

# Тестовая документация





# Что это?



Тестовая документация:

- **Внешняя**
- **Внутренняя**



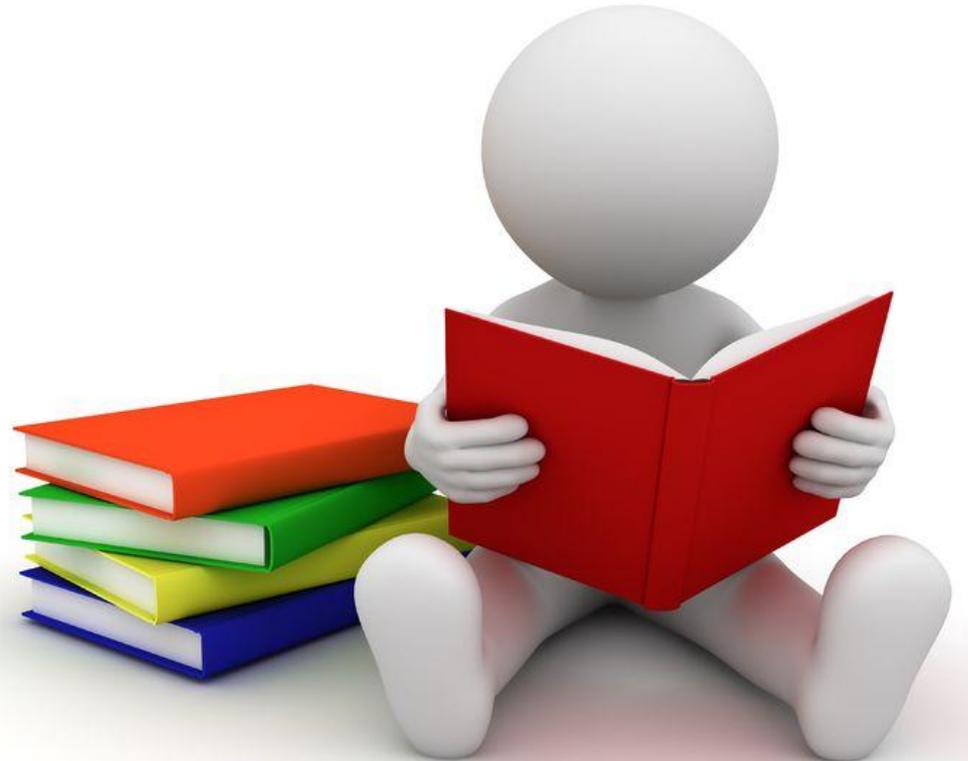


# Внешняя документация



Документация, которая выходит за пределы департамента качества

- **Замечание/Вопрос (Note, Question)**
- **BUG REPORT**
- **CHANGE REQUEST**
- **USER STORY**
- **SPECIFICATION**
- **Test report**



