

# Разработка тестов



# Контрольные вопросы

1. Что такое тестирование ПО.
2. Какие техники тест дизайна знаете.
3. Что представляют собой «классы эквивалентности».
4. Для чего используются «граничные условия».
5. Что такое чек-лист.

# Последовательность разработки и выполнения тестов

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

Для каждого требования необходимо составлять минимум один тест-кейс. Однако, для того, чтобы убедиться, что требование полностью удовлетворяется, может понадобиться несколько тест-кейсов.

# Позитивный/негативный тест

**Позитивное тестирование** – это тестирование с применением сценариев, которые соответствуют нормальному (штатному, ожидаемому) поведению системы. С его помощью можно определить, что система делает то, для чего и была создана.

**Негативным** называют **тестирование**, в рамках которого применяются сценарии, которые соответствуют внештатному поведению тестируемой системы. Это могут быть, например, исключительные ситуации или неверные данные.

# Позитивный/негативный тест

## Пример

### **Классический пример логина:**

- регистрация: ввод логина и пароля, с логином без пароля, через социальные сети;
- авторизация: ввод логина и пароля, вход через социальные сети;
- восстановление пароля.

# Позитивный/негативный тест

## Пример

### *Позитивные сценарии:*

- регистрация через пару логин/пароль, через логин, через социальные сети;
- регистрация при заполнении обязательных полей;
- регистрация при заполнении всех полей;
- проверка возможности авторизации после регистрации;
- регистрация на одном браузере и вход на другом;
- проверка работы функции восстановления пароля.

# Позитивный/негативный тест

## Пример

### *Негативные сценарии:*

- повторная регистрация (имя/пароль уже заняты);
- регистрация без заполнения обязательных полей (все поля пусты);
- регистрация с вводом данных в неверном формате;
- авторизация с неверным логином/паролем.

# Документирование тестов

- Результатом документирования тестов является **тест-кейс**.
- Набор тест-кейсов – **Test Suite**.



# Тестовый случай (*test-case*)

Тесты документируются в виде так называемых "тестовых случаев" (*test-cases*).

Согласно стандарту IEEE Standard Glossary of Software Engineering Terminology, IEEE STD. 610.12:

- ☐ **Тест-кейс (test-case)** - набор тестовых входных данных, условий выполнения и ожидаемых результатов, разработанных с конкретной целью, такой как проверка некоторого пути выполнения программы или проверка соответствия некоторому требованию.
- ☐ **Тест-кейс (test-case)** - это набор входных данных, условий выполнения и ожидаемых результатов, разработанный с целью проверки того или иного свойства или поведения программного средства.

# Зачем нужны тест-кейсы

- «Планирование, и только потом – выполнение!» Тест-кейсы дают структурированный системный подход, что снижает вероятность пропуска ошибки.
- Тест-кейсы – хороший способ хранения части проектной информации.
- Написание тест-кейсов – один из способов протестировать проектную документацию ещё до выхода первого билда.
- Наличие тест-кейсов значительно ускоряет регрессионное тестирование.
- Тест-кейсы – прекрасный способ быстро ввести в курс дела новичка или сотрудника, только что подключившегося к проекту.
- Имея тест-кейсы, можно в любой момент «вспомнить», что делали месяц, полгода, год назад.
- Можно обмениваться тест-кейсами (и «чек-листами») между проектами.
- Тест-кейсы позволяют легко отслеживать прогресс
  - X% тестов выполнено,
  - Y% тестов прошло (завалилось),
  - Z% требований покрыто тестами.

# Свойства тест-кейсов

- Тест-кейсы могут быть:
  - Специфичными или общими.
  - Простыми или сложными.
  - Независимыми или связанными друг с другом.
  - Позитивными или негативными.

# 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.
- Слишком общий тест-кейс сложно выполнять по многим объективным и субъективным причинам, а потому он вполне может остаться невыполненным.
- Однако интеграционные тесты, как правило, бывают более общими, чем иные. Это связано со спецификой интеграционного тестирования.
- Если в тесте прописано много мелких деталей, возрастает время его создания и поддержки.
- Однако недостаток деталей может усложнить работу новичка.

# Простота или сложность

- Рассмотрим на примере. Где в ниже перечисленном простой тест-кейс, а где – сложный?
- **Набор 1:**
  - 1. Откройте файл «1.txt». Файл открыт.
  - 2. Введите слово «Дом». Появляется слово «Дом.
  - 3. Сохраните файл. Кнопка «Сохранить» становится неактивной.
- **Набор 2:**
  - В документе размером более 100 Мб создайте таблицу 100×100, в ячейку 50×50 вставьте картинку размером 30 Мб, применив к ней функцию «Авторасположение». Проверьте результат.

