

Организация тестирования в команде разработчиков. Виды и методы тестирования.

Эффективность поиска дефектов и рецензирования может быть повышена с помощью независимых тестировщиков. Варианты независимости могут быть следующими:

- Отсутствие независимых тестировщиков: разработчики тестируют собственный код;
- Независимые тестировщики в команде разработчиков;
- Независимая команда или группа тестирования в организации, отчитывающаяся менеджеру проекта или исполнительному менеджеру;
- Независимые тестировщики из бизнес-организации или сообщества пользователей;
- Независимые специалисты тестирования для отдельных типов тестирования, например, тестировщики удобства использования, тестировщики безопасности или тестировщики сертификации (которые сертифицируют ПО на соответствие стандартам и правилам);
- Независимые тестировщики, привлеченные на аутсорсинг или сторонние по отношению к организации.

Какие виды тестирования бывают?

- Модульное тестирование (component testing)
- Интеграционное тестирование (integration testing)
- Системное тестирование (system testing)
- Приемочное тестирование (acceptance testing)

Какие методы тестирования существуют?

1. Black Box

Методика тестирования без каких-либо знаний о внутренней работе приложения называется «черным ящиком». Тестер не обращает внимания на архитектуру системы и не имеет доступа к исходному коду. Как правило, при выполнении теста с «черным ящиком» тестер будет взаимодействовать с пользовательским интерфейсом системы, предоставляя входные данные и анализируя выходы, не зная, как и где обрабатываются входы.

Преимущества	Недостатки
Хорошо подходит и эффективен для больших сегментов кода.	Ограниченное покрытие, поскольку на самом деле выполняется только выбранное количество тестовых сценариев.
Кодовый доступ не требуется.	Неэффективное тестирование, из-за того, что тестер только имеет ограниченные знания о приложении.
Четкое разделение перспективы пользователя с точки зрения разработчика с помощью явно определенных ролей.	Слепой охват, поскольку тестер не может ориентироваться на определенные сегменты кода или области ошибок.
Большое количество умеренно квалифицированных тестировщиков может протестировать приложение без каких-либо знаний о реализации, языке программирования или операционных системах.	Тестовые примеры трудно разработать.

2. White Box

Проверка белого ящика - это подробное исследование внутренней логики и структуры кода. Тестирование с использованием белого ящика также называется **тестированием стекла** или **открытым тестированием**. Чтобы выполнить тестирование **белого ящика** в приложении, тестер должен знать внутреннюю работу кода.

Преимущества	Недостатки
Поскольку тестер знает исходный код, становится очень легко узнать, какой тип данных может помочь в эффективном тестировании приложения.	В связи с тем, что для тестирования белых ящиков требуется квалифицированный тестер, затраты увеличиваются.
Это помогает в оптимизации кода.	Иногда невозможно заглянуть в каждый уголок и угол, чтобы обнаружить скрытые ошибки, которые могут создавать проблемы, так как многие пути будут непроверены.

Дополнительные строки кода могут быть удалены, что может привести к скрытым дефектам.	Трудно поддерживать тестирование белых ящиков, поскольку для этого требуются специализированные инструменты, такие как анализаторы кода и инструменты отладки.
Благодаря знаниям тестера о коде, максимальный охват достигается при написании сценария сценария.	

3. Gray Box

Тестирование на серой коробке - это метод тестирования приложения с ограниченным знанием внутренней работы приложения. При тестировании программного обеспечения фраза, чем больше вы знаете, тем лучше переносит массу при тестировании приложения.

Преимущества	Недостатки
Предлагает комбинированные преимущества тестирования черного ящика и белого ящика, где это возможно.	Поскольку доступ к исходному коду недоступен, возможность пройти через код и зону тестирования ограничена.
Тестировщики серого ящика не полагаются на исходный код; вместо этого они полагаются на определение интерфейса и функциональные спецификации.	Тесты могут быть излишними, если разработчик программного обеспечения уже выполнил тестовый пример.
Основываясь на имеющейся ограниченной информации, тестер серого ящика может разработать отличные сценарии тестирования, особенно в отношении протоколов связи и обработки данных.	Тестирование всех возможных входных потоков нереально, поскольку для этого потребуется необоснованное количество времени; поэтому многие программные пути будут непроверены.
Тест выполняется с точки зрения пользователя, а не дизайнера.	

Сравнение White Box, Grey Box, Black Box

Тестирование Black-Box	Тестирование серых коробок	Тестирование белого ящика
Не нужно знать внутреннюю работу приложения.	Тестер имеет ограниченное знание внутренней работы приложения.	Тестер имеет полное представление о внутренней работе приложения.
Также известен как тестирование с закрытым ящиком, тестирование с использованием данных или функциональное тестирование.	Также известен как прозрачное тестирование, поскольку тестер имеет ограниченное знание внутренних аспектов приложения.	Также известен как прозрачное тестирование, структурное тестирование или тестирование на основе кода.
Выполняется конечными пользователями, а также тестировщиками и разработчиками.	Выполняется конечными пользователями, а также тестировщиками и разработчиками.	Обычно выполняются тестировщиками и разработчиками.
Тестирование основано на внешних ожиданиях. Внутреннее поведение приложения неизвестно.	Тестирование выполняется на основе диаграмм базы данных высокого уровня и диаграмм потоков данных.	Внутренние работы полностью известны, и тестер может соответствующим образом создавать тестовые данные.
Он является исчерпывающим и наименее трудоемким.	Частично трудоемкий и исчерпывающий.	Самый исчерпывающий и трудоемкий тип тестирования.
Не подходит для тестирования алгоритмов.	Не подходит для тестирования алгоритмов.	Подходит для тестирования алгоритмов.
Это можно сделать только методом проб и ошибок.	Домены данных и внутренние границы могут быть проверены, если они известны.	Домены данных и внутренние границы могут быть лучше проверены.