



# Разработка тестов и тестовых сценариев

**FT/AT Program**

March, 2020

# Agenda

---

- 1 Определения
- 2 Разработка тестов
- 3 Техники тест-дизайна
- 4 Рекомендации по разработке тестов
- 5 Оформление тест-кейсов
- 6 Свойства хорошего теста
- 7 Процесс разработки тестов
- 8 Тестовые сценарии
- 9 Пример разработки тестов



Разработка тестов и тестовых сценариев

# ОПРЕДЕЛЕНИЯ

**Чек-лист** (check-list) - набор идей тестов.

**Тест-кейс** (test case) - набор входных данных, условий выполнения и ожидаемых результатов, разработанный с целью проверки того или иного свойства или поведения программного средства.

**Тестовый сценарий, тест-сьют** (test scenario, test-suite) - набор тест-кейсов, собранных в группу (последовательность) для достижения некоторой цели.



Разработка тестов и тестовых сценариев

# РАЗРАБОТКА ТЕСТОВ

# Разработка тестов

Перед вами обыкновенная ручка. Давайте подумаем, как её можно протестировать?

- Тесты на основе требований
- Функциональные тесты
- Сравнительные тесты
- Сценарные тесты
- Тесты ошибочных ситуаций
- Тесты интерфейса
- Тесты удобства использования

## (requirements based tests)

- Извлекается и вставляется ли в ручку стержень?
- Присутствует ли держатель, позволяющий цеплять ручку за край кармана?
- Переключается ли ручка из рабочего в нерабочее положение?

**(functional test)**

- Вставить в ручку стержень.
- Переключить в рабочее положение.
- Написать несколько слов.
- Переключить в нерабочее положение.
- Извлечь стержень.



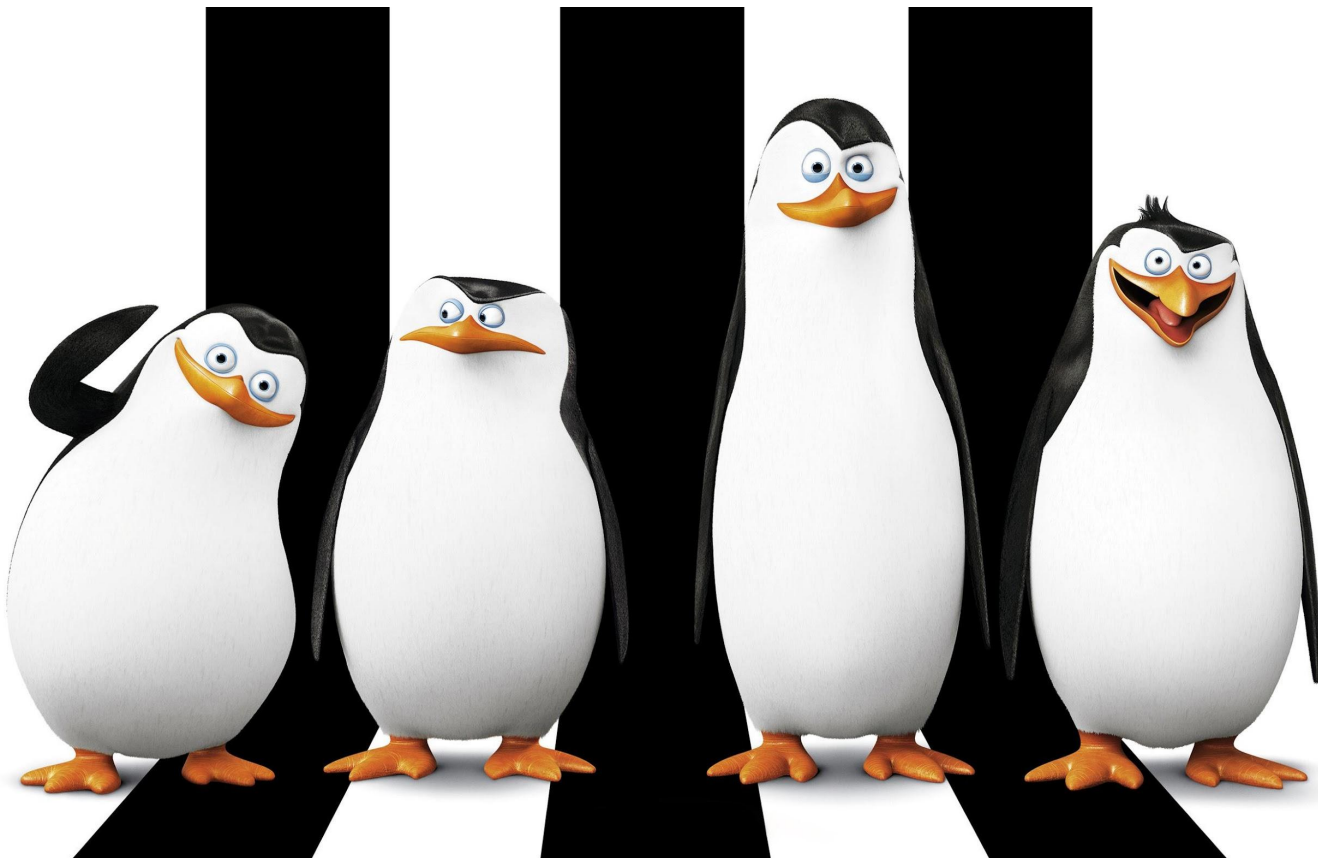


**(parallel test)**

- Что мы можем сказать об этой ручке в сравнении с другими ручками, которые выпускает наша фирма?
- Что мы можем сказать об этой ручке в сравнении с ручками, которые выпускают конкуренты?
- В чём преимущества именно этой модели ручек?

