

Классификация тестирования по уровням

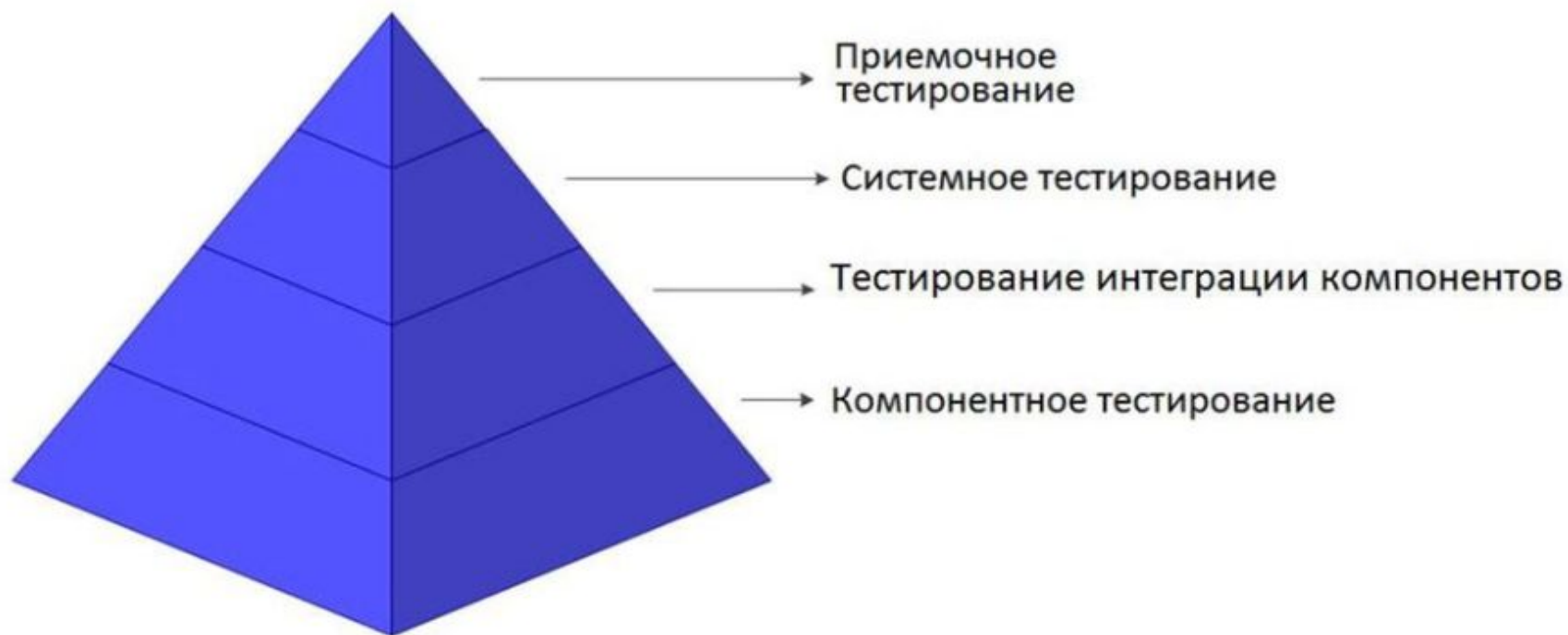
1. Модульное тестирование
2. Интеграционное тестирование
3. Системное тестирование
4. Выходное тестирование
5. Приемочное тестирование
6. Альфа- и бета-тестирование

Классификация тестирования по уровням

Существует несколько уровней тестирования, позволяющих полностью проверить программный продукт и выявить максимальное количество ошибок: **модульное, интеграционное, системное, выходное, приемочное**. Каждый уровень имеет свои цели и компоненты.

Уровни тестирования

Процесс тестирования включает в себя следующие уровни:



Модульное тестирование

Модульное тестирование — это тестирование программы на уровне отдельно взятых модулей, функций или классов.

Цель модульного тестирования состоит в выявлении ошибок в реализации алгоритмов, а также в определении степени готовности системы к переходу на следующий уровень разработки и тестирования.

