**ISTQB основные теоретические понятия**

Программа обучения Базового уровня

1. **Основы тестирования**

**Ошибка > Дефект > Отказ**

Человек может сделать ошибку (просчет), которая порождает дефект (недочет, помеху) в программном коде или документе. Если код с дефектом выполнен, то система может быть не в состоянии сделать то, что должна делать (или сделать то, что от нее не ожидают), порождая отказ.

Тестирование - это возможный способ оценки качества программного обеспечения в терминах найденных дефектов, как для функциональных требований, так и для нефункциональных требований. Тестирование может породить уверенность в качестве программного обеспечения, если не найдены или найдено немного дефектов. Когда во время тестирования находятся ошибки, качество систем программного обеспечения повышается, если эти дефекты исправлены. Тестирование также является деятельностью по обеспечению качества.

**Когда заканчивать тестирование?** Для принятия решения о достаточном объеме тестирования, необходимо принимать во внимание уровень рисков, включая технические риски, риски безопасности и бизнес риски, а так же проектные ограничения, такие как время и бюджет

Тестирование должно предоставить достаточную информацию заинтересованным лицам, чтобы принять обоснованные решения о передаче программного обеспечения или системы, прошедшей тестирование, на следующий шаг разработки или передачи клиентам.

**Активности в тестировании** существуют как до, так и после выполнения самих тестов. В эти активности входят планирование и управление, выбор тестовых условий, разработка и выполнение тестовых сценариев, проверка результатов, оценка критериев выхода, создание отчетов о процессе тестирования и об испытываемой системе и закрытие после того, как фаза тестирования была выполнена. Тестирование также включает рецензирование документации (включая исходный код) и проведение статического анализа.

**Цели тестирования:**

• Обнаружение дефектов

• Повышение уверенности в уровне качества

• Предоставление информации для принятия решений

• Предотвращение дефектов

Также в тестировании на этапе разработки (таком, как компонентное, интеграционное и системное тестирование), **основная цель** может заключаться в том, чтобы вызвать как можно больше отказов, чтобы дефекты в программном обеспечении были идентифицированы и могли быть исправлены.

В приемочном тестировании **основная цель** может состоять в том, чтобы подтвердить, что система работает, как ожидалось и повысить уверенность в том, что она удовлетворяет требованиям.

Тестирование в период сопровождения в основном заключается в проверке отсутствия новых дефектов, которые могли попасть во время разработки изменений. Во время эксплуатационного тестирования **основная цель** может заключаться в том, чтобы оценить системные характеристики, такие как надежность или доступность.

**Тестирование и отладка**

Динамическое тестирование может выявить отказы, вызванные дефектами

Отладка – это действия разработчиков, которые находят, анализируют и устраняют причину отказа.

*Ответственность за тестирование обычно несут тестировщики, а за отладку - разработчики.*

**Семь принципов тестирования:**

1. Тестирование демонстрирует наличие дефектов
2. Исчерпывающее тестирование недостижимо
3. Раннее тестирование
4. Скопление дефектов
5. Парадокс пестицида
6. Тестирование зависит от контекста
7. Заблуждение об отсутствии ошибок (Обнаружение и исправление дефектов не помогут, если созданная система не подходит пользователю и не удовлетворяет его ожиданиям и потребностям.)

**Основной процесс тестирования состоит из следующих направлений деятельности**:

• Планирование и управление

Планирование тестирования - это действия, направленные на определение целей тестирования и описание задач тестирования для достижения этих целей и миссии.

Управление тестированием – это постоянное сопоставление текущего положения дел с планом и отчетность о состоянии дел, включая отклонения от плана.

• Анализ и проектирование

Анализ и проектирование тестов - это деятельность, во время которой общие цели тестирования материализуются в тестовые условия и тестовые сценарии.

• Внедрение и реализация

Реализация и выполнение тестов – это деятельность, где процедуры тестирования или автоматизированные сценарии задаются последовательностью тестовых сценариев, а также собирается любая информация, необходимая для выполнения тестов, разворачивается окружающая среда, и запускаются тесты.

• Оценка критериев выхода и создание отчетов

Оценка критериев выхода - это деятельность, где выполнение тестов оценивается согласно определенным целям. Она должна быть выполнена для каждого уровня тестирования. Написание итогового отчета о тестировании для заинтересованных лиц.

• Действия по завершению тестов

Действия по завершению тестирования собирают данные о завершенных испытаниях для объединения опыта, тестового обеспечения, фактов и цифр. Действия по завершению тестирования происходят на тех этапах проекта, когда система программного обеспечения выпущена, тестирование завершено (или прервано), этап был завершен, или релиз по сопровождению был закончен.