- Что произойдёт, если препятствовать выходу стержня в рабочее положение?
  - Какое усилие и где надо приложить к ручке, чтобы её сломать?
    - Если стержень застрял, легко ли его извлечь?
  - Что произойдёт, если писать по стеклу, асфальту?

**Scenario-based tests**



- Измерения: высота, ширина, длина, вес.
- Цвет.
- Читаемость логотипа фирмы-производителя.

**GUI tests**

### usability tests

- Есть ли у нас какие-либо замечания по юзабилити ручек от пользователей?
- Есть ли у нас представители целевых групп, чтобы привлечь их к тестированию?
- Как много времени у пользователя занимает переключение ручки из нерабочего положения в рабочее и обратно?
- Как быстро пользователь понимает, как пользоваться ручкой?
- Как быстро пользователь привыкает к этой ручке?
- Легко ли понять, какие стержни подходят к ручке?
- Легко ли заменить стержень?
- Может ли ручкой пользоваться левша?

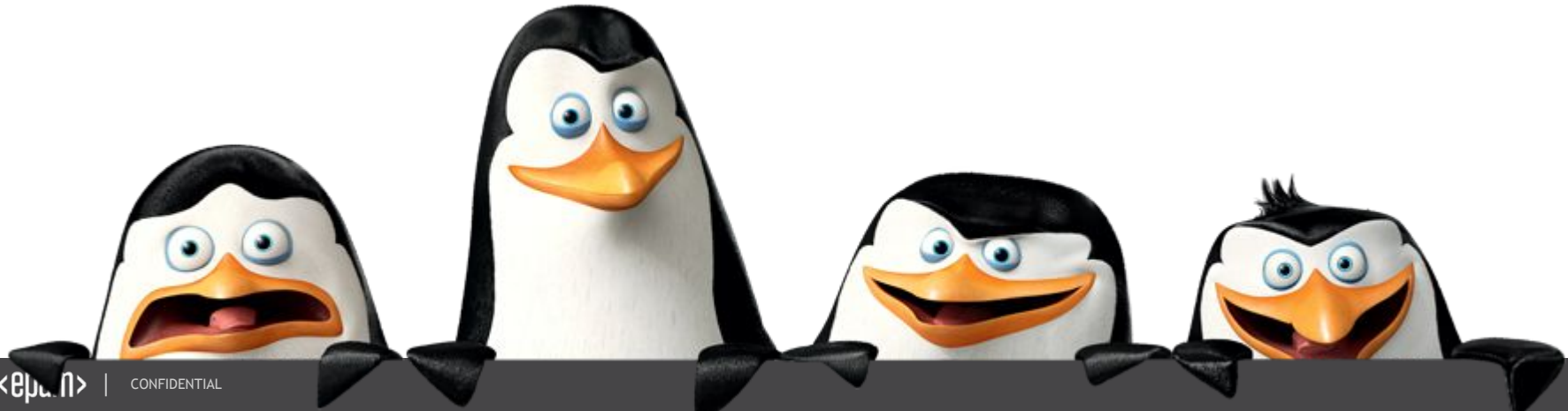
### packaging/documentation tests

- Вложена ли в упаковку копия текста о гарантийных обязательствах?
- Ясно ли видно на упаковке, что внутри?
- Легко ли открыть упаковку?
- Насколько материалы упаковки вредны для окружающей среды?
- Есть ли какие-то особые требования к упаковке?
- На сайте, в каталоге, на упаковке написано и нарисовано одно и то же?
- Текст на упаковке и в гарантийном обязательстве - на одном и том же языке?
- На упаковке и в документации нет грамматических ошибок, опечаток и т.д.?

# Разработка тестов. Стрессовые тесты

- При какой температуре расплавится пластиковая часть ручки?
- При какой температуре потечёт стержень?
- При какой температуре ручка перестает писать?
- Какое воздействие необходимо применить к ручке, чтобы сломать её?
- Пишет ли ручка под водой? А по мокрой бумаге?
- Если ручку уронить в песок - что произойдёт?
- А если уронить со стола?
- А если из окна офиса?

**stress tests**



### performance tests

- Сколько текста можно написать ручкой в единицу времени?
- Как быстро ручку можно привести в рабочее положение?
- Как много раз ручку можно переключить из нерабочего в рабочее положение, прежде чем её начнёт заедать?

## configuration tests

- Какие стержни подходят к нашей ручке?
- На каких поверхностях она может писать?





- Подлежит ли этот продукт какому-то виду лицензирования?
- Необходима ли какая-то особая сопроводительная документация?
- Ясно ли из документации ручки видно, в какой стране она произведена?
- Существуют ли какие-то законодательные особенности, препятствующие распространению нашего продукта?

**regulation tests**



Без **записанного чек-листа** уже через несколько минут идеи начинают дублироваться, теряться, искажаться и т.п.

A photograph of a busy city street at sunset. The scene is filled with pedestrians, buildings, and streetlights. A large blue text overlay is positioned in the center of the image. The background shows a street with a large building on the left, a Starbucks on the right, and a sign for 'Superdry' in the distance. The sun is low in the sky, creating a warm, golden glow. The text 'Разработка тестов и тестовых сценариев' is written in white on a blue rectangular background. Below it, the main title 'ТЕХНИКИ ТЕСТ ДИЗАЙНА' is written in large, white, bold letters on a larger blue rectangular background.