Модульное тестирование проводится по принципу «белого ящика», т.е. основывается на знании внутренней структуры программы, и часто включает те или иные методы анализа покрытия кода.

Модульное тестирование проводится непосредственно разработчиком программного обеспечения и позволяет проверять все внутренние структуры и потоки данных в каждом модуле. Этот вид тестирования является частью этапа разработки.

Модульное тестирование

На *уровне модульного тестирования* проще всего обнаружить дефекты, связанные с алгоритмическими ошибками и ошибками кодирования алгоритмов, с использованием локальных переменных и ресурсов. Ошибки, связанные с неверной трактовкой данных, некорректной реализацией интерфейсов, совместимостью, производительностью и т.п., обычно пропускаются на уровне модульного тестирования и выявляются на более поздних стадиях тестирования

Модульное тестирование

Модульное тестирование включает в себя:

- проверку программного кода с использованием некоторого инструментального средства для выявления синтаксических ошибок в программном коде (синтаксическую проверку);
- проверку кода на соответствие стандартам кодирования (например, соответствия оформления программного кода требованиям организации-разработчика);

• технический обзор программного кода

Модульное тестирование

При выполнении модульного тестирования можно использовать технологию либо структурного, либо функционального тестирования или и ту, и другую. Структурное тестирование является одним из видов тестирования «белого ящика». Его главная идея — правильный выбор тестируемого программного пути. В противоположность ему функциональное тестирование относится к категории тестирования «черного ящика». Каждая функция программы тестируется путем ввода ее входных данных и анализа выходных. При этом внутренняя структура программы учитывается очень редко. После успешного завершения модульного тестирования все измененные модули и наборы тестов сохраняются в базе данных проекта.

Интеграционное тестирование

Интеграционное тестирование проводится для проверки совместной работы отдельных модулей и предшествует тестированию всей системы как единого целого. **Интеграционное тестирование** — это тестирование части системы, состоящей из двух и более модулей. Основная задача интеграционного тестирования — поиск ошибок в реализации и интерпретации интерфейсного взаимодействия между модулями.

Интеграционное тестирование

Элементами интеграционного тестирования являются:

- проверка функциональности, т.е. проверка соответствия отдельных функций, выполняемых связанными модулями, функциям, заданным в спецификациях требований;
- проверка наличия и корректности промежуточных результатов;
- проверка корректности передачи информации между модулями (проверка интеграции).

Интеграционное тестирование

С технологической точки зрения ***интеграционное тестирование*** представляет собой количественно развитие модульного, поскольку также, как и модульное тестирование, оперирует интерфейсами модулей и подсистем и требует создания тестового окружения, включая заглушки на месте отсутствующих модулей. Основная разница между модульным и интеграционным тестированием состоит в целях, т.е. в типах обнаруживаемых дефектов. Это определяет стратегию выбора входных данных и методов анализа.

Интеграционное тестирование

Интеграционное тестирование ведется итерационно, с постепенным подключением модулей и подсистем. Оно осуществляется независимым тестировщиком.

Ошибки, выявленные в ходе интеграционного тестирования, заносятся в базу данных проекта. Результаты интеграционного тестирования включаются в отчет о ходе тестирования при завершении цикла тестирования.

Системное тестирование

Системное тестирование проводится независимым тестировщиком при условии успешного завершения интеграционного тестирования. **Системное тестирование** качественно отличается от интеграционного и модульного уровней, рассматривает тестируемую систему в целом и оперирует на уровне пользовательских интерфейсов, в отличие от последних фаз интеграционного тестирования, которое оперирует на уровне интерфейсов модулей.

Системное тестирование

Основная задача системного тестирования — выявление проблем, связанных с работой системы в целом (например, неверное использование ресурсов системы, несовместимость с окружением, непредусмотренные сценарии использования, отсутствующая или неверная функциональность, неудобство в применении и т.п.).

Системное тестирование производится над проектом в целом с помощью метода «черного ящика», т.е. структура программы не имеет никакого значения, для проверки доступны только входы и выходы, видимые пользователю.

