



Компания хороших идей

Bellintegrator

Классификация тестирования

Докладчик: Капустина Оксана

План занятия

1. Уровни тестирования
2. Виды тестирования
3. Типы тестирования



Уровни тестирования



Уровни тестирования

- Компонентное или Модульное тестирование (Component Testing or Unit Testing)
- Интеграционное тестирование (Integration Testing)
- Системное тестирование (System Testing)
- Приемочное тестирование (Acceptance Testing)

Модульное тестирование

Входные требования - Архитектура компонентов или модель “нижнего уровня” системы (Component Design или Low Level Design)

Объект тестирования - Разработанные компоненты

Цель модульного тестирования - изолировать отдельные части программы, протестировать их и показать, что в отдельности они работоспособны.

Интеграционное тестирование

Входные требования - Архитектура системы или модель “верхнего уровня” системы (System Design или High Level Design)

Объект тестирования - Собранная из компонентов система или подсистема

Целью интеграционного тестирования выявление ошибок взаимодействия между модулями программы

Интеграционное тестирование

Подходы к интеграционному тестированию:

- **Снизу вверх** (*Bottom Up Integration*)
- **Сверху вниз** (*Top Down Integration*)
- **Большой взрыв** ("*Big Bang*" *Integration*)

Интеграционное тестирование

Уровни интеграционного тестирования:

- **Компонентный** (*Component Integration testing*)
- **Системный** (*System Integration Testing*)

Системное тестирование

Входные требования - Системные спецификации
(System Specification)

Объект тестирования - Разработанная система

Цель системного тестирования выявление ошибок, связанных с работой системы в целом

Системное тестирование

Этапы системного тестирования:

- **Альфа-тестирование**
- **Бета-тестирование**

Приемочное тестирование

Входные требования - Требования (Requirements)

Объект тестирования - Разработанная система

Целью приемочного тестирования является оценка готовности системы для ее выпуска

Приемочное тестирование

Пользовательское тестирование (UAT)

это тестирование, которое проводят бизнес пользователи, с целью определить пригодность системы для внедрения.

Проблемы UAT:

- Проводят не тестировщики;
- Нет желания проводить непонятную работу;
- Непонимание пользователями деталей работы системы;
- Мало времени;
- Сложность взаимодействия.

Виды тестирования



Виды тестирования

- По объекту тестирования
- По исполняемости кода
- По доступу к коду
- По степени автоматизации
- Связанное с изменениями
- По признаку позитивности сценариев
- По плану/без плана тестирования

По объекту тестирования

Функциональное тестирование

это проверка соответствия разработанных компонентов установленным требованиям.

Проводится в двух аспектах:

- требования
- бизнес-процессы

По объекту тестирования

Тестирование производительности

это автоматизированное тестирование, имитирующее работу определенного количества бизнес пользователей на каком либо общем (разделяемом ими) ресурсе.

Типы:

- Нагрузочное тестирование
- Стресс-тестирование
- Объемное тестирование
- Тестирование стабильности

По объекту тестирования

Конфигурационное тестирование

направленно на проверку работы программного обеспечения при различных конфигурациях системы.

Цель - проверка объекта тестирования на совместимость с объявленным в спецификации оборудованием, операционными системами и программными продуктами.

По объекту тестирования

Юзабилити тестирование

это вид тестирования, направленный на установление степени удобства использования, обучаемости, понятности и привлекательности для пользователей разрабатываемого продукта в контексте заданных условий.

По объекту тестирования

Тестирование интерфейса пользователя

это проверка реакции элементов интерфейса на действия пользователя и того, как приложение обрабатывает действия, связанные с использованием клавиатуры и мыши, либо тач-действий.

По объекту тестирования

Тестирование безопасности

это оценка уязвимости программного обеспечения к различным атакам (вирусы, атаки хакеров, несанкционированный доступ к конфиденциальным данным).

Цель:

- Проверить, что пользователь может обращаться только к тем функциям или данным, к которым он имеет доступ;
- Проверить, что только тем пользователям, которые вошли в систему (приложение), разрешено проводить различные операции.

По объекту тестирования

Тестирование локализации

это проверка правильности и корректности перевода элементов интерфейса пользователя, проверка правильности перевода системных сообщений и ошибок, проверка перевода раздела "Помощь"/"Справка" и сопроводительной документации.

Цель – убедиться, что приложение поддерживает многоязыковый интерфейс и функции.

По исполняемости кода

- **Статическое тестирование**

процедура выявления дефектов в продукте без прогона программного кода, т.е. проверка программы без запуска.

- **Динамическое тестирование**

процесс тестирования, производимый над работающей системой или подсистемой.