Разработка тестов и тестовых сценариев

# ТЕХНИКИ ТЕСТ ДИЗАЙНА

# Наиболее распространенные техники тест дизайна

- **Эквивалентное Разделение (Equivalence Partitioning - EP)**. Как пример, у вас есть диапазон допустимых значений от 1 до 10, вы должны выбрать одно верное значение внутри интервала, скажем, 5, и одно неверное значение вне интервала - 0.
- **Анализ Граничных Значений (Boundary Value Analysis - BVA)**. Если взять пример выше, в качестве значений для позитивного тестирования выберем минимальную и максимальную границы (1 и 10), и значения больше и меньше границ (0 и 11). Анализ Граничный значений может быть применен к полям, записям, файлам, или к любого рода сущностям имеющим ограничения.
- **Причина / Следствие (Cause/Effect - CE)**. Это, как правило, ввод комбинаций условий (причин), для получения ответа от системы (Следствие). Например, вы проверяете возможность добавлять клиента, используя определенную экранную форму. Для этого вам необходимо будет ввести несколько полей, таких как "Имя", "Адрес", "Номер Телефона" а затем, нажать кнопку "Добавить" - эта "Причина". После нажатия кнопки "Добавить", система добавляет клиента в базу данных и показывает его номер на экране - это "Следствие".

# Наиболее распространенные техники тест дизайна

- **Предугадывание ошибки (Error Guessing - EG)**. Это когда тестировщик использует свои знания системы и способность к интерпретации спецификации на предмет того, чтобы "предугадать" при каких входных условиях система может выдать ошибку. Например, спецификация говорит: "пользователь должен ввести код". Тестировщик, будет думать: "Что, если я не введу код?", "Что, если я введу неправильный код? ", и так далее. Это и есть предугадывание ошибки.
- **Исчерпывающее тестирование (Exhaustive Testing - ET)** - это крайний случай. В пределах этой техники вы должны проверить все возможные комбинации входных значений, и в принципе, это должно найти все проблемы. На практике применение этого метода не представляется возможным, из-за огромного количества входных значений.

# Эквивалентное Разделение. Классы эквивалентности

**Класс эквивалентности** (equivalence class) - набор тестов, полное выполнение которого является избыточным и не приводит к обнаружению новых дефектов.



# Признаки эквивалентности

Они направлены на поиск одной и той же ошибки

Если один из тестов обнаруживает ошибку, другие ее тоже, скорее всего обнаружат, и наоборот

Тесты используют схожие наборы входных данных

Для выполнения тестов мы совершаем одни и те же операции

Тесты генерируют одинаковые выходные данные или приводят приложение в одно и то же состояние

Все тесты приводят к срабатыванию одного и того же блока обработки ошибок

Ни один из тестов не приводит к срабатыванию блока обработки ошибок

## Классы эквивалентности. Граничные условия

---

**Граничные условия** (border conditions) - это те места, в которых один класс эквивалентности переходит в другой.

*Граничные условия очень важны, и их обязательно следует проверять в тестах, т.к. именно в этом месте чаще всего и обнаруживаются ошибки.*



# Классы эквивалентности. Пример 1

Проверить реакцию приложения на ввод слишком короткого (менее трёх символов) или слишком длинного (более 20-ти символов) имени пользователя, которое может содержать только английские буквы, цифры и знак подчёркивания.

## Классы эквивалентности:

1. **Позитивный тест:** строка допустимых символов длиной от трёх до 20-ти символов.
2. **Негативный тест:** строка короче трёх символов.
3. **Негативный тест:** строка длиннее 20-ти символов.
4. **Негативный тест:** строка длиной от трёх до 20-ти символов, содержащая недопустимые символы.

## Тесты:

1. ABC, ABCDEFGHIJKLMNOPQRST, abc\_12\_def
2. AA, {пустая строка}
3. AAAAAAAAAAAAAAAAAAAAAAAAAA (21 символ)
4. Abcd#23456%#@#&#%^#

## Классы эквивалентности. Пример 2

*«Чтобы добавить файл в свою фотогалерею на сайте, пользователь должен кликнуть по кнопке Открыть, выбрать файл и кликнуть по кнопке ОК».*

Давайте абстрагируемся от пользовательского интерфейса и подумаем о файле.  
**Какие случаи нам надо будет проверить?**

## Классы эквивалентности. Пример 3

**Дата:**

**03/12/06**

m/dd/yy

Есть вопросы? ☺

Какие здесь классы эквивалентности?

**Допустимая дата, недопустимая дата.**

Какая дата является допустимой?

<b>Год</b>	<b>Месяц</b>	<b>День</b>
<b>0-99</b>	<b>1-12</b>	<b>1-28</b>
		<b>1-29</b>
		<b>1-30</b>
		<b>1-31</b>
<b>Разделитель /</b>		

# Классы эквивалентности. Пример 3

Год	Месяц	День
0-99	1-12	1-28 1-29 1-30 1-31

Тест	Допустимо	Недопустимо
Минимальный день, месяц, год	1/1/0	0/1/0 1/0/0 1/1/-1
Максимальный день, месяц, год	12/31/99	13/31/99 12/32/99 12/31/100
Максимальный день	2/29/20 2/28/01 04/30/65	2/30/20 2/29/01 04/31/65
Буквы, спецсимволы, некорректный разделитель	-	March 3.12.06 !@#\$%^&*()_+ {}

Пустое поле (Допустимо или нет – зависит от требований)

- **Классы эквивалентности не всегда очевидны.**
- Как правило, негативных тестов получается больше, чем позитивных.
- Принадлежность теста к позитивным или негативным зависит от требований.



