков или кода
- Информацию о выполнении задач, распределении и использовании ресурсов, трудозатратах
- Стоимость тестирования, включая сравнение стоимости с выгодой от нахождения следующего дефекта или от выполнения следующего теста

Цели, содержание и аудитория отчетов о тестировании

Цель отчетности состоит в обобщении и предоставлении информации по итогам и во время тестирования. Отчет, подготовленный во время тестирования, называется отчетом о ходе тестирования; отчет, подготовленный по итогам, называется итоговым отчетом о тестировании. Во время мониторинга и контроля тестирования руководитель тестирования регулярно публикует для заинтересованных сторон отчеты о ходе тестирования. Помимо разделов, общих для отчета о ходе тестирования и итогового отчета о тестировании, отчет о ходе тестирования может включать:

- Статус активностей тестирования и прогресс по сравнению с планом тестирования
- Факторы, препятствующие прогрессу
- Тестирование, запланированное на следующий отчетный период
- Качество объекта тестирования

Когда критерии выхода выполнены, руководитель тестирования готовит итоговый отчет о тестировании. В этом отчете приводится краткое описание выполненного тестирования (на основе последнего отчета о ходе тестирования и другой информации). Обычно итоговые отчеты и отчеты о ходе тестирования включают:

- Резюме проведенного тестирования
- Информацию о том, что произошло во время тестирования
- Информацию об отклонениях от плана, включая отклонения в расписании, длительности выполнения или затратах
- Информацию о качестве тестирования и качестве продукта с точки зрения критериев выхода или критериев завершения
- Информацию о факторах, которые блокировали или продолжают блокировать тестирование
- Метрики дефектов, тестовых сценариев, покрытия, прогресса тестирования и использования ресурсов
- Информацию об остаточных рисках
- Перечень тестовых артефактов, которые можно повторно использовать

Содержимое отчета о тестировании зависит от проекта, корпоративных требований и жизненного цикла разработки программного обеспечения. Например, для сложных, формальных проектов с большим числом

заинтересованных лиц может потребоваться более подробная и строгая отчетность. В качестве другого примера: в гибкой разработке отчеты о ходе тестирования могут быть частью панели задач, сводки по дефектам и диаграммы сгорания, которые обсуждаются на ежедневных встречах (см. ISTQB AT Базовый уровень. Тестировщик в сфере Гибких методологий). Помимо контекста проекта, при подготовке отчетности нужно учитывать особенности аудитории. Тип и объем информации, включаемые в отчет для технических специалистов или группы тестирования, будут отличаться от информации в отчете для менеджмента. В первом случае может быть важна подробная информация о типах дефектов и тенденциях. В последнем случае может быть уместным отчет более высокого уровня (например, статус дефектов, сгруппированных по приоритету, информация о бюджете и расписании, успешные/неуспешные/непроверенные тестовые условия). Стандарт ИСО (ISO/IEC/IEEE 29119-3) описывает два типа отчетов о тестировании: отчет о ходе тестирования и отчет о завершении тестирования (называемый в этом документе итоговым отчетом) и содержит структуру и примеры оформления отчетов каждого типа.

Управление конфигурацией

Целью управления конфигурацией является обеспечение и поддержка целостности компонента или системы, тестового обеспечения и их взаимосвязей между собой на протяжении жизненного цикла проекта и продукта. Для поддержки тестирования управление конфигурацией может потребовать выполнения следующих условий:

- Все элементы тестирования однозначно идентифицированы, связаны между собой, находятся под версионным контролем, все изменения в них отслеживаются
- Все элементы тестового обеспечения однозначно идентифицированы, связаны между собой и с версиями элементов тестирования, находятся под версионным контролем, все изменения отслеживаются, обеспечивая трассируемость на протяжении всего процесса тестирования
- Все документы и программные элементы идентифицированы и однозначно указаны в тестовой документации Инфраструктура и процедуры управления конфигурацией должны быть подготовлены и выполнены на этапе планирования тестирования.

Риски и тестирование

Определение риска

Риск подразумевает наступление некоторого негативного события в будущем. Уровень риска можно определить через вероятность события и серьезность последствий.