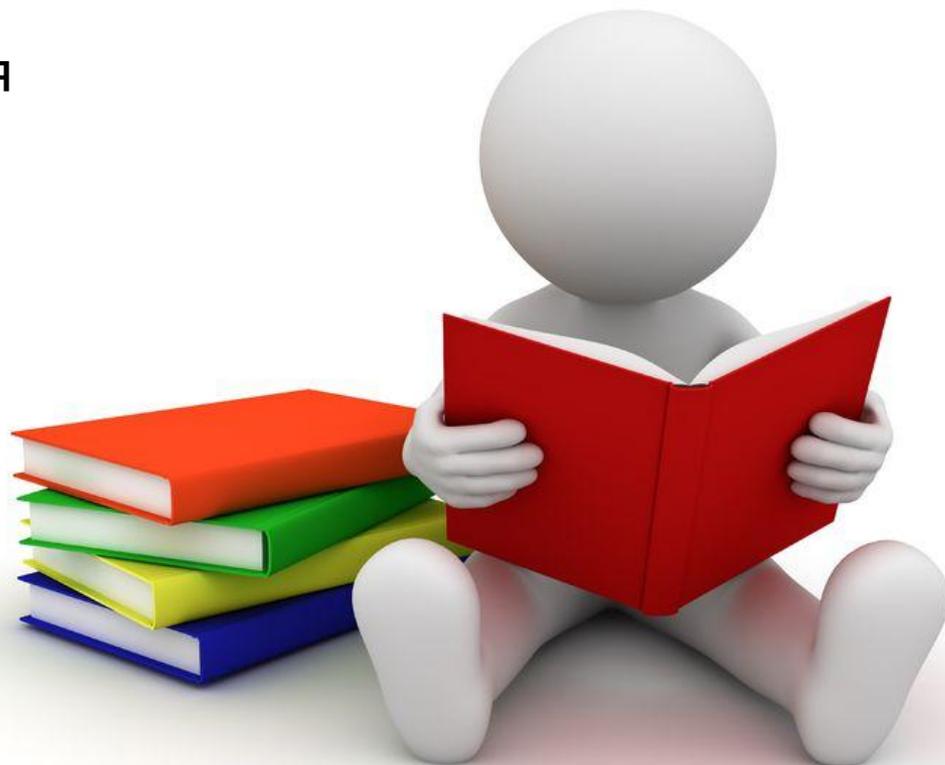


# Внутренняя документация



Документация, которая используется  
внутри департамента качества

- **TEST PLAN**
- **USE CASE**
- **TEST CASE**
- **CHECK LIST**





# TEST PLAN



**Тест план (Test Plan)** — это документ, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования и заканчивая указанием необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения.



Стандарт **IEEE829-1998** описывает шаблон Тест плана и пункты, входящие в стандартную структуру документа

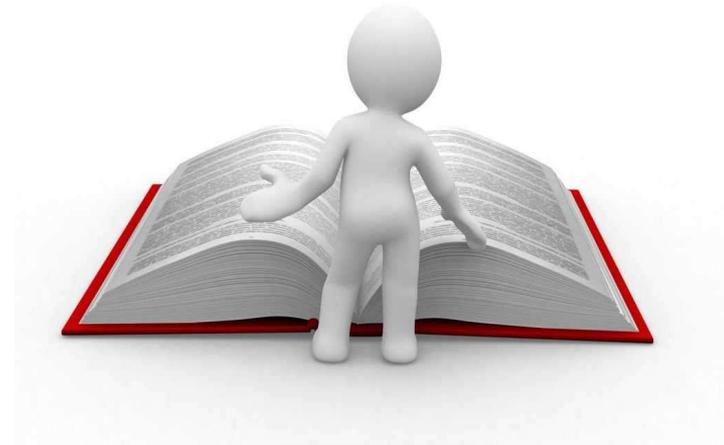


# TEST PLAN



Хороший **тест план** должен как минимум описывать следующее:

- **Что будет тестироваться?**
- **Как будет тестироваться?**
- **Когда будет тестироваться?**
- **Критерии начала тестирования**
- **Критерии окончания тестирования**
- **Необходимое для тестирования оборудование и программные средства**
- **Риски и пути их разрешения**