Разработка тестов и тестовых сценариев

# РЕКОМЕНДАЦИИ ПО РАЗРАБОТКЕ ТЕСТОВ

# Рекомендации по разработке тестов

- Начинайте с **простых очевидных тестов**.
- Затем **переходите к более сложным тестам**.
- Помните о граничных условиях.
- Если остаётся время, занимайтесь **исследовательским тестированием**.

## Последовательность разработки и выполнения тестов

- Простые позитивные.
- Простые негативные.
- Сложные позитивные.
- Сложные негативные.

A photograph of a busy city street at sunset. The sun is low in the sky, creating a warm, golden glow and long shadows. Pedestrians are walking across the street. Buildings line the street, including one with a Union Jack flag and another with a Starbucks logo. A sign for 'Superdry' is visible on a building. The overall scene is a vibrant, urban environment.

Разработка тестов и тестовых сценариев

# ОФОРМЛЕНИЕ ТЕСТ-КЕЙСОВ



# Оформление. Тест-кейсы в Excel

<p>Приоритет</p> <p>Smoke Critical Path Extended</p> <p>Идентификатор</p>	<p>UG_</p> <p>Ст.1</p> <p>2</p>	<p>A R97</p>	<p>Галерея</p>	<p>Загрузка файла</p>	<p><b>Галерея, загрузка файла, имя со спецсимволами</b></p> <p>Приготовления: создать непустой файл с именем #\$\$%^&amp;... 1. Нажать кнопку «Загрузить картинку» 2. Нажать кнопку «Выбрать» 3. Выбрать из списка подготовленный файл 4. Нажать кнопку «ОК» 5. Нажать кнопку «Добавить галерею»</p>	<p>1. Появляется окно загрузки картинки 2. Появляется диалоговое окно браузера выбора загрузки изображения 3. Диалоговое окно файла закрывается, в поле «Файл» появляется полное имя файла 4. Диалоговое окно файла закрывается, в поле «Файл» появляется полное имя файла 5. Выбранный файл появляется в списке файлов галереи</p>
---	---------------------------------	--------------	----------------	-----------------------	--	---

Заглавие (суть) теста

Приоритет

Smoke  
Critical Path  
Extended

Идентификатор

Связанное с тестом требование

Модуль и подмодуль

Шаги

Исходные данные, необходимые для выполнения теста

Ожидаемый результат по каждому шагу

# Оформление тест-кейсов

**Краткое описание** (summary) - предельно сжатое изложение сути теста.

**Пререквизиты** (preconditions) - ситуация и/или условие, предшествующее тесту.

**Данные** (test data) - информация, необходимая для выполнения теста.

**Ссылка на требования** (bind to requirement) - ссылка на требования, которые покрывает данный тест.

**Описание** (Test description/Test Steps) - описание действия, выполняемого на данном шаге.

**Ожидаемый результат** (expected result) - результат выполнения действия, который будет получен в случае корректной работы приложения.

A photograph of a busy city street at sunset. The sun is low in the sky, creating a warm, golden glow and long shadows. Pedestrians are walking across the street. Buildings line the street, including one with a Union Jack flag and another with a Starbucks logo. A sign for 'Superdry' is visible on a building. The overall atmosphere is vibrant and urban.

Разработка тестов и тестовых сценариев

# СВОЙСТВА ХОРОШЕГО ТЕСТА

# Специфичность или общность?

Оба тест-кейса делают одну и ту же проверку.

**Какой из них лучше?**

1. В поле А ввести 10	4. Значение в поле С равно 25
2. В поле В ввести 15	
3. Нажать кнопку «Сложить»	
4. Проверить значение в поле С	

1. Проверить, что программа суммирует два числа корректно	4. Суммирует корректно
---	------------------------

Оказывается, оба варианта - плохие. **Почему?**

## Специфичность или общность?

Когда все детали прописаны до мелочей:

- при повторных выполнениях теста всегда будут выполняться строго одни и те же действия, что снижает вероятность обнаружить ошибку;
- возрастает время создания и поддержки теста.

Общий тест-кейс сложно выполнять по многим объективным и субъективным причинам, а потому:

- он сложен для начинающих тестировщиков;
- он вполне может остаться невыполненным.

# Специфичность или общность?

## Сложение А и В

1. В поле А ввести корректное целое число
2. В поле В ввести корректное целое число
3. Нажать кнопку «Сложить»
4. Проверить значение поля С
5. Повторить шаги 1-4 для значений: 0, максимального и минимального допустимого значений.

4. Значение поля С равно сумме А и В

- Здесь мы **не привязаны к конкретным значениям**.
- Мы знаем, как **проверить результат**.
- Мы **сокращаем время** написания и поддержки теста ссылкой на шаги 1-4.
- Мы **перечислили значения**, представляющие для нас особый интерес.

# Простота или сложность?

## Простые тесты:

1. Откройте файл «1.txt». Файл открыт.
2. Введите слово «Дом». Появляется слово «Дом».
3. Сохраните файл. Кнопка «Сохранить» становится неактивной.

## Сложный тест:

1. В документе размером более 100 Мб создайте таблицу 100x100, в ячейку 50x50 вставьте картинку размером 30 Мб, применив к ней функцию «Авторасположение». Проверьте результат.

