***Виды тестовой документации?***

Баг-репорт – описание выявленного случая несоответствия производимого продукта требованиям, к нему выдвигаемым – ошибки или ее проявления. Он обязательно должен содержать следующие элементы:

Запрос на изменение (улучшение) – описание неявных/некритичных косвенных требований, которые не были учтены при планировании/реализации продукта, но несоблюдение, которых может вызвать неприятие у конечного потребителя. И пути/рекомендации по модификации продукта для соответствия им.

Стратегия тестирования (Test strategy) —Это относительно небольшой статический документ, который предшествует плану тестирования. Прежде чем писать объемный и подробный план, стоит формализовать некоторые базовые подходы к тестированию и убедиться, что все заинтересованные лица понимают одинаково, что и как будет тестироваться.

Стратегия включает:

* Роли и обязанности в команде тестирования
* Область тестирования
* Тестовые инструменты
* Тестовая среда
* График тестирования
* Сопутствующие риски

Тест-план (Test plan, план тестирования) – документ, описывающий весь объем работ по тестированию, начиная с описания объекта, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения. Ответственность за создание документа плана тестирования лежит на Test Lead или менеджере по тестированию. Содержание:

* Что надо тестировать?
* Что будете тестировать?
* Что не будем тестировать?
* Как будете тестировать?
* Когда будете тестировать?
* Критерии начала тестирования
* Приостановки
* Возобновления
* Критерии окончания тестирования
* Окружение тестируемой системы (описание программно-аппаратных средств)
* Необходимое для тестирования оборудование и программные средства (тестовый стенд и его конфигурация, программы для автоматизированного тестирования и т. д.)
* Риски и пути их разрешения

Тестовый сценарий — это описание начальных условий, входных данных, действий пользователя и ожидаемого результата. Создание тестовых сценариев обеспечивает полное покрытие тестами

* Тестовый набор/комплект (Test Suite) — Некоторый набор формализованных Test case, объединенных между собой по общему логическому признаку.
* Test case (тест-кейс) – Это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или ее части. Более строго — формализованное описание одной показательной проверки на соответствие требованиям прямым или косвенным. Обязательно должен содержать следующую информацию:
* Идея проверки.
* Описание проверяемого требования или проверяемой части требования.
* Используемое для проверки тестовое окружение.
* Шаги для приведения продукта в состояние, подлежащее проверке.
* Входные данные для использования при воспроизведении шагов.
* Ожидаемый результат.
* Прочую информацию, необходимую для проведения проверки.
* Чек-лист (лист проверок) – перечень формализованных Test case в упрощенном виде удобном для проведения проверок.
* Матрица покрытия требований (RTM — Requirements Traceability Matrix) - это документ, который связывает требования с тест-кейсами.