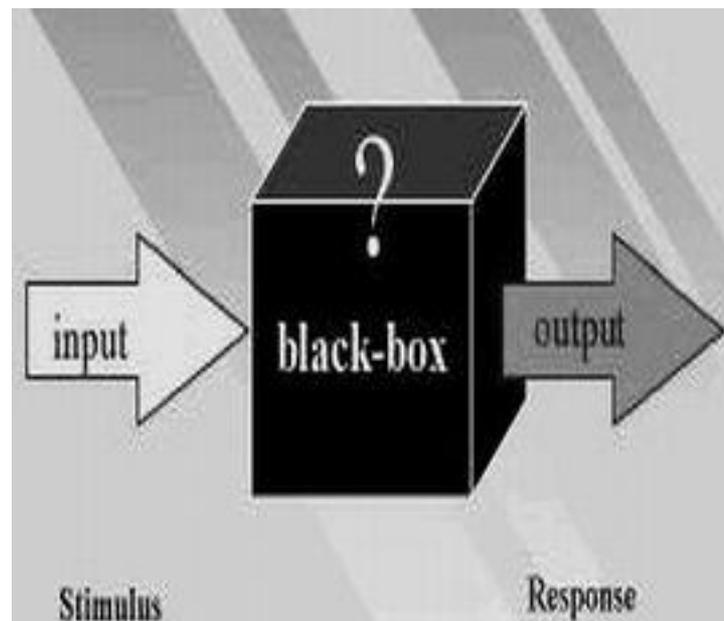
По доступу к коду

- **Тестирование черного ящика (black box)**
- **Тестирование белого ящика (white box)**
- **Тестирование серого ящика (grey box)**

По доступу к коду

Тестирование черного ящика (black box)

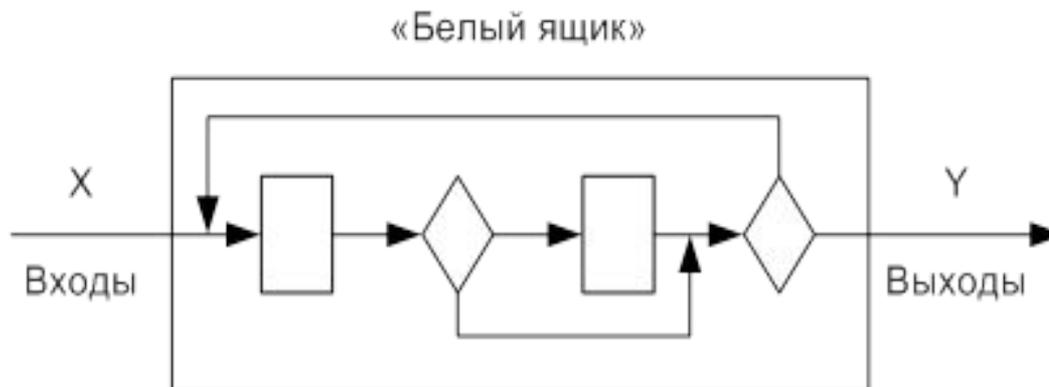
Тестирование через интерфейс пользователя не имея доступа к исходному коду продукта, на основании документации.



По доступу к коду

Тестирование белого ящика (white box)

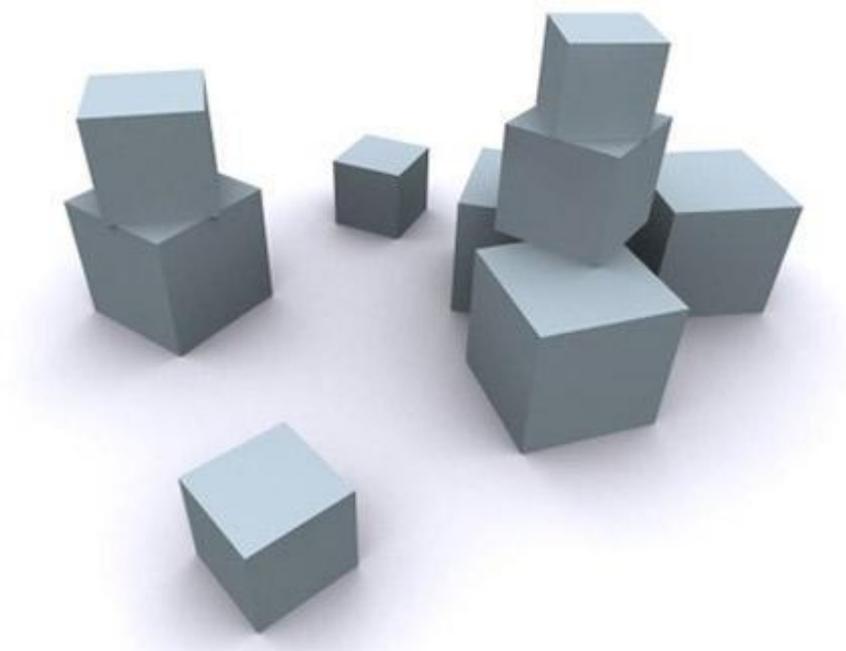
Тестирование, при котором тестировщик должен обладать знаниями о внутренних механизмах работы продукта и иметь доступ к исходному коду приложения