**Простые тесты оперируют за раз одним объектом.**

## Преимущества простых тестов:

- Их легко выполнять.
  - Они понятны новичкам.
- Они упрощают диагностику ошибки.  
Они делают наличие ошибки очевидным.

## Преимущества сложных тестов:

- Больше шансов что-то сломать.
- Пользователи, как правило, используют сложные сценарии.
- Программисты сами редко проверяют такие варианты.

# Независимость или связанность?

---

**Независимые тесты** не ссылаются ни на какие другие.

**Связанные тесты** явно или неявно (в рамках сценария) ссылаются на другие (как правило, на предыдущий).



# Независимость или связанность? Преимущества

Независимые тесты	Связанные тесты
Легко и просто выполнять	Они имитируют работу реальных пользователей
Такие тесты могут работать даже после краха приложения на других тестах	Они удобны для интеграционного тестирования
	Они удобны для разбиения на части тестов с большим количеством шагов
Такие тесты можно группировать любым образом и выполнять в любом порядке	Следующий в наборе тест использует данные и состояние приложения. Подготовленные предыдущим

- Используйте **активный залог**: («open», «paste», «click»). В русском языке используйте **безличную форму**: «открыть» (вместо «откройте»).
- Описывайте **поведение системы**: «появляется окно...», «приложение закрывается».
- Используйте **простой технический стиль**.
- **ОБЯЗАТЕЛЬНО** указывайте **ТОЧНЫЕ названия** всех элементов приложения.
- Не объясняйте **базовые понятия** работы с ОС.

A photograph of a busy city street at sunset. The sun is low in the sky, creating a warm, golden glow and long shadows. The street is filled with pedestrians. On the left, there is a large, ornate building with a balcony and a flag. A street sign reads 'AGENT STREET W1'. In the background, a red flag with the word 'Superdry' is visible. On the right, there is a Starbucks logo and a red awning. The overall scene is a vibrant, urban setting.

Разработка тестов и тестовых сценариев

# И ЕЩЕ ПАРА СЛОВ О ХОРОШИХ ТЕСТАХ

# Каким должен быть хороший тест?

Обладает высокой вероятностью обнаружения ошибки.

1/1 = 1

-5/0 = err!

Исследует соответствующую («ту, которую надо») область приложения.

Сохранение файла

1. Открыть «Файл -> Печать».
2. ...

ЗАЧЕМ?!

Выполняет какие-то интересные действия.

...  
3. Файл с именем «%^##76 / // \ ^^ [ ] :  
.jpg» загрузить в качестве аватарки.  
...

Как это можно  
сделать?



# Каким должен быть хороший тест?

Не является избыточным по отношению к другим тестам.

1+1	3+4	1+98	17+98
1+2	7+5	20+21	
2+1	8+1	30+12	

Делает обнаруженную ошибку очевидной.

23746568964589658923689  
2365 +  
2375641647647647816478 =  
823645895648965896

???????????????

100 + 50 = 212

!!!

# Каким должен быть хороший тест?

Позволяет легко диагностировать ошибку.

...  
7. Годовой баланс должен сойтись.  
...

...  
3. Сумма 2 и 7 равна 9.  
...

A photograph of a busy city street at sunset. The sun is low in the sky, creating a warm, golden glow and long shadows. Pedestrians are walking across the street. In the background, there are multi-story buildings, some with scaffolding. A sign for 'Superdry' is visible on a building. A Starbucks logo is also visible on a building. A street sign for 'GENT STREET W1' is visible on a building. A Union Jack flag is flying from a building on the left. A red traffic light is visible in the foreground.

Разработка тестов и тестовых сценариев

# ПРОЦЕСС РАЗРАБОТКИ ТЕСТОВ



# Процесс разработки тестов

---

1. Начинайте **как можно раньше**, ещё до выхода первого билда.
2. Разбивайте приложение на **отдельные части/модули**.
3. Для каждой области/модуля **пишите чек-лист**.
4. Пишите **вопросы**, уточняйте детали, добавляйте «**косметику**», используйте **copy-paste**.
5. Получите **рецензию** коллег-тестировщиков, разработчиков, заказчиков.
6. **Обновляйте тесты**, как только обнаружили ошибку или изменилась функциональность.



Разработка тестов и тестовых сценариев

# ТЕСТОВЫЕ СЦЕНАРИИ

## Тестовый сценарий. Определение

**Тестовый сценарий** (test scenario) - набор тестов (тест-кейсов), собранных в последовательность **для достижения некоторой цели**. Может быть составлен из связанных или независимых тестов.

*Хороший тестовый сценарий всегда **следует некоторой логике**, например: типичному использованию приложения, удобству тестирования, распределению функций по модулям и т. д.*

# Рекомендации по оформлению

Используйте фильтры

Используйте группировку

Используйте отдельные листы

	B	C	D	E	F	G	H	I
1	UI_001	(All) (Top 10...) (Custom...) A B C	R1001	Gallery	Upload	<b>Load File, Correct</b> 1. Open load dialog 2. Choose file 3. Click OK 4. Close window		
2	UI_001.2	A	R1001	Gallery	Upload	<b>Load File, Incorrect</b> 1. Open load dialog 2. Choose file 3. Click OK 4. Close window		
3	UI_001.3	A	R1001	Gallery	Upload	<b>Load File, Correct</b> 1. Open load dialog 2. Choose file 3. Click OK 4. Close window		
4	UI_001.4	B	R1001	Gallery	Upload			
5	UI_001.5	A	R1001	Gallery	Upload			
6								
8								
9								
10								
11								
12								

## Тестовый сценарий. Общие рекомендации

---

- Пишите сценарий **для отдельной части** приложения.
- Пишите **отдельно сценарии** для Smoke и Critical Path тестов.
- Постепенно **повышайте сложность** тестов.
- Организуйте сценарий **логично**.