# TEST PLAN

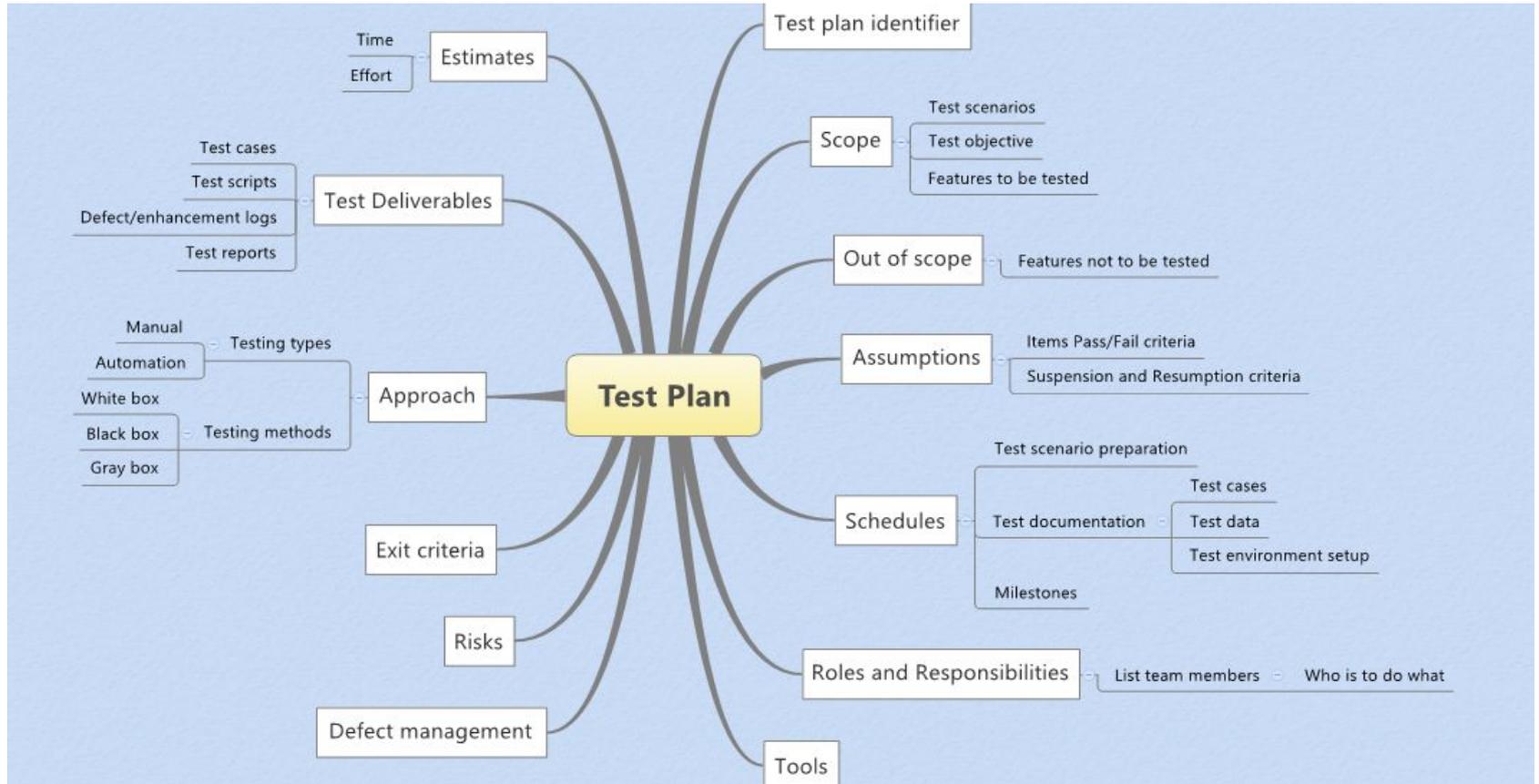


## Способы отображения:

- В виде **традиционного документа** с использованием Microsoft Excel или Microsoft Word, Google docs и т. д.
- Используя методики визуализации с помощью **mind map**, таблиц, диаграмм, коротких схем
- С помощью **профессиональных инструментов** – систем для управления процессами на проектах, в том числе этапом тестирования (EasyQA, TestCaseLab, TestLodge)

# TEST PLAN

Пример использования графической диаграммы для отображения тест-плана



# USE CASE

**Use case** (сценарий использования/пользовательский сценарий) - описание действий, которые может осуществлять система в ответ на внешние воздействия пользователей или других программных систем.

Текст

Диаграмма

**Текстовая форма** может состоять из следующих пунктов:

- Имя сценария
- Цель
- Актеры
- Предварительные условия
- Активаторы
- Порядок Событий
- Альтернативные пути и дополнения

**Диаграмма:**

- Имя
- Актеры
- Варианты использования
- Комментарии





# USE CASE



**Пример 1.** Разблокировать учетную запись пользователя (простой короткий пример, без альтернативного потока событий):

<b>Действующие лица</b>	Администратор, Система
<b>Цель</b>	Изменить статус учетной записи пользователя на «активный».
<b>Предусловие</b>	Учетная запись пользователя неактивна.
<b>Успешный сценарий:</b>	
	1. Администратор выбирает пользователя и активирует «Разблокировать».
	2. Система переключает учетную запись пользователя в статус «активный», и посылает сообщение пользователю на email (если «User Account → email» не пусто).
<b>Результат</b>	Учетная запись пользователя была переведена в статус «активный».