# Простота или сложность

- Простые тесты оперируют за раз одним объектом.
- Каковы **преимущества** простых тест-кейсов?
  - Их легко выполнять.
  - Они понятны новичкам.
  - Они упрощают диагностику ошибки.
  - Они делают наличие ошибки очевидным.
- Каковы **преимущества** сложных тест-кейсов?
  - Больше шансов что-то сломать.
  - Пользователи, как правило, используют сложные сценарии.
  - Программисты сами редко проверяют такие варианты.
- Следует постепенно повышать сложность тестов.

# Независимость или связанность

- Каковы **преимущества** независимого самостоятельного тест-кейса?
  - Его легко и просто выполнить.
  - Такие тесты могут работать даже после краха приложения на других тестах.
  - Такие тесты можно группировать любым образом и выполнять в любом порядке.
- Каковы **преимущества** наборов тесно связанных тестов?
  - Они имитируют работу реальных пользователей.
  - Они удобны для интеграционного тестирования.
  - Они удобны для разбиения на части тестов с большим количеством шагов.
  - Следующий в наборе тест использует данные и состояние приложения, подготовленные предыдущим.
- Промышленным стандартом являются независимые тесты. Использование сценариев не запрещено, но не следует делать их слишком длинными.

# Позитивность или негативность

- Позитивные тесты ...
- Негативные тесты ...



# Что должен содержать тест-кейс (максимум)

- Идентификатор теста (*id*)
- Связанное с тестом требование (*related requirement*)
- Краткое заглавие теста (*title*)
- Модуль и подмодуль приложения, к которым относится тест (*module, submodule*)
- Приоритет теста (*priority: smoke, critical, extended; A, B, C, D*)
- Исходные данные, необходимые для теста (*initial data*) (обычно включается в шаги выполнения)
- Шаги для выполнения теста (*steps*)
- Ожидаемые результаты (*expected results*)
- Поле для пометки, прошёл тест или нет (*status*)
- Последний полученный актуальный результат (*actual result*), связанный с тестом баг (если есть) (*related bug*)
- Указать автора теста (*author*), время последнего выполнения теста (*last time run*) (часто эта информация указывается в заголовке файла)

# Что должен содержать тест-кейс (минимум)

- **Номер тест-кейса** — уникальный идентификатор тест-кейса (такие системы как TestRail, TestLink и подобные автоматически присваивают тест-кейсам уникальные номера). Если у вас тысячи тест-кейсов, то будет удобнее сообщить номер тест-кейса ссылаясь на него, а не пытаться словами рассказать, где и как найти определённый тест-кейс.
- **Заголовок** — краткое, понятное и ёмкое описание сути проверки.
- **Предусловия** — описание действий, которые необходимо предварительно выполнить или учесть, и которые не имеют прямого отношения к проверке.
- **Шаги проверки (выполнения)** — описание последовательности действий, которые необходимо выполнить для проверки.
- **Ожидаемый результат** — проверка, которая устанавливает, что ожидают получить, после выполнения определённых действий в соответствующем шаге.

В зависимости от специфики компании могут присутствовать дополнительные атрибуты для заполнения: приоритет, функциональный блок, программа, ссылка на требование, номер требования и т.д.

# Документирование тестов

Приоритет	Связанное с тестом требование	Заглавие (суть) теста	Ожидаемый результат по каждому шагу
UG_U 1.12	А R97	Галерея, загрузка файла, имя со спецсимволами Приготовление: создать непустой файл с именем #\$\$%^&.jpg	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
Идентификатор	Модуль и подмодуль	Шаги	Исходные данные, необходимые для выполнения теста

# Рекомендации по написанию тест-кейсов

- ❑ Пишите набор для отдельной части приложения (модуля)
- ❑ Помните, что заголовки тестов отражают их суть. Правильно формулируйте и оформляйте заголовки
- ❑ Помните о необходимых приготовлениях к тесту. Описывайте их в ***preconditions***
- ❑ Не повторяйте в нескольких тестах одни и те же шаги, выносите их в ***preconditions***
- ❑ Четко формулируйте ***Expected result***
- ❑ Пишите отдельно набор для *smoke-теста* и *теста критического пути*.
- ❑ Постепенно повышайте сложность тестов.
- ❑ Организуйте сценарий логично.
- ❑ Используйте **один тест для ОДНОЙ проверки**.