Разработка тестов и тестовых сценариев

# ТЕСТОВОЕ ПОКРЫТИЕ

- Тестовое Покрытие (Test Coverage) - это одна из метрик оценки качества тестирования, представляющая из себя плотность покрытия тестами требований, либо исполняемого кода.

**Traceability matrix — Матрица соответствия требований** — это двумерная таблица, содержащая соответствие функциональных требований (functional requirements) продукта и подготовленных тестовых сценариев (test cases).

В заголовках колонок таблицы расположены требования, а в заголовках строк — тестовые сценарии. На пересечении — отметка, означающая, что требование текущей колонки покрыто тестовым сценарием текущей строки.

A photograph of a busy city street at sunset. The sun is low in the sky, creating a warm, golden glow and long shadows. Pedestrians are walking across the street. Buildings line the street, including one with a Union Jack flag and another with a Starbucks logo. A sign for 'Superdry' is visible on a building. The overall atmosphere is vibrant and urban.

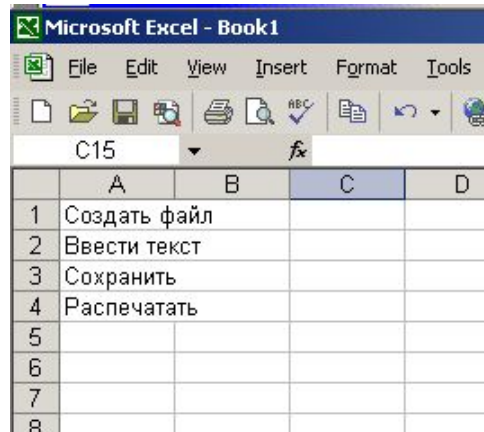
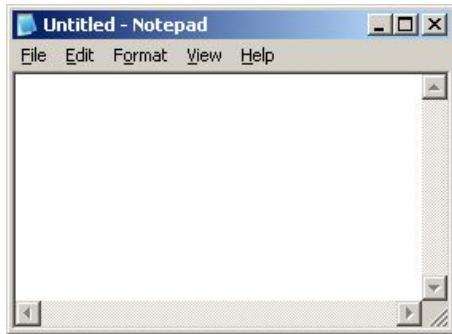
Разработка тестов и тестовых сценариев

# ПРИМЕР РАЗРАБОТКИ ТЕСТОВ



# Пример разработки тестов

1. Что такое Notepad?
2. Какие функции для него наиболее важны?

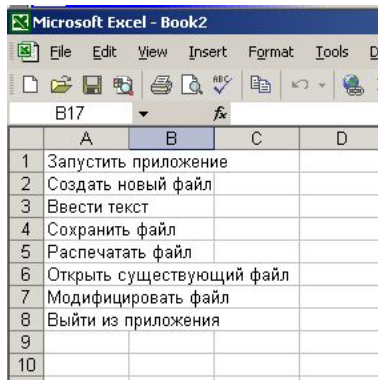


The image shows a screenshot of the Microsoft Excel application window. The title bar reads "Microsoft Excel - Book1". The menu bar includes "File", "Edit", "View", "Insert", "Format", and "Tools". The active cell is C15. The spreadsheet contains the following text:

	A	B	C	D
1	Создать файл			
2	Ввести текст			
3	Сохранить			
4	Распечатать			
5				
6				
7				
8				

# Пример разработки тестов

Итак, вот наш Smoke test:



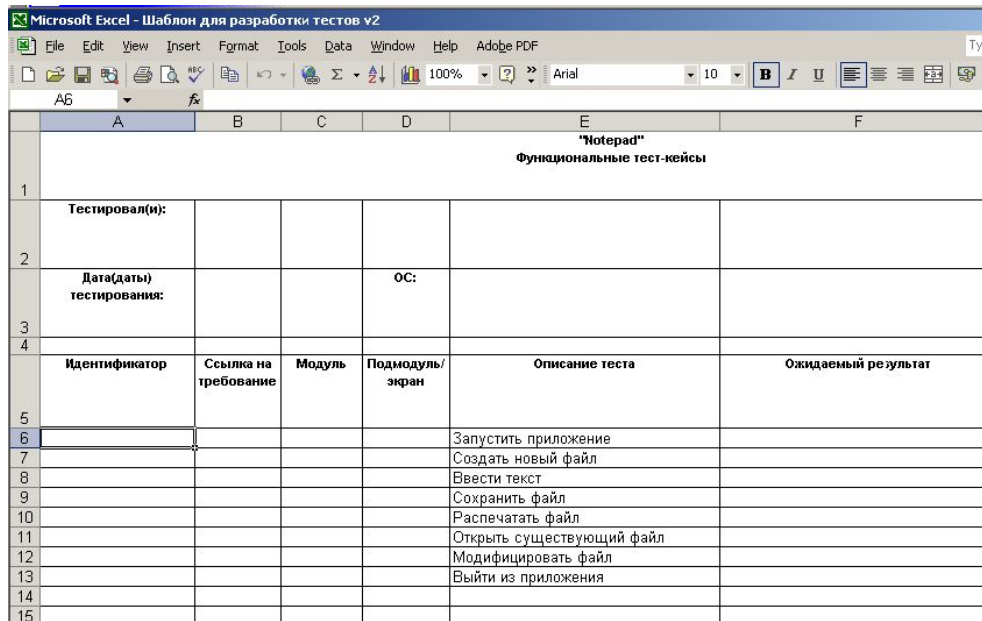
The screenshot shows a Microsoft Excel spreadsheet with a list of steps in column B. The steps are:

	A	B	C	D
1		Запустить приложение		
2		Создать новый файл		
3		Ввести текст		
4		Сохранить файл		
5		Распечатать файл		
6		Открыть существующий файл		
7		Модифицировать файл		
8		Выйти из приложения		
9				
10				

Перенесём его в шаблон для разработки тестов.