Тестированию подлежат коды и

Системное тестирование

Категории тестов системного тестирования:

- полнота решения функциональных задач;
- корректность использования ресурсов (утечка памяти, возврат ресурсов);
- оценка производительности;
- эффективность защиты от искажения данных и некорректных действий;
- проверка инсталляции и конфигурации на разных платформах;
- корректность документации.

Системное тестирование

Объемы данных, используемых на этом уровне тестирования, таковы, что более эффективным подходом является полная или частичная автоматизация тестирования, что приводит к необходимости создания гораздо более сложной тестовой системы, чем система тестирования, применяемая на уровне тестирования модулей или их комбинаций.

Ошибки, выявленные при системном тестировании, заносятся в базу данных проекта. Результаты системного тестирования включаются в отчет о ходе тестирования.

Выходное тестирование

Выходное тестирование осуществляется с целью проверки готовности программного обеспечения для поставки заказчику/пользователям. Это завершающий этап тестирования, проводимый независимым тестировщиком, включающий в себя проверку на корректность инструкций по установке, а также проверку комплектности документации. Ошибки, выявленные при выходном тестировании, заносятся в базу данных проекта. При успешном завершении выходного тестирования программного обеспечения заказчику поставляется программный продукт вместе с отчетом о результатах тестирования.

Приемочное тестирование

Приемочное тестирование проводится организацией, отвечающей за инсталляцию, сопровождение программной системы и обучение конечного пользователя. Это последний уровень тестирования, после которого продукт вводится в эксплуатацию.

Тестирование первых четырех уровней проводится внутри организации-разработчика, а приемочное тестирование выполняется совместно с представителем заказчика. Тестирование первого уровня проводит разработчик программного обеспечения на этапе разработки, а тестирование остальных уровней осуществляют независимые тестировщики.

Альфа-тестирование и бета-тестирование

Существуют еще две разновидности тестирования, отличающиеся друг от друга временем исполнения, — альфа-тестирование и бета-тестирование.

Альфа-тестирование — это реальная работа с программным обеспечением, проводимая потенциальными пользователями или заказчиками, либо имитация реальной работы разработчиками.

Чаще всего альфа-тестирование проводится на ранней стадии разработки продукта, но уже когда реализована вся или почти вся функциональность системы. В некоторых случаях альфа-тестирование может применяться для законченного продукта в качестве внутреннего приемочного тестирования.

Альфа-тестирование может выполняться с использованием отладчика или какого-либо инструментария, который помогает быстро выявлять ошибки. Обнаруженные ошибки могут быть переданы тестировщикам для дополнительного исследования.

Альфа-тестирование и бета-тестирование

Бета-тестирование — это уже интенсивное использование почти готовой версии программного обеспечения с полным набором запланированных функций. Целью его является выявление максимального числа ошибок в работе программного обеспечения для их последующего устранения перед окончательным выходом продукта на рынок, к массовому потребителю. В отличие от альфа-тестирования, проводимого силами разработчиков или тестировщиков, бета-тестирование предполагает привлечение сторонних пользователей, которым доступна упомянутая предварительная версия продукта (бета-версия).

Альфа-тестирование и бета-тестирование

Кроме того, **бета-тестирование** может использоваться в рекламных целях как часть стратегии продвижения продукта на рынок (например, бесплатная раздача бета-версий позволяет привлечь широкое внимание пользователей к окончательной дорогой версии продукта), а также для получения предварительных отзывов о нем от широкого круга будущих пользователей.

Бета-версия не является финальной версией продукта, поэтому разработчик не гарантирует полное отсутствие ошибок, которые могут нарушить работу компьютера и/или привести к потере данных.

Задание 1. Заполните таблицу:

Классификация тестирования по уровням

Уровни тестирования	Цель тестирования	Методы тестирования	Объекты тестирования	Элементы тестирования	Кем проводится

Задание 2. Ответьте на вопросы

1. Дайте определение модульного тестирования.
2. Что такое интеграционное тестирование?
3. В чем заключается разница между модульным и интеграционным тестированием?
4. Чем отличается системное тестирование от интеграционного и модульного уровней?
5. Что такое альфа- и бета-тестирование?