Риски проекта и продукта

Риск продукта – это возможное несоответствие некоторого артефакта (спецификации, компонента, системы или теста и т.д.) потребностям пользователей и заинтересованных сторон. Когда риски продукта связаны с конкретными характеристиками качества (функциональной пригодностью, надежностью, производительностью, практичностью, безопасностью, совместимостью, сопровождаемостью и переносимостью), их также называют рисками качества. Примеры рисков продукта:

- Программное обеспечение не выполняет функции, указанные в спецификации
 - Программное обеспечение не выполняет функции, ожидаемые пользователями, клиентами и/или заинтересованными сторонами
 - Системная архитектура не поддерживает нефункциональные требования
 - Вычисления выполняются неверно в каких-то ситуациях
 - Неверная реализация в коде структуры управления циклом
 - Неадекватное время отклика высоконагруженной системы
 - Отзывы пользователей о продукте показывают, что их ожидания не оправдались
- Риски проекта связаны с событиями, препятствующими достижению целей проекта.

Примеры рисков проекта:

- Проектные проблемы:
 - о Задержки поставки, выполнения задач, выполнения критериев выхода или критериев готовности
 - о Неточные оценки, перераспределение средств на проекты с более высоким приоритетом или общие сокращения затрат по всей организации могут привести к неадекватному финансированию
 - о Поздние изменения могут привести к существенным доработкам
- Организационные проблемы:
 - о Недостаток навыков, обучения или численности персонала

- о Проблемы с персоналом могут вызвать конфликты и серьезные трудности

- о Пользователи, бизнес-пользователи или эксперты предметной области могут быть заняты другими работами

- Политические проблемы:

- о Тестировщики не могут адекватно сообщать о своих потребностях и/или результатах тестирования

- о Разработчики и/или тестировщики не могут отслеживать информацию, полученную при тестировании и рецензировании (не стремясь улучшить методы разработки и тестирования)

Может быть неправильное отношение или ожидания от тестирования (недооценивается важность обнаружения дефектов во время тестирования и т.д.)

- Технические проблемы:

- о Требования могут быть определены недостаточно хорошо
- о Требования могут быть невыполнимыми в текущих условиях

- о Тестовая среда может быть не готова вовремя

- о Преобразование данных, планирование миграции и их инструментальная поддержка могут быть не готовы вовремя

- о Слабые стороны процесса разработки могут влиять на согласованность или качество артефактов проекта, таких как дизайн, код, конфигурация, тестовые данные и тестовые сценарии

- о Проблемы управления дефектами могут привести к накоплению дефектов и росту технического долга

- Проблемы с поставщиками:

- о Третья сторона может не предоставить необходимый продукт или услугу, или обанкротиться

- о Контрактные проблемы могут вызвать серьезные трудности для проекта

Проектные риски могут влиять как на разработку, так и на тестирование. В некоторых случаях руководители проектов несут ответственность за управление всеми проектными рисками; руководители тестирования обычно отвечают за управление рисками, связанными с тестированием проекта.

Тестирование, основанное на рисках, и качество продукта

Риски используются для распределения усилий во время тестирования. Они используются для принятия решения о том, где и когда начинать тестирование, и для выявления областей, требующих большего внимания. Тестирование используется с целью снижения вероятности возникновения неблагоприятного события и снижения последствий при наступлении этого события. Тестирование используется как деятельность по снижению риска и для обеспечения обратной связи, как по выявленным рискам, так и по остаточным (нерешенным) рискам. Тестирование, основанное на рисках, обеспечивает возможность заранее снизить уровень риска продукта. Оно включает анализ рисков продукта (идентификацию рисков, оценку вероятности и последствий от их наступления). Полученная информация используется для планирования тестирования, разработки, подготовки и выполнения тестовых сценариев, а также для мониторинга и контроля тестирования. Ранний анализ рисков продукта способствует успеху проекта. В рамках подхода, основанного на рисках, результаты анализа рисков продукта могут быть использованы для:

- Выбора методов тестирования
- Выбора уровней и типов тестирования, которые необходимо выполнить (например, тестирования безопасности, тестирования доступности)
- Определения объема тестирования
- Приоритезации тестирования с целью найти критические дефекты как можно раньше
- Выявления каких-либо дополнительных мер, помимо тестирования, которые могут снизить риски (например, обучения менее опытных проектировщиков) Тестирование, основанное на рисках, опирается на коллективные знания заинтересованных сторон проекта и выполнение ими анализа рисков продукта. Чтобы минимизировать вероятность неуспеха продукта, активности по управлению рисками обеспечивают дисциплинированный подход к:
 - Анализу (и повторной оценке на регулярной основе) того, что может пойти не так (рисков)
 - Определению, какие риски важны для дальнейшей работы
 - Выполнению действий по снижению этих рисков
 - Разработке планов действий в чрезвычайных ситуациях, если риски станут реальными событиями. Кроме того, тестирование может выявить новые

риски, помочь определить, какие риски следует смягчить, и снизить неопределенность в отношении рисков.