# Пример разработки тестов

Фактически, это - чек-лист. И сами пункты грамотно сформированного чек-листа - (почти) готовые заголовки тест-кейсов.



The screenshot shows a Microsoft Excel spreadsheet titled "Microsoft Excel - Шаблон для разработки тестов v2". The spreadsheet is set up for functional testing of "Notepad".

	A	B	C	D	E	F
1	<b>"Notepad"</b> Функциональные тест-кейсы					
2	Тестировал(и):					
3	Дата(даты) тестирования:			ОС:		
4	Идентификатор	Ссылка на требование	Модуль	Подмодуль/экран	Описание теста	Ожидаемый результат
5						
6					Запустить приложение	
7					Создать новый файл	
8					Ввести текст	
9					Сохранить файл	
10					Распечатать файл	
11					Открыть существующий файл	
12					Модифицировать файл	
13					Выйти из приложения	
14						
15						

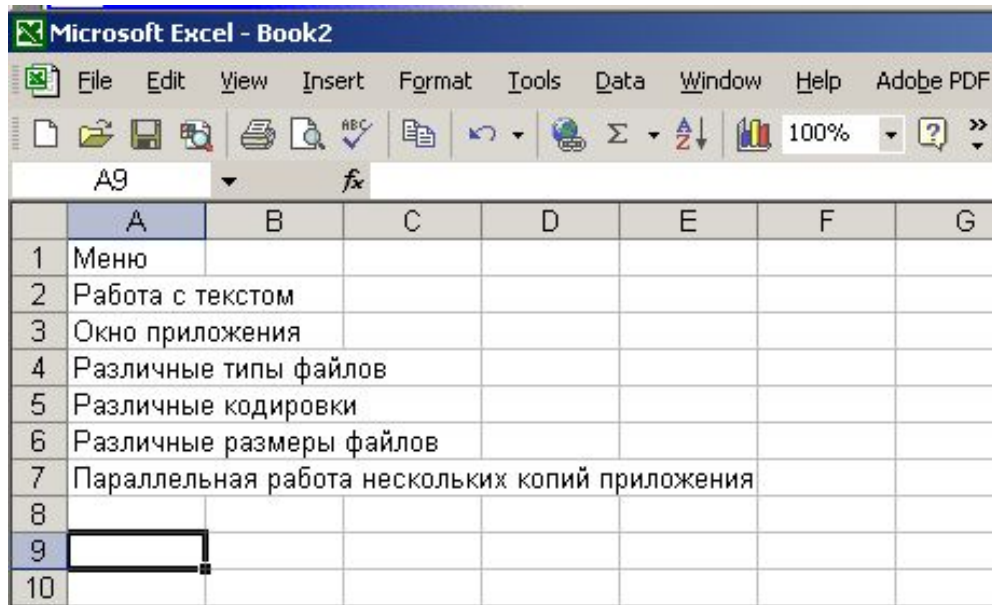
# Пример разработки тестов

Когда мы распишем наши тесты по правилам, Smoke Test примет следующий вид:

Идентификатор	Ссылка на требование	Модуль	Подмодуль/экран	Описание теста	Ожидаемый результат	Статус ("Не тестировано", "выполнено успешно", "выполнение завершилось ошибкой")
ST_001	R1	Приложение не запущено		<b>Запустить приложение</b> 1. Выполнить команду потера из командной строки	1. Появляется окно notepad с пустым файлом	Не тестировано
ST_002	R1, R16	Приложение		<b>Создать новый файл</b> 1. Выполнить последовательность команд "Файл" -> "Создать" с использованием меню	1. Создается новый файл (в рабочей области приложения пусто)	Не тестировано
ST_003	R34, R75.7	Приложение		<b>Ввести текст</b> 1. Набрать несколько слов 2. Удалить несколько слов	1. В рабочей области приложения отображается набранный текст 2. Удаляемые слова пропадают из рабочей области приложения	Не тестировано
ST_004	R23	Приложение	Работа с файлами	<b>Сохранить файл</b> 1. Создать новый файл. Ввести немного текста. 2. Выполнить последовательность команд "Файл" -> "Сохранить" с использованием меню 3. Выбрать каталог для сохранения файла и ввести имя файла 4. Нажать кнопку "Сохранить"	1. Создается новый файл, введенный текст отображается в рабочей области приложения 2. Появляется диалоговое окно "Сохранить файл" <b>Какой каталог для сохранения должен отображаться по умолчанию?</b> 3. Имя файла отображается в строке ввода 4. Диалоговое окно "Сохранить файл" исчезает, на диске в указанном каталоге появляется сохраненный файл	Не тестировано
ST_005	R45, R57, R92	Приложение	Работа с файлами	<b>Распечатать файл</b> 1. Выполнить последовательность команд "Файл" -> "Печать" с использованием меню 2. Следовать инструкциям	1. Открывается диалог "Печать документа" <b>Какова реакция приложения на отсутствие в системе установленных принтеров?</b>	Не тестировано

