

Почему тестирование важно

Тестирование важно потому что все мы делаем ошибки.

«Делать ошибки — это то, что мы делаем лучше всего»

Риск — это фактор, который в будущем может привести к непредвиденным последствиям.



Тестирование ПО — рассмотрим тестирование как процесс:

1. тестирование включает в себя более чем одно действие
2. тестирование проходит через все части разработки ПО начиная с требований
3. тестирование бывает статическим и динамическим
4. планирование выполняется до и после тестирования
5. подготовка — мы должны выбрать что тестирование будет делать, выбираем условие и разрабатываем тестовые сценарии
6. оценка — как выполнение тестов удовлетворяет ПО и критериям

Цели тестирования:

1. Определить, что ПО удовлетворяет требованиям
2. Демонстрирование, что ПО удовлетворяет целям пользователя
3. Поиск дефектов

Дефект — ошибка в компоненте или системе, которая может привести к невозможности выполнению компонентов или системы их функции.

Принципы тестирования:

1. Тестирование показывает существующий дефект
2. Искрывающие тестирование невозможно
3. Раннее тестирование
4. Ошибки кластеризуются
5. Парадокс пестицидов (результаты тестирования должны быть инвариантны)
6. Тестирование контекстнозависимо
7. Отсутствие ошибок обманчиво

Основной процесс тестирования

Базовые шаги:

1. Планирование и контроль
2. Анализ и проектирование
3. Разработка и исполнение
4. Проверка выходных критериев и отчета
5. Действия связанные с покрытием тестом

Планирование и контроль

Во время этого этапа мы должны понимать цели тестирования. Необходимо понять цели и задачи заказчика, владельца продукции, самого проекта.

1. Политика тестирования. Дает нам правила тестирования.
2. Стратегия тестирования. Общий высокоуровневый подход к тестированию.

Анализ и проектирование

На данном этапе необходимо выявить условия тестирования (трансформируют цели в осязаемые условия тестирования и архитектуру тестов).

Выполняемые действия:

1. Оценка базы тестирования
2. Проектирование тестов
3. Оценка тестируемости системы
4. Проектирование тестового окружения (хранилища БД)

Разработка и исполнение

Переводим тестовые условия в реальный код (тестовые сценарии). Затем тестовые сценарии объединяются в тестовые наборы.

Проверка выходных критериев и отчета

Генерация отчетов

Уровни тестирования

Тестирование разделяется по нескольким уровням:

1. Тестирование компонентов(Модульное тестирование)



2. Интеграционное тестирование. Тестирование интерфейсов между компонентами взаимодействия различных частей системы. Существует несколько уровней интеграционного тестирования.

- a. Тестирование интеграции компонентов
- b. Тестирование интеграции систем
 - i. Интегрирование всего
 - ii. Интегрирование последовательно
- c. Подходы
 - i. Снизу вверх
 - ii. Сверху вниз
 - iii. По спецификации

Типы тестирования

1. Функциональное тестирование (Тестирование черного ящика)
2. Не функциональное тестирование (Тестирование характеристик продукта (нагрузочное, производительности и прочее))
3. Структурное тестирование (Тестирование белого ящика)
4. Тестирования изменений
 - a. Подтверждающее тестирование
 - b. Регрессионное тестирование

Статические техники тестирования

Формальные шаги процесса:

1. Планирование
2. Начало ревью
3. Подготовка отчета
4. Обсуждение
5. Переработка кода
6. Завершение

1. Статические техники тестирования

1. Провести статический анализ кода (профилировщик)
2. Провести анализ на стиль кода (взять код и требования к написанию кода, должно быть автоматически)
3. Провести ревью кода (берем код и зрительно просматриваем и ищем ошибки)

Бек "Экстремальное программирование. Разработка через тестирование"