  
**USE  
CASE**  
**Пример 2. Авторизация пользователя (с альтернативными путями):****Действующие лица:** Пользователь, Система**Цели:**  
Пользователь: авторизоваться в системе и начать работать;  
Система: идентифицировать пользователя и его права.**Успешный сценарий:**

1. Пользователь запускает систему. Система открывает сессию пользователя, предлагает ввести логин и пароль.
2. Пользователь вводит логин и пароль.
3. Система проверяет логин и пароль.
4. Система создает запись в истории авторизаций (IP адрес пользователя, логин, дата, рабочая станция).
5. Система выдает пользователю сообщение по поводу успешной авторизации (здесь может быть ссылка на сообщение в Приложении).

**Результат:** Пользователь успешно авторизирован и может работать с системой.**Расширения:**

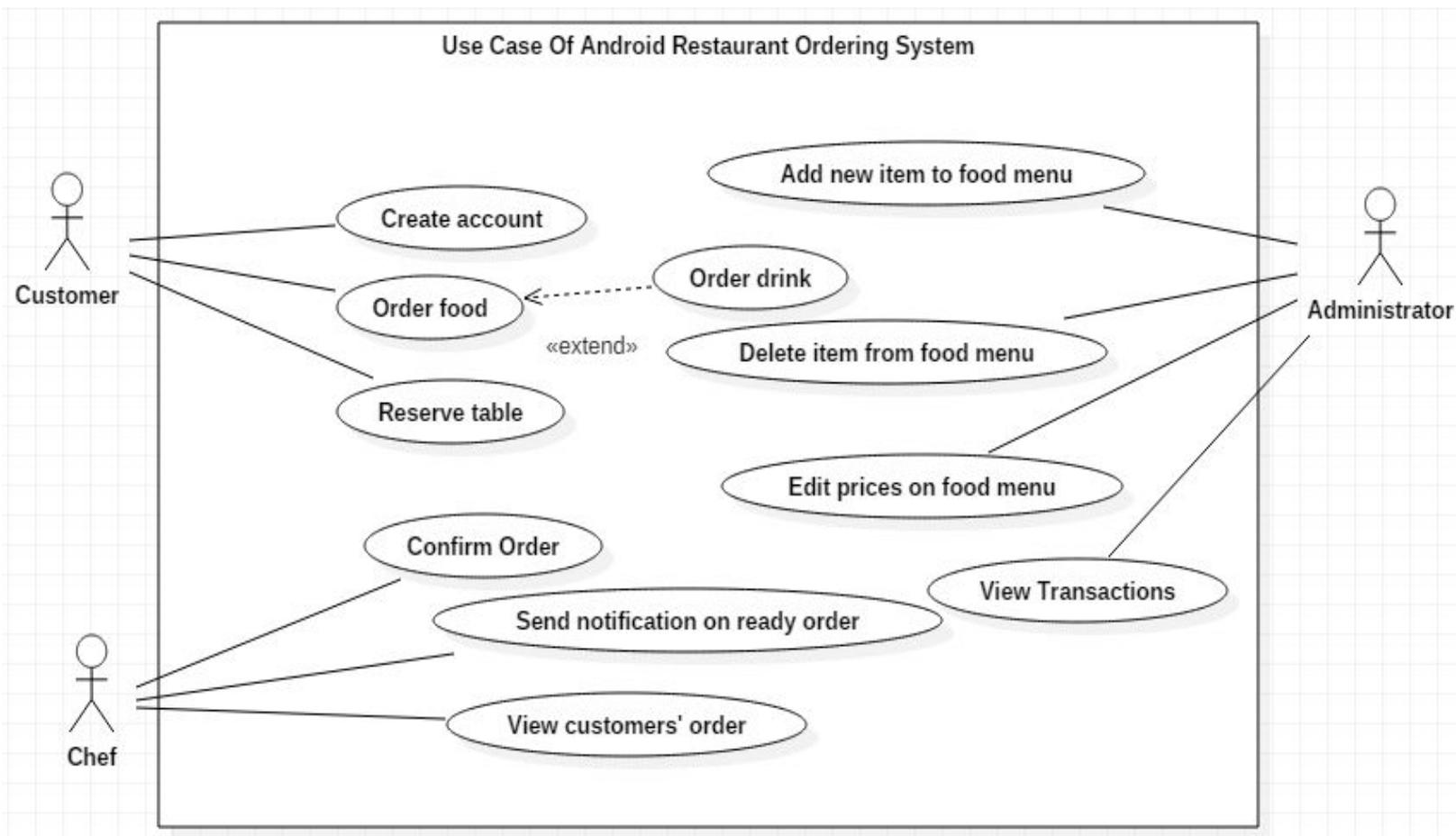
Нет доступа к БД.  
Система выдает сообщение (здесь может быть ссылка на сообщение в Приложении).