# Пример разработки тестов

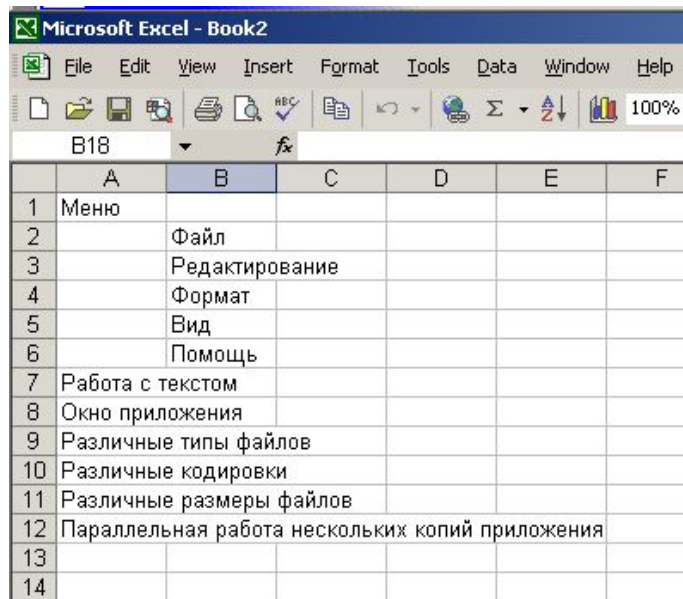
Аналогичным образом начинаем и продолжаем работать с тестом критического пути:



	A	B	C	D	E	F	G
1	Меню						
2	Работа с текстом						
3	Окно приложения						
4	Различные типы файлов						
5	Различные кодировки						
6	Различные размеры файлов						
7	Параллельная работа нескольких копий приложения						
8							
9							
10							

# Пример разработки тестов

Детализируем чек-лист:

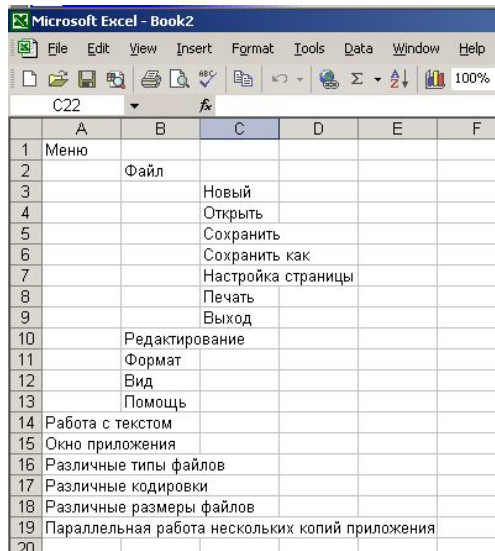


The screenshot shows a Microsoft Excel spreadsheet titled "Microsoft Excel - Book2". The spreadsheet contains a checklist with 14 rows and 6 columns (A-F). The first column (A) contains row numbers 1 through 14. The second column (B) contains the checklist items. The third column (C) is empty. The fourth column (D) is empty. The fifth column (E) is empty. The sixth column (F) is empty. The first row (1) contains "Меню". The second row (2) contains "Файл". The third row (3) contains "Редактирование". The fourth row (4) contains "Формат". The fifth row (5) contains "Вид". The sixth row (6) contains "Помощь". The seventh row (7) contains "Работа с текстом". The eighth row (8) contains "Окно приложения". The ninth row (9) contains "Различные типы файлов". The tenth row (10) contains "Различные кодировки". The eleventh row (11) contains "Различные размеры файлов". The twelfth row (12) contains "Параллельная работа нескольких копий приложения". The thirteenth row (13) is empty. The fourteenth row (14) is empty.

	A	B	C	D	E	F
1	Меню					
2		Файл				
3		Редактирование				
4		Формат				
5		Вид				
6		Помощь				
7		Работа с текстом				
8		Окно приложения				
9		Различные типы файлов				
10		Различные кодировки				
11		Различные размеры файлов				
12		Параллельная работа нескольких копий приложения				
13						
14						

# Пример разработки тестов

Продолжаем детализацию до тех пор, пока не получим логичный и достаточный набор тестов. После этого переносим его в шаблон и работаем аналогично тому, как мы делали это при разработке Smoke Test.



The screenshot shows a Microsoft Excel spreadsheet with a menu structure. The menu items are listed in column A, and their sub-items are listed in column B. The sub-items are: Файл, Новый, Открыть, Сохранить, Сохранить как, Настройка страницы, Печать, Выход, Редактирование, Формат, Вид, and Помощь. The remaining rows in column A list other menu items: Работа с текстом, Окно приложения, Различные типы файлов, Различные кодировки, Различные размеры файлов, and Параллельная работа нескольких копий приложения.

	A	B	C	D	E	F
1	Меню					
2		Файл				
3			Новый			
4			Открыть			
5			Сохранить			
6			Сохранить как			
7			Настройка страницы			
8			Печать			
9			Выход			
10		Редактирование				
11		Формат				
12		Вид				
13		Помощь				
14	Работа с текстом					
15	Окно приложения					
16	Различные типы файлов					
17	Различные кодировки					
18	Различные размеры файлов					
19	Параллельная работа нескольких копий приложения					
20						

A world map is centered on the page, showing the continents in a light blue color. The background of the entire slide is a gradient from dark blue at the top to purple at the bottom. The word "Вопросы?" is written in a white, serif font, centered over the map.

Вопросы?