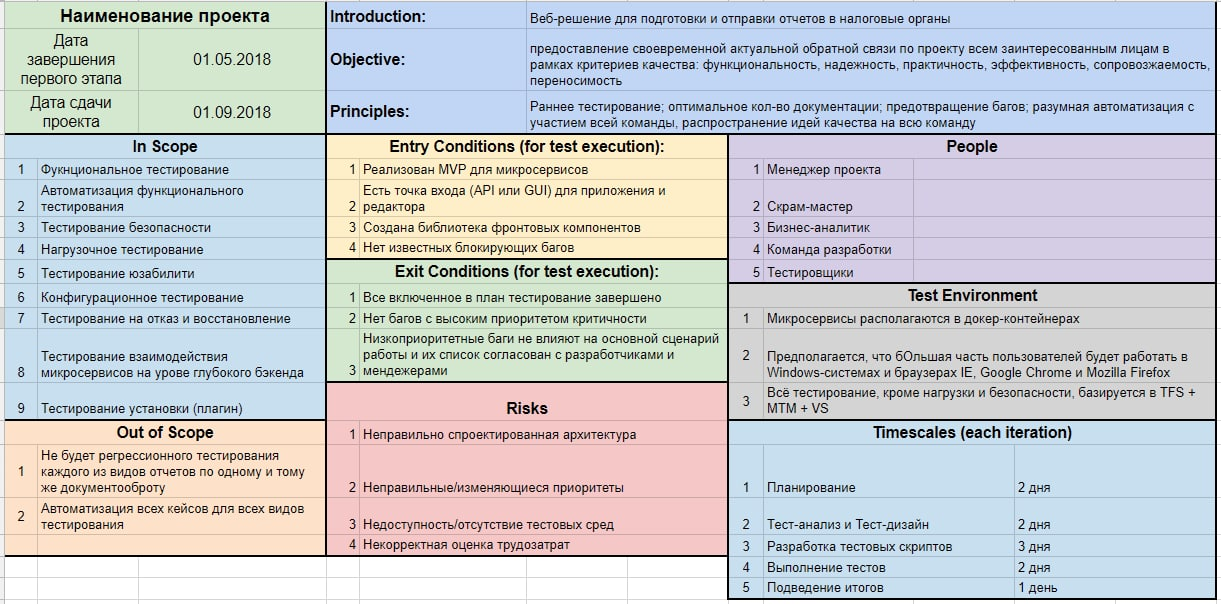
***Виды тест планов?***

Чаще всего на практике приходится сталкиваться со следующими видами тест планов:

* Мастер Тест План (Master Plan or Master Test Plan)
* Тест План (Test Plan), назовем его детальный тест план

Явное отличие Мастер Тест Плана от просто Тест Плана в том, что мастер тест план является более статичным в силу того, что содержит в себе высокоуровневую (High Level) информацию, которая не подвержена частому изменению в процессе тестирования и пересмотра требований. Сам же детальный тест план, который содержит более конкретную информацию по стратегии, видам тестировании, расписанию выполнения работ, является "живым" документом, который постоянно претерпевает изменения, отражающие реальное положение дел на проекте.

В повседневной жизни на проекте может быть один Мастер Тест План и несколько детальных тест планов, описывающих отдельные модули одного приложения.



***Что такое документ бизнес-требований (BRD)?***

BRD предоставляет полное бизнес-решение для проекта, включая документацию о потребностях и ожиданиях клиентов. BRD выполняет следующие задачи.

***Три основных категории дефектов:***

* Wrong: это указывает на несоответствие в требовании и реализации. Это подразумевает отклонение от данной спецификации.
* Missing: конечный продукт не имеет функции, соответствующей требованию. Это отличается от спецификаций и означает, что вы неправильно задокументировали требование.
* Extra: вы добавили функцию, которую клиент не запрашивал. Это снова отклонение от спецификации. И пользователям продукта может понравиться эта функция. Но это все еще дефект, потому что это не часть спецификаций.

## *Что такое утечка дефектов и релиз бага? (Bug Leakage & Bug Release)*

Релиз бага - это когда программное обеспечение или приложение передается группе тестирования, зная, что дефект присутствует в выпуске. При этом приоритет и серьезность ошибки низки, поскольку ошибка может быть удалена до окончательной передачи обслуживания.

Утечка бага - когда баг обнаруживается конечными пользователями или заказчиком, а не обнаруживается группой тестирования во время тестирования программного обеспечения.

**Инкрементальный vs Итерационный подход**

**Подход Инкрементальный** использует определенное количество шагов , и развитие идет от начала до конца в линейной траектории прогрессии.

Инкрементная разработка выполняется в несколько этапов: от проектирования, внедрения, тестирования / проверки, обслуживания. Они могут быть разбиты далее на подэтапы, но большинство инкрементальных моделей следует той же схеме. [**Модель**](http://en.wikipedia.org/wiki/Waterfall_model) « [**Водопад»**](http://en.wikipedia.org/wiki/Waterfall_model) - это традиционный метод постепенного развития.

**Итерационный подход** не имеет определенное количество шагов, а развитие осуществляется в циклах.

Итеративная разработка меньше связана с отслеживанием прогресса отдельных функций. Вместо этого основное внимание уделяется созданию рабочего прототипа и добавлению функций в циклы разработки, где этапы инкрементной разработки выполняются для каждого цикла. **[Agile Modeling](http://en.wikipedia.org/wiki/Agile_Modeling" \t "_blank)** - это типичный итеративный подход.