**\*a** Результат: пользователь не может войти.**2a** Пользователь выбирает: «Напомнить пароль».  
Вызывается сценарий «Напомнить пароль».

Пользователь с введенными логином и паролем не найден.  
Результат: отказ в авторизации.  
Система выдает сообщение (здесь может быть ссылка на сообщение в Приложении).

**3a** Переход на шаг 2.

USE  
CASE



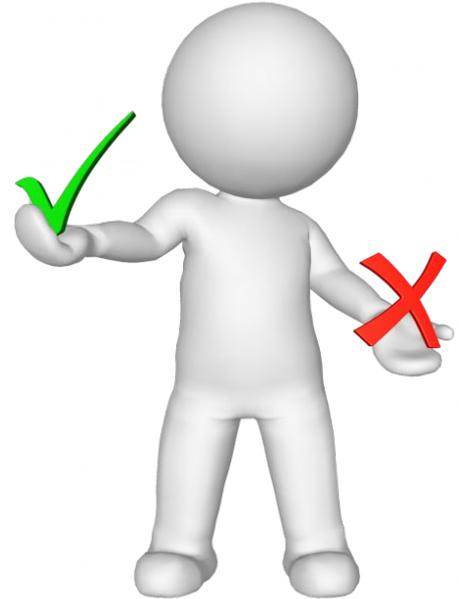
# TEST CASE

**Test Case** (Тестовый случай) — это последовательность действий направленная на проверку какого-либо функционала, описывающая как придти к фактическому результату

**Позитивный тест кейс** использует только корректные данные и проверяет, что приложение правильно выполнило вызываемую функцию.

**Негативный тест кейс** применяется для тестирования, в рамках которого применяются сценарии, что соответствует внештатному поведению тестируемой системы.

