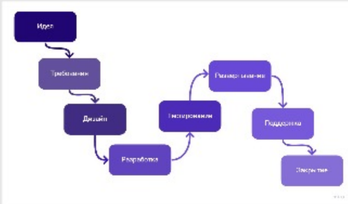
SDLC и STLC



Структура команды:

1. Разработчики
2. Аналитики
3. Тестировщики
4. Дизайнеры
5. Project Manager
6. DevOps

Анализ требований

Пользовательская история (User Story)
User story создается по шаблону:
"As a [personal], I [want to], [so that]."
«Как [тип клиента], [хочу то-то], [чтобы делать что-то]».

Похожий подход можно применить и при составлении тест-кейсов по паттерну **AAA (Arrange, Act, Assert).**

Пример:
Как менеджер, я хочу видеть, как продвигается работа у моих коллег, чтобы можно было составлять более точные отчеты о наших успехах и неудачах.

Характеристики качественных требований:
Завершенность, атомарность, непротиворечивость, недвусмысленность, выполнимость, обязательность, прослеживаемость, модифицируемость, проранжированность, корректность, проверяемость.

Планирование

Структура тест-плана
1. Цель
2. Области, подвергаемые тестированию
3. Области, не подвергаемые тестированию
4. Тестовая стратегия и подходы
5. Критерии. Чаще всего это критерии начала и завершения тестирования
6. Ресурсы: программные, аппаратные, человеческие, финансовые
7. Расписание
8. Роли и ответственность
9. Оценка рисков
10. Документация
11. Метрики - числовые характеристики показателей качества

Матрица трассируемости (traceability matrix, RTM) - двумерная таблица, описывающая связь двух сущностей (например, требований и тестовых сценариев). Таблица позволяет производить прямую и обратную трассировку от одной сущности к другой, обеспечивая таким образом возможность определения покрытия и оценки влияния предполагаемых изменений.

Пример матрицы трассируемости

	TC 1	TC 2	TC 3	TC 4
REQ 1		+		
REQ 2	+			
REQ 3		+	-	
REQ 4				+

Проектирование тестов

Эквивалентное разбиение
1. Определение позитивных и негативных классов
2. Выбор одного представителя для тестирования

Анализ граничных значений
1. Выбор граничных значений для границ классов
2. Нижняя граница-1, нижняя граница, верхняя граница, верхняя граница+1
3. Граница-1, граница, граница+1

Попарное тестирование
1. Тестирование всех комбинаций пар параметр-значение

Таблица принятия решений
1. В процессе создания таблицы, тестировщик определяет условия (входы) и результирующие действия системы (выходы).
2. Пары условий и действий образуют строки таблицы, при этом условия указываются сверху, а действия – снизу.

Предугадывание ошибок

Обязательные атрибуты чек-листа:
1. ID (авто) и заголовок
2. Название проверки
4. Статус проверки

Дополнительные атрибуты:
1. Ожидаемый результат

Обязательные атрибуты тест-кейса:
1. ID (авто) и заголовок
2. Шаги
3. Ожидаемый результат
4. Статус проверки

Дополнительные атрибуты:
1. Предусловия и постусловия
2. Тестовые данные
3. Вложения

Настройка окружений

Local	Можно проводить тестирование отдельных функций localhost, 127.0.0.1
Dev	Песочница для разработчиков dev.test.com
QA	Основное тестирование qa.test.com
Stage	Стабилизация, Регрессия stage.test.com
Pre-Prod	
Prod	Конечный пользователь test.com

Выполнение тестов

1. Генерация автотестов и тестовых данных для них
2. Генерация идей для тестирования различных модулей и типов приложений
3. Генерация тестов для Postman
4. и т.д.

Завершение тестирования

Атрибуты отчета о тестировании
1. Краткое описание отражает основные достижения, проблемы, выводы и рекомендации.
2. Команда тестировщиков
3. Описание процесса тестирования - перечень работ за конкретный период
4. Расписание - детализированное расписание работы команды тестировщиков и/или личные расписания участников команды.
5. Статистика по новым дефектам в виде таблицы
6. Список новых дефектов
7. Статистика по всем дефектам - таблица, содержащая данные обо всех дефектах за время проекта
8. Рекомендации
9. Приложения - фактические данные (графики и диаграммы)

Представь, что ты тестировщик программного обеспечения.

Перед тобой стоит задача проанализировать пользовательскую историю согласно характеристикам качественных требований: завершенность, атомарность, непротиворечивость, недвусмысленность, выполнимость, обязательность, прослеживаемость, модифицируемость, проранжированность, корректность, проверяемость.

В качестве результата пришли, какие неточности есть в требованиях. Особое внимание удели валидациям.

