



ЦЕЛИ И ЗАДАЧИ ТЕСТИРОВАНИЯ ПО

ЧТО ТАКОЕ ТЕСТИРОВАНИЕ?

- Сэм Канер - «Тестирование – это поиск ошибок».
- Ли Копланд - «Тестирование – это сведение к минимуму риска пропуска ошибки».
- Крупнейший институт инженеров IEEE утверждает, что «Тестирование – это проверка продукта на соответствие требованиям».
- В некоторых источниках даже можно найти утверждения, что «Тестирование – это процесс, направленный на демонстрацию корректности продукта».
- **Тестирование программного обеспечения** — процесс исследования программного обеспечения (ПО) с целью получения информации о качестве продукта.

1. Целью тестирования является обнаружение ошибок в тестируемом объекте, а не доказательство их отсутствия.

2. Тестировщики не отвечают за качество. Они помогают тем, кто за него отвечает.

3. Тестирование даёт тем большую экономическую отдачу, чем на более ранних стадиях работы над проектом оно выявило дефект.

4. Тестирование имеет смысл прекращать тогда, когда устранены все критические и 85% и более некритических дефектов программы, т.к. дальнейшее тестирование, как правило, является неоправданной статьёй расходов.

Несколько определений

Дефект (баг, глюк; defect, bug) – любое несоответствие фактического и ожидаемого результата (согласно требованиям или здравому смыслу).

Тест-кейс (test case) – набор входных данных, условий выполнения и ожидаемых результатов, разработанный с целью проверки того или иного свойства или поведения программного средства.

Тест-план (test plan) – часть проектной документации, описывающая и регламентирующая процесс тестирования.

Билд (build) – промежуточная версия программного средства (финальный билд часто называют релизом (release)).

Основные виды ошибок

- Логическая ошибка - наиболее серьезная из всех ошибок. Когда написанная программа на любом языке компилирует и работает правильно, но выдает неправильный вывод, недостаток заключается в логике основного программирования. Это ошибка, которая была унаследована от недостатка в базовом алгоритме.
- Синтаксические (в записи конструкций языка программирования) и семантические (связанные с неправильным содержанием действий и использованием недопустимых значений величин) ошибки устраняются на этапе компиляции.
- Ошибка компиляции исправляются на стадии разработки.
- Ошибки среды выполнения могут возникнуть в результате нехватки ресурсов носителя. можно исправить, вернувшись к стадии кодирования.
- Арифметическая ошибка - ведущая к бесконечному результату, логическая ошибка, которая может быть исправлена только путем изменения алгоритма
- Ошибки ресурса - переполнение буфера, использование неинициализированной переменной, нарушение прав доступа и переполнение стека
- Ошибка взаимодействия несоответствие программного обеспечения с аппаратным интерфейсом или интерфейсом прикладного программирования. В случае веб-приложений, ошибка интерфейса может быть результатом неправильного использования веб-протокола.

Продукты, подвергаемые тестированию

Тестировать можно (и нужно!):

- ✓ **Программы** при их непосредственном запуске и исполнении (software).
- ✓ **Код** программ без запуска и исполнения (code).
- ✓ **Прототип** программного продукта (product prototype).
- ✓ **Проектную документацию** (project documentation):
 - **Требования** к программному продукту (product requirements).
 - Функциональные **спецификации** к программному продукту (functional specifications).
 - **Архитектуру** (architecture) и дизайн (design).
 - План проекта (project plan) и тестовый план (test plan).
 - **Тестовые случаи сценарии** (test cases,).
- ✓ **Сопроводительную документацию** (и документацию для пользователей):
 - Интерактивную **помощь** (on-line help).
 - **Руководства** по установке (Installation guide) и использованию программного продукта (user manual).

Проверка соответствия программы требованиям, осуществляемая путем наблюдения за ее работой в специальных, искусственно созданных ситуациях, выбранных определенным образом.

ПРОЦЕСС ТЕСТИРОВАНИЯ

Test
Design

- ✓ План
Тестирования
- ✓ Выбор стратегии

Тест-план

- ✓ Анализ Документации
- ✓ Подробное описание
тестов и оборудования

Тест кейсы

Test
Planning

Test
Execution

- ✓ Анализ результатов

Финальный отчет

Analysis &
Reporting

- ✓ Выполнение тестов
- ✓ Поддержка,
редактирование тестов
- ✓ Обнаружение и
документирование ошибок

Отчеты об ошибках
Журналы испытаний

План Тестирования (Test Plan) - это документ, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения.