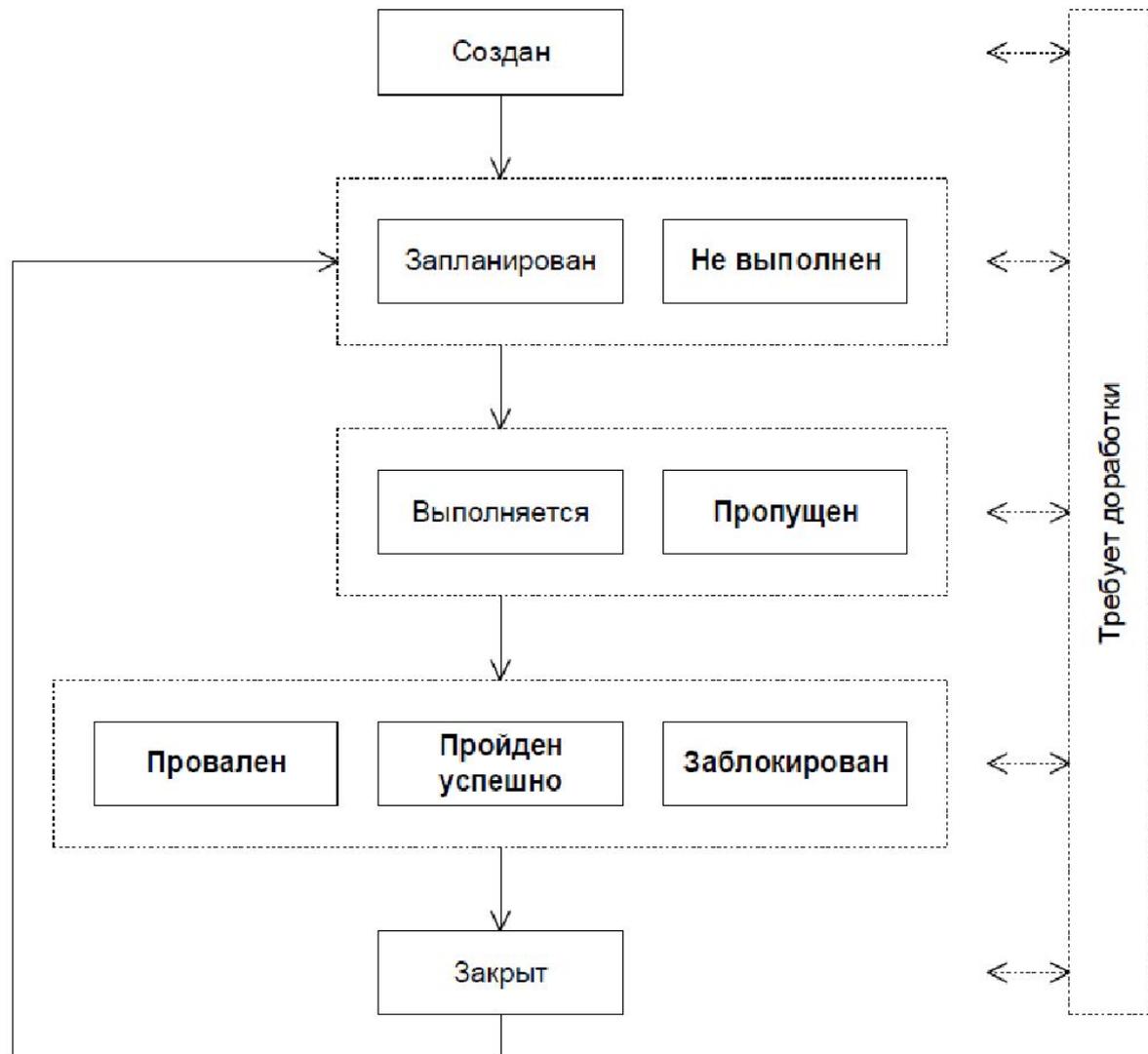
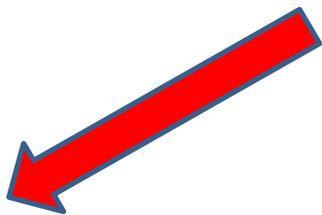
Набор тест-кейсов называется тест-комплексом (**test suite**)



# ЖИЗНЕННЫЙ ЦИКЛ TEST CASE

Test case result:

1. Положительный исход (PASS), если ФР равен ОР, либо
2. Отрицательный исход (FAIL), если ФР не равен ОР: найден баг!



# TEST CASE

## Стандартные атрибуты тест-кейса

- **Test case number (ID)\***
- Приоритет Тест-кейса (Test Case Priority)
- Связанное с тест-кейсом требование (requirement)
- Модуль и подмодуль приложения
- **Test case name (Summary/Title) \***
- **Preconditions\***
- **Steps\***
- **Expected Result\***
- История редактирования (Revision History)



# TEST CASE

The diagram shows a table with 7 columns. Callouts point to specific cells or groups of cells:

- Идентификатор** points to the first cell: UG\_U1.12
- Приоритет** points to the second cell: A
- Связанное с тест-кейсом требование** points to the third cell: R97
- Заглавие (суть) тест-кейса** points to the fourth cell: Загрузка файла
- Ожидаемый результат по каждому шагу тест-кейса** points to the seventh cell, which contains a list of 5 steps.
- Модуль и подмодуль приложения** points to the first three cells (UG\_U1.12, A, R97).
- Исходные данные, необходимые для выполнения тест-кейса** points to the fourth and fifth cells (Загрузка файла, Галерея).
- Шаги тест-кейса** points to the sixth cell, which contains a list of 5 steps.