# При написании тест-кейсов:

- Используйте активный залог: (*«open», «paste», «click»*). В русском языке используйте безличную форму: «открыть» (вместо «откройте»)
- Описывайте поведение системы: «появляется окно...», «приложение закрывается»
- Если возникают вопросы в ходе разработки тестов, пишите их прямо в документ с тестами, помечая цветом
- Использовать шрифтовые выделения (**ж**, к, *ч*, цвет)
- Используйте простой технический стиль
- **ОБЯЗАТЕЛЬНО** указывайте **ТОЧНЫЕ** названия всех элементов приложения
- Не объясняйте базовые понятия работы с ОС
- Используйте возможности ПО, в котором разрабатываются тесты (фильтр, группировка,...)

# Критерии хорошего тест-кейса

- ✓ Обладает высокой вероятностью обнаружения ошибки.
- ✓ Не выполняет ненужных действий.
- ✓ Не является избыточным по отношению к другим тестам.
- ✓ Исследует соответствующую («ту, которую надо») область приложения.
- ✓ Позволяет легко диагностировать ошибку.
- ✓ Делает обнаруженную ошибку очевидной.
- ✓ Независим (каждый тест-кейс – это индивидуальный сценарий с точкой входа и точкой выхода).

# Тестовый набор (сценарий)

## (*Test Suite*)

- **Тестовый набор** – набор тестов (тест-кейсов), собранных в последовательность для достижения некоторой цели.

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

# Шаги разработки тестовых сценариев

- Сбор информации (требования, мок-апы и т. д.)
- Разделение приложения на модули
- Написание чек-листов
- Написание тест-кейсов

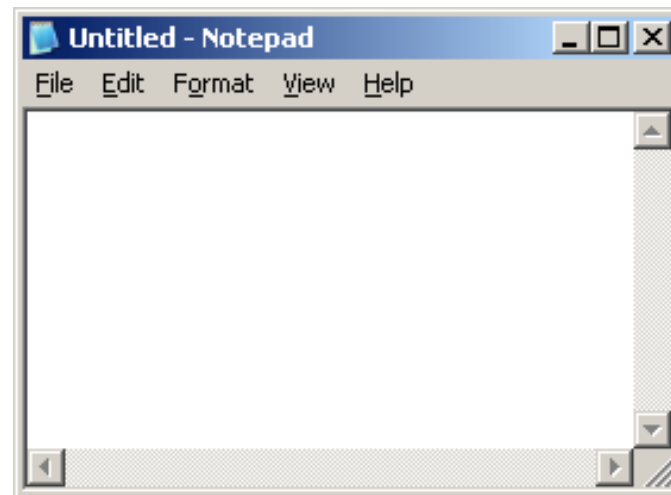


# Шаги разработки тестовых сценариев

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

# Учимся составлять первый тест-кейс

- Что такое notepad?
- Какие функции для него важны?

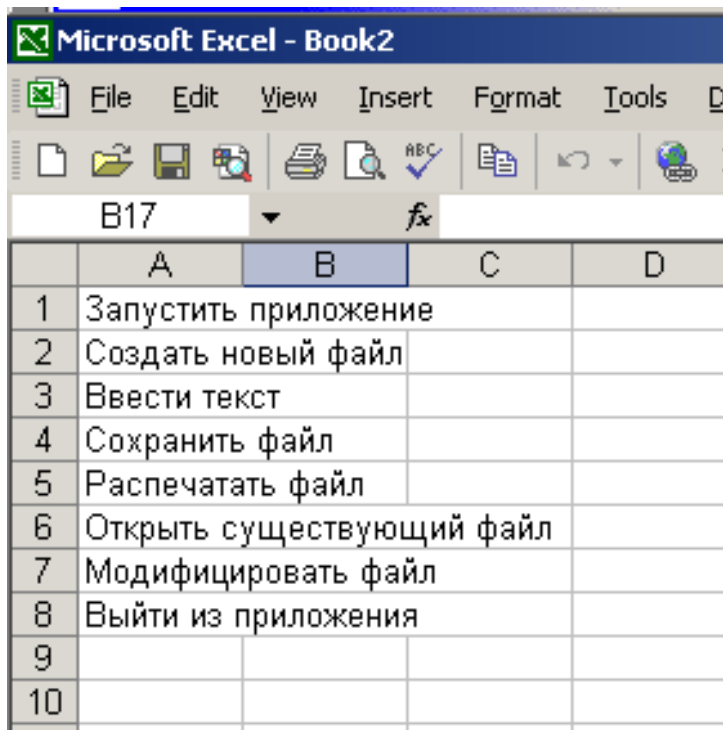


	A	B	C	D
1	Создать файл			
2	Ввести текст			
3	Сохранить			
4	Распечатать			
5				
6				
7				
8				

Что еще?