По доступу к коду

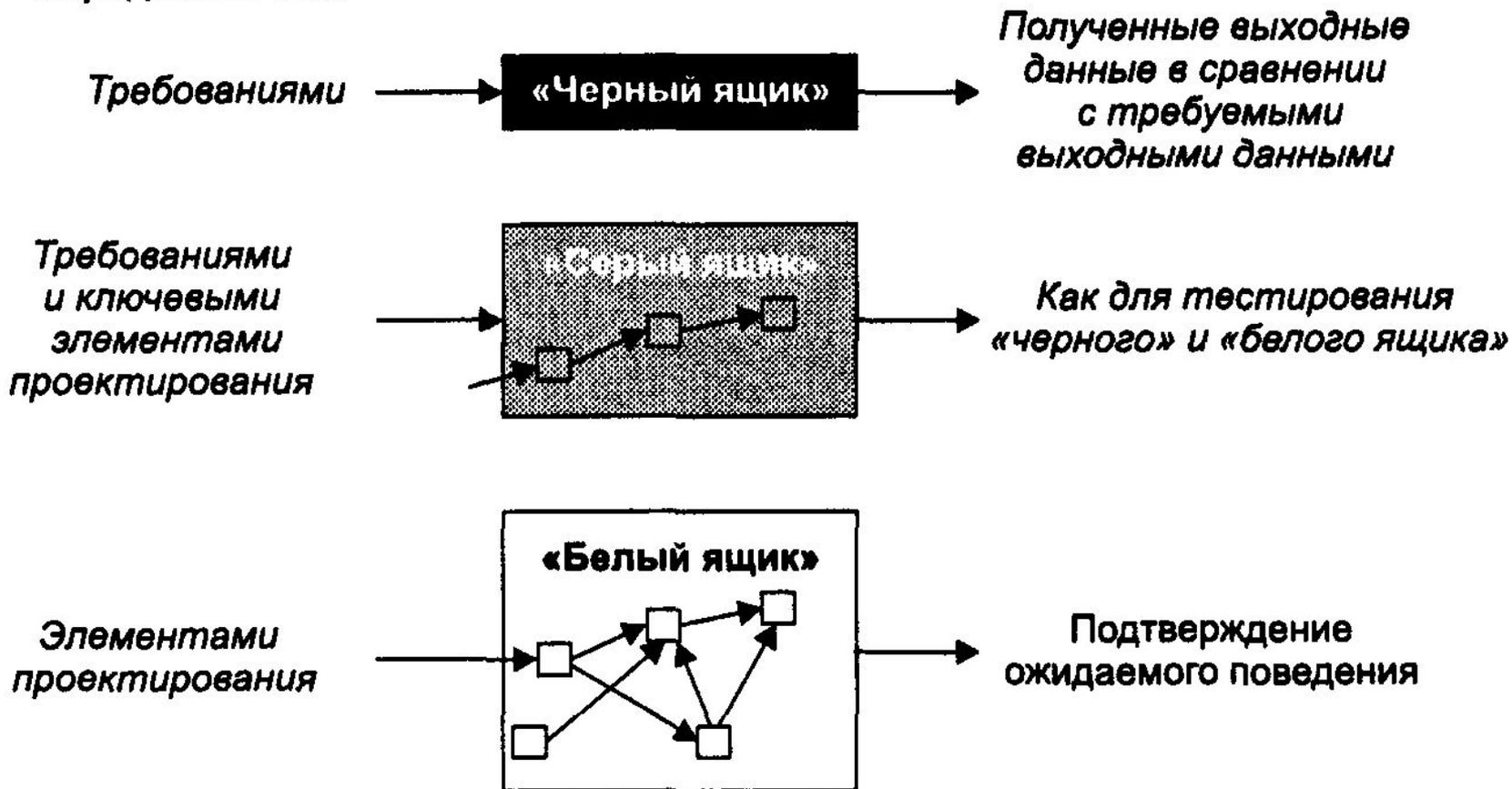
Тестирование серого ящика (grey box)

Тестируются как требования к системе на основании документации, так и особенности реализации.
Объединение тестирования черного и белого ящиков.