UG_U1.12	A	R97	Галерея	Загрузка файла	<b>Галерея, загрузка файла, имя со спец-символами</b> Приготовления: создать непустой файл с именем #\$\$%^&.jpg. 1. Нажать кнопку «Загрузить картинку». 2. Нажать кнопку «Выборать». 3. Выбрать из списка подготовленный файл. 4. Нажать кнопку «ОК». 5. Нажать кнопку «Добавить в галерею».	1. Появляется окно загрузки картинки. 2. Появляется диалоговое окно браузера выбора файла для загрузки. 3. Имя выбранного файла появляется в поле «Файл». 4. Диалоговое окно файла закрывается, в поле «Файл» появляется полное имя файла. 5. Выбранный файл появляется в списке файлов галереи.
----------	---	-----	---------	----------------	---	--



# TEST CASE



## Чего не должно быть в тест-кейсе

- Зависимостей от других тест-кейсов
- Нечеткой формулировки шагов или ожидаемого результата
- Отсутствия необходимой для прохождения тест-кейса информации
- Излишней детализации



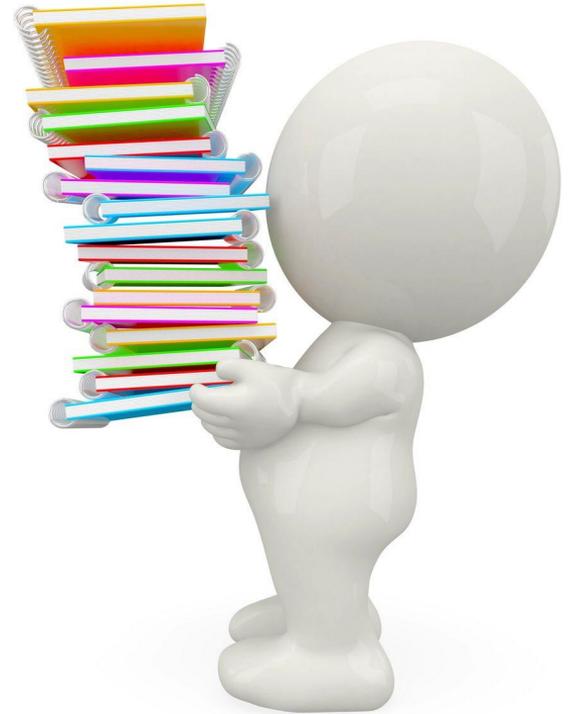


# TEST CASE



## Способы хранения

- В виде **традиционного документа** с использованием Microsoft Excel или Microsoft Word
- **Online документы** - Google docs, Google spreadsheets
- **Профессиональные инструменты** (TestRail, VersionOne)

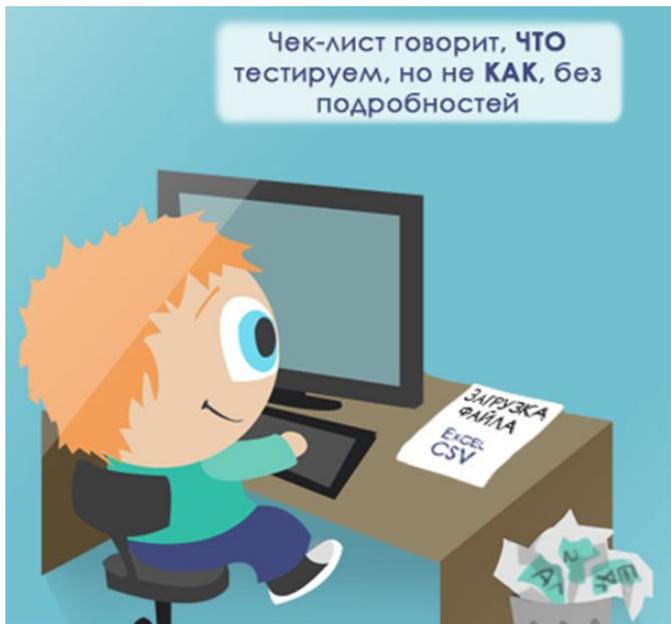




# CHECK LIST



**Check list** (чек-лист) — это документ, описывающий что должно быть протестировано в тезисной форме без обращения к деталям и четко сформулированным шагам



