По знанию системы:

Метод тестирования черного ящика – когда мы тестируем, но не знаем код, белый – когда имеем, серый – совмещенный режим.

По уровню тестирования:

Требования к ПО – Системный тест

Архитектура – Интеграционный тест

Спецификация модуля – Модульный тест

Системное тестирование:

* Проверка соответствия установленным требованиям
* Выявление дефектов, связанных с работой системы в целом:
* Отсутствующая функциональность
* Неверное использование ресурсов системы
* Непредусмотренные комбинации данных пользовательского уровня
* Несовместимость с окружением
* Непредусмотренные сценарии использования
* Неудобство в применении и т.п.
* Метод «Черного ящика»

Интеграционное тестирование:

* Проверка связи между компонентами, а также взаимодействия с различными частями системы (ОС, оборудование и т.д.)
* Поиск дефектов, связанных с ошибками в реализации и интерпретации взаимодействия между модулями
* Требует создания тестового окружения
* Метод «Белого ящика»

Модульное тестирование:

* Проверка отдельно взятых модулей, функций или классов
* Выявление локализованных в модуле ошибок в алгоритмах или реализации алгоритмов
* Требует создания тестового окружения
* Метод «Белого ящика»

По методам сборки модулей:

Монолитный

Инкрементальный (два вида: сверху-вниз и снизу-вверх)

Сравнение методов

Монолитное:

- Большие трудозатраты, связанные со сложностью идентификации ошибок

+ Большие возможности распараллеливания работ

Инкрементальное:

- Требует затрат на разработку заглушек и драйверов

+ Связано с меньшей трудоемкостью идентификации ошибок за счет локализации тестируемого кода

Недостатки Сверху-вниз

* Проблема разработки достаточно интеллектуальных заглушек, т.е. заглушек, способных к использованию различных режимов работы
* Сложность организации и работы среды для реализации исполнения модулей в нужной последовательности
* Параллельная разработка модулей верхним и нижних модулей не всегда приводит к эффективной реализации модулей из-за подстройки нижних уровней к уже протестированным верхним

Недостатки Снизу-вверх

* Запаздывание проверки концептуальных особенностей тестируемого комплекса

Комбинирование уровней тестирования

* В каждом конкретном проекте должны быть определены задачи, ресурсы и технологии для каждого уровня тестирования
* Задача тестировщика и менеджеров – оптимально распределить ресурсы между тремя уровнями тестирования так, чтобы каждый из возможных типов дефектов был «адресован»
* Например, перенос усилий на поиск фиксированного типа дефектов из области системного в область модульного может сэкономить время и деньги тестирования

По степени автоматизации:

Ручное

Автоматизированное

Автоматическое

По целям:

Функциональное

Не функциональное: производительности, инсталляционное, конфигурационное, безопасности, локализации, удобства.

Функциональное тестирование – рассматривает заранее указанное поведение и основывается на анализе спецификаций функциональности компонента или системы в целом. Может приводится в аспекте требований и бизнес процессов. Может приводится на всех уровнях тестирования.

Тестирование производительности

Это автоматизированное тестирование, имитирующее работу определенного количества бизнес пользователей на каком-либо общем ресурсе.

Ящики

Виды

Инкриментальные

Сверху вниз

Комбинирование

Функциональное нефункциональное тестирование

Тестирование производительности