## *В чем отличие Kanban от Scrum?*

В отличие от Scrum, в команде канбан отсутствуют роли владельца продукта и модератора, а процесс разработки делится не на универсальные спринты, а на стадии выполнения задач («Планируется», «Разрабатывается», «Тестируется», «Завершено»). Жизненный цикл задачи отображается на канбан-доске, физической или электронной. Такая визуализация делает рабочий процесс открытым и понятным для всех участников, что особенно важно в Agile, когда у команды нет одного формального руководителя.

***Настройка тестовой среды в STLC*** - Среда тестирования определяет условия программного и аппаратного обеспечения, при которых тестируется рабочий продукт. Настройка тестовой среды является одним из важнейших аспектов процесса тестирования и может выполняться параллельно с этапом разработки тестового набора. Команда тестирования может быть не вовлечена в это действие, если клиент / команда разработчиков предоставляет среду тестирования, и в этом случае команда тестирования должна выполнить проверку готовности (тестирование дыма) данной среды. Какие задачи выполняются на этапе настройки среды тестирования STLC?

1. Ознакомление с необходимой архитектурой, настройка среды и подготовка требований к оборудованию и ПО для среды тестирования.
2. Настройка тестовой среды и тестовых данных
3. Выполнение Smoke тестирования

***Activity life cycle***

IOS: Inactive, Active, Background, Suspended

Android: Create, Start, Resume, Pause, Stop, Destroy

***Эмулятор vs симулятор***

* эмулятор пытается дублировать внутреннее устройство устройства, то есть представляет собой полную повторную реализацию конкретного устройства или платформы. Эмулятор действует точно так же, как и реальное устройство.
* симулятор пытается дублировать поведение устройства.  Вы как будто бы работаете в настоящей ОС, но при этом под капотом оно полностью или частично "фальшивое"

## В чем отличия тестирования мобильного приложения от десктопного и веб?

* В первую очередь это разные операционные системы и разная архитектура «железа», хотя сейчас прогресс нацелен на унификацию (например, такие гиганты, как Microsoft и Apple. У MS это планшеты-ноутбуки Surface на базе ARM и Windows 10, Apple в июне 2020 года заявила о переходе на ARM-архитектуру в компьютерах).
* Пока еще актуальное различие – аппаратные ресурсы. Мощность начинки и количество памяти.
* Самое очевидное различие в аппаратной части помимо мощности – наличие разнообразных датчиков и модулей связи в мобильном устройстве, а также нескольких камер, вибромотора, сканера отпечатков и т. д.
* Наличие датчика ориентации уже предполагает тестирование в портретной и ландшафтной ориентациях. Добавьте к этому множество разрешений дисплея, их различные типы матриц со своими особенностями отображения и т.п.
* Помимо этого, в мобильном устройстве очень большое внимание уделяется обработке разнообразных прерываний (входящий звонок, уведомление, нажатие кнопки блокировки, выгрузка из ОЗУ и т. д. )
* Основная функция мобильных устройств – по-прежнему связь. Голосовая, но также и через мобильный интернет, что усложняет тестирование по сравнению с десктопами.
* Прогресс в обновлениях ОС. В мобильных устройствах это происходит гораздо чаще и имеет большее значение в связи с большой конкуренцией.
* Различия в гайдлайнах для ОС.
* Десктопные приложения чаще всего загружаются с официального веб-сайта производителя. Мобильное приложения почти всегда загружается из соответствующего ОС магазина приложений.
* Большее количество вариантов и комбинаций ОС/железа и т.п. в мобильных устройствах (сюда же вытекающее следствие необходимости использования эмуляторов)
* Браузеры «стационарны», в то время как при тестировании мобильных приложений нужно учитывать ориентацию, прерывания, связь, наличие дополнительных модулей.
* С т.з. связи веб приложение фактически становится бесполезным при потере интернет-соединения (хотя в последнее время это иногда не совсем так), для нативного мобильного приложения ничего не изменится\*.
* Публикация и распространение. Для того, чтобы люди начали пользоваться мобильным приложением, необходим аккаунт разработчика и пройденная модерация в магазине приложений.