Тест дизайн (Test Design) - это этап процесса тестирования ПО, на котором проектируются и создаются тестовые случаи (тест кейсы), в соответствии с определёнными ранее критериями качества и целями тестирования.

Тестовый случай (Test Case) - это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

Баг/Дефект Репорт (Bug Report) - это документ, описывающий ситуацию или последовательность действий приведшую к некорректной работе объекта тестирования, с указанием причин и ожидаемого результата.

Каждый тест определяет:

- свой набор исходных данных и условий для запуска программы;
- набор ожидаемых результатов работы программы.

КАЧЕСТВО ПО

Для того, что бы понять, что продукт соответствует требованиям пользователя и/или заказчика применяют верификацию и валидацию.

Верификация отвечает на вопрос: «Соответствует ли продукт требованиям?», а валидация: «Можно ли использовать продукт для определенных целей?»

Верификация – это процесс оценки системы или её компонентов с целью определения удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа. Т.е. выполняются ли наши цели, сроки, задачи по разработке проекта, определенные в начале текущей фазы.

Валидация – это определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе.

КЛАССИФИКАЦИЯ ТЕСТИРОВАНИЯ ПО

Статическое тестирование (static testing) - это процесс анализа самой разработки программного обеспечения, иными словами – это тестирование без запуска программы (проверка кода, требований, функциональной спецификации, архитектуры, дизайна и т.д.)

Примерами ошибок, которые потенциально можно выявить с помощью автоматического статического тестирования, могут быть:

- утечки ресурсов (утечки памяти, неосвобождаемые файловые дескрипторы и т.д.)
- возможность переполнения буфера (buffer overflows)
- ситуации частичной (неполной) обработки ошибок

Динамическое тестирование (dynamic testing) - это тестовая деятельность, предусматривающая эксплуатацию (запуск) программного продукта.

- Оно делится на несколько подтипов: тестирование белого ящика, тестирование черного ящика, а иногда выделяют и тестирование серого ящика.
- Эта классификация уже относится к **методам тестирования**, т.е. **как именно** тестируют программу.

Сравнение методов

- Достоинство *статических методов* состоит в сравнительно небольшом количестве необходимых ресурсов. Однако их реализация может содержать непредсказуемый процент брака (нереализуемых путей). Кроме того, в этих системах переход от покрывающего множества путей к полной системе тестов пользователь должен осуществить вручную (трудоемко).
- *Динамические методы* требуют значительно больших ресурсов как при разработке, так и при эксплуатации, однако увеличение затрат происходит, в основном, за счет разработки и эксплуатации аппарата определения реализуемости пути (символический интерпретатор, решатель неравенств). Достоинство этих методов заключается в том, что их продукция имеет некоторый качественный уровень - реализуемость путей. Методы реализуемых путей дают самый лучший результат.

МЕТОДЫ ТЕСТИРОВАНИЯ

Метод белого ящика (white-box testing, glass-box testing) – тестирование, при котором тестировщик *имеет доступ к коду*. Его еще называют *тестированием стеклянного ящика* или *тестированием прозрачного ящика*..

Тесты основаны на **знании кода приложения и его внутренних механизмов**.

Метод белого ящика часто используется на стадии, **когда приложение ещё не собрано воедино**, но необходимо проверить каждый из его компонентов, модулей, процедур и подпрограмм.

МЕТОДЫ ТЕСТИРОВАНИЯ

Метод чёрного ящика (black-box testing) заключается в том, что тестировщик имеет доступ к ПО только через те же интерфейсы, что и заказчик или пользователь.

Тестирование чёрного ящика ведётся с использованием **спецификаций или иных документов**, описывающих требования к системе на основе применения пользовательского интерфейса для ввода входных и получения выходных данных.

Цель данного метода – **проверить работу всех функций** приложения на соответствие функциональным требованиям.

Метод серого ящика (gray box testing) – совокупность подходов из методов белого и чёрного ящика.

Этот метод, как правило, используется при тестировании веб-приложений, когда тестировщик знает принципы функционирования технологий, на которых построено приложение, но может не видеть кода самого приложения.