1. **Место тестирования в жизненном цикле (ЖЦ) разработки ПО**

**Модели разработки ПО:**

* V-модель (Последовательная модель разработки), используют четыре уровня тестирования: компонентное, интеграционное, системное и приемочное.
* Итеративно-инкрементные модели разработки (Итеративно-инкрементный процесс разработки – это процесс, основывающийся на требованиях, дизайне, сборке и тестировании системы, созданной в результате серии коротких циклов разработки)
* Waterfall
* Спиральная (используется в экспериментальной разработке)

**Уровни тестирования**

* Компонентное тестирование

Компонентное тестирование (также известное как модульное) занимается поиском дефектов и верификацией функционирования программных модулей, программ, объектов, классов и т.п., которые можно протестировать изолированно. В процессе могут быть использованы заглушки, драйвера и эмуляторы. На практике компонентное тестирование обычно производится разработчиками, которые пишут код. Дефекты обычно исправляются сразу после того, как становятся известны, без занесения их в базу дефектов. Один из подходов к компонентному тестированию – составить автоматизированные тестовые сценарии до кодирования. Это называется разработкой, управляемой тестированием.

* Интеграционное тестирование

Интеграционное тестирование проверяет интерфейсы между компонентами, взаимодействие различных частей системы, таких как операционная системы, файловая система, аппаратное обеспечение, и интерфейсы между системами.

Интеграционное тестирование может состоять из одного или более уровней и может быть выполнено на тестовых объектах разного размера следующим образом:

1. Компонентное интеграционное тестирование проверяет взаимодействие между программными компонентами и производится после компонентного тестирования

2. Системное интеграционное тестирование проверяет взаимодействие между системами или между аппаратным обеспечением и может быть выполнено после системного тестирования. В этом случае, разработчики могут управлять только одной стороной интерфейса.

Тестирование специфичных нефункциональных характеристик ( например, производительности), может быть включено в интеграционное тестирование, наравне с функциональным.

На каждой стадии интеграции тестировщики концентрируют все внимание именно на интеграции как таковой. Например, если интегрируется модуль А с модулем В, они проверяют взаимодействие модулей, а не функциональность каждого из них, т.к. она должна быть проверена во время компонентного тестирования.

Подходы к интеграционному тестированию:

- Большой взрыв

- Снизу вверх

- Сверху вниз

* Системное тестирование

Системное тестирование сконцентрировано на поведении тестового объекта как целостной системы или продукта. Область тестирования должна быть четко определена.

Во время системного тестирования тестовое окружение должно быть как можно ближе к предполагаемому эксплуатационному окружению системы для минимизации риска пропуска отказов, связанных с эксплуатационным окружением системы.

Системное тестирование должно заниматься исследованием функциональных и нефункциональных требований к системе и качеством обрабатываемых данных.

Системное тестирование чаще всего выполняет независимая тестовая команда.

* Приемочное тестирование

Приемочным тестированием системы чаще всего занимаются заказчики или пользователи системы, а также другие заинтересованные лица.

**Основная цель** приемочного тестирования – проверка работоспособности системы, частей системы или отдельных нефункциональных характеристик системы. Поиск дефектов не является главной целью приемочного тестирования.

Приемочное тестирование оценивает готовность системы к развертыванию и использованию.

**Типичные виды приемочного тестирования:**

- Пользовательское приемочное тестирование

Обычно проверяет готовность системы для использования в бизнесе.

Эксплуатационное (приемочное) тестирование

Приемочное тестирование, проводимое системными администраторами, включает:

• Тестирование резервного копирования \ восстановления

• Восстановление после сбоев

• Управление пользователями

• Задачи сопровождения

• Задачи загрузки и миграции данных

• Периодическая проверка уязвимостей системы

Контрактное и правовое приемочное тестирование

Контрактное приемочное тестирование выполняется для проверки требований, предъявляемых контрактом к разрабатываемому ПО.

Приемочное тестирование на соответствие стандартам выполняется для проверки соответствия стандартам государственным, юридическим или стандартам безопасности.

Альфа и бета тестирование (или тестирование в условиях эксплуатации)

Разработчики рыночного, или коробочного, ПО часто хотят получить отзывы от потенциальных или существующих заказчиков до того, как начнется продажа продукта.

Альфа тестирование выполняется организацией, разрабатывающей продукт, но не группой разработчиков.

Бета тестирование, или тестирование в условиях эксплуатации, выполняется покупателями или потенциальными заказчиками на их собственных мощностях.

**Типы тестирования**