История для анализа:

Как покупатель,
Я хочу оплатить заказ с помощью карт Visa или MasterCard,
Чтобы совершить покупку.
Критерии приемки:
Пользователь должен ввести номер карты, срок действия, CVV, имя и фамилию держателя.
Номер карты должен состоять из 16 цифр.
Срок действия карты должен быть валидным.
CVV должен состоять из 3 цифр.
Имя и фамилия держателя должны быть введены латиницей.
Исключительные случаи:
Если данные введены неверно, отображается сообщение: "Проверьте корректность введенных данных".
При ошибке оплаты отображается сообщение: "Ошибка оплаты. Попробуйте другой способ или попробуйте позже".

Представь, что ты тестировщик программного обеспечения.

Перед тобой стоит задача создать тест-план для интернет магазина, в котором есть следующие модули: регистрация, логин, каталог, корзина, модуль оплаты, история заказов, страница "Контакты", "О компании", "Политика по использованию".

В качестве результата пришли шаблон тест-плана и его наполнение в виде таблицы. Особое внимание удели тестовой стратегии и типам тестирования, которые нужно применять.

Тест-план должен включать в себя следующие атрибуты:
1. Цель
2. Области, подвергаемые тестированию
3. Области, не подвергаемые тестированию
4. Тестовая стратегия и подходы
5. Критерии. Чаще всего это критерии начала и завершения тестирования
6. Ресурсы: программные, аппаратные, человеческие, финансовые
7. Расписание
8. Роли и ответственность
9. Оценка рисков
10. Документация
11. Метрики - числовые характеристики показателей качества

Классы эквивалентности

Представь, что ты тестировщик программного обеспечения. Перед тобой стоит задача определить позитивные, негативные классы, провести анализ граничных значений для этого требования:

Предположим, что для расчета скидки в онлайн-магазине, пользователь должен ввести в поле свой возраст.
Допускается ввод только чисел. Расчет скидки производится следующим образом:
Возраст покупателя от 0 до 17 лет - **скидка 50%**
Возраст покупателя от 18 до 54 лет - **скидка 25%**
Возраст покупателя от 55 и выше - **скидка 75%**

Результат оформи в виде таблицы.

Попарное тестирование

Представь, что ты тестировщик программного обеспечения. Сформируй таблицу для тестирования, используя технику тест-дизайна попарное тестирование или Pairwise Testing.

Объект тестирования:

Таблица принятия решений

Представь, что ты тестировщик программного обеспечения. Используя таблицу принятия решений создай примеры тестов в виде таблицы, которые нужно сделать для этих требований:

Предугадывание ошибок

Представь, что ты тестировщик программного обеспечения.

Используя технику тест-дизайна "предугадывание ошибок" предположи на что стоит обратить внимание при тестировании интернет-магазина со следующими модулями: регистрация, логин, каталог, корзина, модуль оплаты, история заказов, страница "Контакты", "О компании", "Политика по использованию".

Чек-лист

Представь, что ты тестировщик программного обеспечения.

Создай чек-лист для тестировании интернет-магазина со следующими модулями: регистрация, логин, каталог, корзина, модуль оплаты, история заказов, страница "Контакты", "О компании", "Политика по использованию".

В чек-листе должно быть как минимум 5 проверок на каждый модуль, а также тесты должны позитивные, негативные, функциональные и нефункциональные проверки.

Результат оформи в виде таблицы.

Тест-кейсы

Представь, что ты тестировщик программного обеспечения.

Создай тест-кейсы для тестировании интернет-магазина со следующими модулями: регистрация, логин, каталог, корзина, модуль оплаты, история заказов, страница "Контакты", "О компании", "Политика по использованию".

В тест-кейсах должны быть:
1. Подробные шаги
2. Ожидаемый результат для каждого шага
3. Статус проверки

Я хочу видеть по 3 тест-кейса для каждого модуля.

Результат оформи в виде таблицы.

Представь, что ты тестировщик программного обеспечения.

Создай отчет по результатам тестирования интернет-магазина со следующими модулями: регистрация, логин, каталог, корзина, модуль оплаты, история заказов, страница "Контакты", "О компании", "Политика по использованию".

Обязательные атрибуты отчета:
1. Краткое описание отражает основные достижения, проблемы, выводы и рекомендации.
2. Команда тестировщиков
3. Описание процесса тестирования - перечень работ за конкретный период
4. Расписание - детализированное расписание работы команды тестировщиков и/или личные расписания участников команды.
5. Статистика по новым дефектам в виде таблицы
6. Список новых дефектов
7. Статистика по всем дефектам - таблица, содержащая данные обо всех дефектах за время проекта
8. Рекомендации
9. Приложения - фактические данные (графики и диаграммы)

Результат оформи в виде таблицы.