## Основные элементы чек-листа:

- список необходимых проверок
- сборка проекта на которой производилась проверка (Build)
- тестовое окружение (Win/MC OS/browser/etc.)
- ресурс (QA Engineer)
- статус (результат) проверки



# CHECK LIST



**Skipped** –  
проверяться не  
будет по какой-  
либо причине

**Failed** – найден  
один или более  
багов

**Not run** – еще  
не проверено

## Status

**Passed**– проверка  
пройдена успешно,  
багов не найдено

**Blocked** –  
невозможно  
проверить, т.к. один  
из багов блокирует  
текущую проверку

**In Progress** –  
текущий пункт, над  
которым работает  
тестировщик

Passed
Failed
Blocked
Not run
Skipped



## CHECK LIST

### Способы отображения:



- В виде **традиционного документа** с использованием Microsoft Excel или Microsoft Word, Google docs
- Используя методики визуализации с помощью **mind map**, таблиц, диаграмм
- С помощью **профессиональных инструментов** (например, Sitechco)



# CHECK LIST

	Иванов А. А.	Петров Б. Б.	Фёдоров В.В.	Васечкин Г. Г.
Сайт "example.edu"	Google Chrome 64.0.3282.140	Mozilla Firefox 58.0.2	IE 11.0	Opera 51.0.2830.34
Регистрация и Личный профиль				
Регистрация на сайте	Passed	Passed	Passed	Passed
Редактирование профиля	Failed +	http://bt-w.qatestlab.com/view.php?id=...		Failed
Форма обратной связи				
Валидация полей	Failed	Failed	Passed	Passed
Отправка письма/уведомления	Blocked	Blocked	Passed	Passed
Доставка письма/уведомления	Blocked	Blocked	Skipped	Skipped

## Полезные советы при оформлении чек-листа:

- **Сокращайте.** Это не сочинение. Это просто набор пунктов. Чёткость формулировок!
- **Удаляйте ненужное.** Включайте только то, что вы можете забыть.
- **Выделяйте важное.** Капслоком или жирным шрифтом. Главное — не переборщите.
- **Постоянно редактируйте.** Чек-лист должен быть актуальным.

# TEST CASE vs CHECK LIST



Параметр	Чеклисты	Тест-кейсы
Время составления	Не требует много времени, так как пункты не содержат всех деталей проверки.	Длительный процесс.
Уровень знаний	Для опытного сотрудника тестирование по чек листу занимает меньше времени за счет глубокого понимания системы	Тестировать с помощью тест кейсов может даже сотрудник с небольшим знанием продукта.
Тестирование нового функционала	Чек лист не очень хорошо подходит при подготовке к тестированию новой фичи, поскольку приемочное тестирование требует проверки всех деталей	Новый функционал лучше тестировать по детальным тест кейсам.
База знаний	Чек листы не могут быть носителем требований, так как не содержат достаточных деталей.	Часто внутренняя тестовая документация - это единственное место, где собраны все требования.
Поддерживаемость	Чек листы проще держать в up-to-date состоянии. Мало текста и мало детального описания.	Тест кейсы требуют более частого апдейта.
Гибкость тестирования	Чек листы с меньшей вероятностью вызывают эффект пестицида.	Тест-кейсы ограничивают воображение тестировщика, увеличивая эффект пестицида.

# TEST CASE vs CHECK LIST



Выбор в пользу одного из этих типов документов зависит от следующих факторов:

1. **Длительность проекта**
2. **Сроки на выполнение тестирования**
3. **Запрос менеджмента**
4. **Вид тестирования**
5. **Выбор команды**
6. **Наличие и формат требований**



## Источники информации для тестировщика



- **Внешняя документация** (User story, спецификация, change request)
- **Личное общение** с участниками проекта (менеджерами, программистами, бизнес аналитиками и т.д.)
- **Исследование** (результат собственного опыта, полученного в ходе экспериментов над программой)
- **Аналогичные проекты**
- **Черновики** руководства пользователя, заметки разработчиков, макеты, e-mails и т.д.)





## Какой тип документации использовать



**Зависит от следующих факторов:**

- специфики проекта
- сроков
- наличия информации
- состава команды
- требований