Управление дефектами. Поскольку одной из целей тестирования является обнаружение дефектов, необходимо регистрировать дефекты, обнаруженные во время тестирования. Способ регистрации дефектов может варьироваться в зависимости от контекста, тестируемого компонента или системы, уровня тестирования и модели жизненного цикла. Любые выявленные дефекты должны быть изучены и отслеживаться от момента обнаружения и классификации до принятия решения по ним (например, исправления дефектов и успешного подтверждающего тестирования решения, отсрочки до следующего релиза, трактовки как постоянного ограничения продукта и т. д.). Чтобы управлять дефектами до принятия решения по ним, в организации должен существовать процесс управления дефектами, который включает в себя жизненный цикл и правила классификации дефектов. Этот процесс должен быть согласован со всеми, кто участвует в управлении дефектами, включая проектировщиков, разработчиков, тестировщиков и владельцев продуктов. В некоторых организациях регистрация и отслеживание дефектов могут быть очень слабо формализованными. Во время процесса управления дефектами некоторые из отчетов могут содержать описания ложных срабатываний, а не фактических сбоев из-за дефектов. Например, тест мог пройти неуспешно при сбое сетевого соединения или таймауте. Такое поведение не является следствием дефекта объекта тестирования, но аномалией, которую необходимо исследовать. Тестировщики должны попытаться свести к минимуму количество ложных срабатываний, принимаемых за дефекты. Дефекты могут быть обнаружены во время кодирования, статического анализа, рецензирования, динамического тестирования или эксплуатации программного продукта. Дефекты могут сообщать о проблемах в коде или эксплуатируемых системах, в документации любого типа, включая требования, пользовательские истории и критерии приемки, документацию разработки, документацию тестирования, руководства пользователя или руководства по установке. Чтобы иметь продуктивный и эффективный процесс управления дефектами, организации могут определять стандартный набор атрибутов, правила классификации и жизненный цикл дефектов. Типичные отчеты о дефектах имеют следующие цели:

- Предоставлять разработчикам и другим сторонам информацию о произошедших негативных событиях, чтобы они могли определить побочные эффекты, изолировать проблему с минимальными затратами на воспроизведение и исправить потенциальные дефекты по мере необходимости, или решать проблемы другими способами

- Обеспечить руководителей тестирования инструментами отслеживания качества продукта и влияния на тестирование (например, если сообщается о большом количестве дефектов, то тестировщики будут вынуждены тратить много времени на отчетность по найденным дефектам вместо того, чтобы запускать тесты; следовательно, нужно больше подтверждающего тестирования)

- Предоставить идеи для совершенствования процессов тестирования и разработки Сообщение о дефекте, созданное во время динамического тестирования, обычно включает:

- Идентификатор дефекта
- Заголовок и краткое описание найденного дефекта
- Дату сообщения о дефекте, информацию об авторе сообщения
- Идентификацию элемента тестирования (проверяемого элемента конфигурации) и среды
- Фазу жизненного цикла разработки, в которой обнаружен дефект
- Описание дефекта, достаточное для его воспроизведения и принятия решения, включая системные журналы, скриншоты, дампы базы данных или записи (если они созданы во время выполнения теста)
- Ожидаемые и фактические результаты
- Область или степень влияния дефекта на интересы заинтересованной стороны (критичность)
- Срочность/приоритет для исправления
- Статус дефекта (например, открыт, отложен, дубликат, ожидает исправления, ожидает проверки, повторно открыт, закрыт)
- Выводы, рекомендации и согласования
- Глобальные проблемы, например, области, которые будут затронуты исправлением дефекта
- История изменений, например, последовательность действий членов команды проекта, чтобы изолировать дефект, исправить и подтвердить исправление дефекта
- Ссылки, включая ссылку на тестовый сценарий, который обнаружил дефект

Некоторые из этих деталей могут автоматически включаться и/или настраиваться при использовании инструментов управления дефектами.

Например, автоматическое присваивание идентификатора дефекта, назначение и обновление статуса дефекта на протяжении жизненного цикла и т.д. Дефекты, обнаруженные при статическом тестировании, в частности, при рецензировании, как правило, документируются по-другому, например, в примечаниях к протоколу совещания. Пример содержимого сообщения о дефекте можно найти в стандарте ИСО (ISO/IEC/IEEE 29119-3) (где сообщения о дефектах называются сообщениями об инцидентах).