По доступу к коду

**Входные данные
определяются...**



По степени автоматизации

- **Ручное тестирование (manual testing)**
- **Автоматизированное тестирование (automated testing)**
- **Полуавтоматизированное тестирование (semi automated testing)**

Связанные с изменениями

- **Тестирование сборки (Build Verification Test)**
- **Дымовое тестирование (Smoke Testing)**
- **Санитарное тестирование или проверка согласованности/исправности (Sanity Testing)**
- **Регрессионное тестирование (Regression Testing)**

По признаку позитивности сценариев

- **Позитивное тестирование (positive testing)**
тестирование на данных или сценариях, которые соответствуют ожидаемому поведению системы.
- **Негативное тестирование (negative testing)**
тестирование на данных или сценариях, которые соответствуют нештатному поведению тестируемой системы.

Тестирование по плану

Сценарное тестирование - это подход к тестированию, который предполагает формальное описание тестов в форме чеклистов или тест кейсов.

Тестирование по плану

Преимущества:

- Требуется меньше навыков от тестера, который выполняет тесты
- Дает возможность легче оценить тестовое покрытие
- Стабильность и неизменяемость тест кейсов дает возможность проследить качество продукта с течением времени
- Обеспечивает более структурированный подход к тестированию
- Как правило, выполнение тестов не требует дополнительных исследований и общения с разработчиками, аналитиками, работы со спецификациями, что дает выигрыш по времени

Тестирование по плану

Недостатки:

- Тестирование по тест кейсам ограничивает свободу тестера
- Формирует привычки в тестировании, потому что тестер привыкает делать такие проверки, как описаны в тест кейсах
- Тест кейсы потихоньку теряют свою эффективность и перестают находить баги
- Постоянное выполнение одних и тех же тестов со временем утомляет
- Требует много времени на поддержку тест кейсов: обновление, удаление, исправление
- Тестер медленно развивает свое знание продукта и его контекста по сравнению с исследовательским тестированием

Тестирование без плана

- **Ad hoc testing**

это тестирование без подготовки, экспромтом. Такое тестирование может быть выполнено любым человеком.

- **Exploratory testing**

совмещение проектирования теста и его выполнения.

Тестирование без плана

Преимущества исследовательского тестирования:

- Хорошо развивает навыки, дает тестеру больше свободы
- Не требует времени для написания тестовых сценариев, все время уделяется тестированию

Тестирование без плана

Недостатки исследовательского тестирования:

- Неопытным тестерам в первое время бывает трудно из-за того, что они предоставлены сами себе, у них нет хотя бы примеров тестов, которые нужно выполнять
- Без тест кейсов сложнее оценить тестовое покрытие
- Изучение продукта отнимает время от непосредственного тестирования

Типы тестирования



Типы тестирования

- **Дымовой тест (Smoke test)**
- **Критический тест (Critical path test)**
- **Расширенный тест (Extended test)**

Типы тестирования

ДЫМОВОЙ ТЕСТ (Smoke test)

самый первый и короткий тест, проверяющий работу основной функциональности программного продукта

Типы тестирования

Шаги Smoke-теста должны быть последовательны, например:

1. Открыть Файл
2. Изменить файл
3. Сохранить файл
4. Закрыть файл
5. Открыть измененный файл
6. Проверить наличие сохраненных изменений

Типы тестирования

Критический тест (Critical path test)

проверяется основная функциональность программного продукта, критичная для конечного пользователя, при стандартном его использовании

Типы тестирования

Критический тест (Critical path test)

Примеры шагов:

Для текстовых полей:

- Оставить пустым
- Ввести 1 или несколько символом
- Пробелы вначале
- Пробелы в конце
- Пробелы в середине
- Только пробелы
- Максимальное количество символов
- Максимальное количество символов+1
- Другой регистр
- Специальные символы
- Другой язык

Типы тестирования

Критический тест (Critical path test)

Примеры шагов:

Для числовых полей:

- Оставить пустым
- Минимальное значение
- Минимальное значение-1
- Максимальное значение
- Максимальное значение+1
- Специальные символы
- Среднее значение из допустимого интервала
- Несколько чисел
- Отрицательное значение
- Ноль вначале
- Пробелы вначале или в конце
- Ноль после разделяющего символа для десятичных значений

Типы тестирования

Критический тест (Critical path test)

Примеры шагов:

Для даты:

- Оставить пустым
- Минимальное значение
- Минимальное значение-1
- Максимальное значение
- Максимальное значение+1
- Специальные символы
- Текст
- Числовые значения
- Нулевые значения
- Другой разделитель даты
- Февраль, 28,29,30
- Нули перед значениями (01/02/2007)

Типы тестирования

Расширенный тест (Extended test)

проверяется нестандартное использование программного продукта.

Типы тестирования

Расширенный тест (Extended test)

Примеры шагов:

Для Web-приложения, например, таковы:

- Несколько пользователей работающие с приложением одновременно с разных удаленных компьютеров
- Несколько пользователей, работающих одновременно и затрагивающих разные области приложения
- Большое число удаленных пользователей

Спасибо за внимание!