# Учимся составлять первый тест-кейс

- Итак, вот наш Smoke test



The screenshot shows a Microsoft Excel spreadsheet with a menu bar (File, Edit, View, Insert, Format, Tools) and a toolbar. The active cell is B17. The spreadsheet contains a list of actions in a table format:

	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				

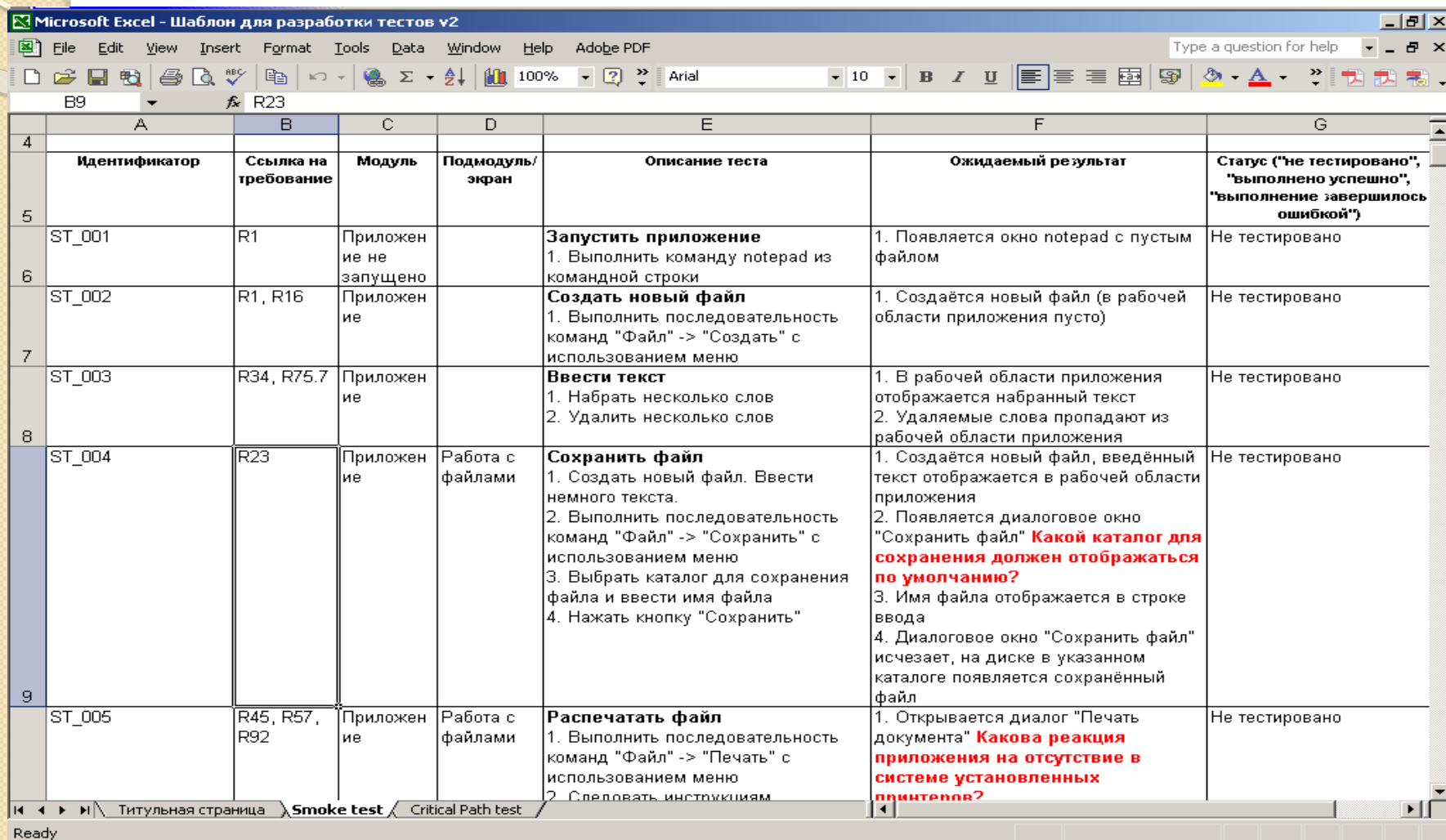
Перенесём его в  
шаблон  
для  
разработки тестов.

Фактически, это – чек-лист. И сами пункты грамотно сформированного чек-листа – готовые заголовки кейсов.

## Разработка тестов

# Учимся составлять первый тест-кейс

Когда мы распишем наши тесты по правилам, Smoke Test (чек-лист) примет следующий вид:

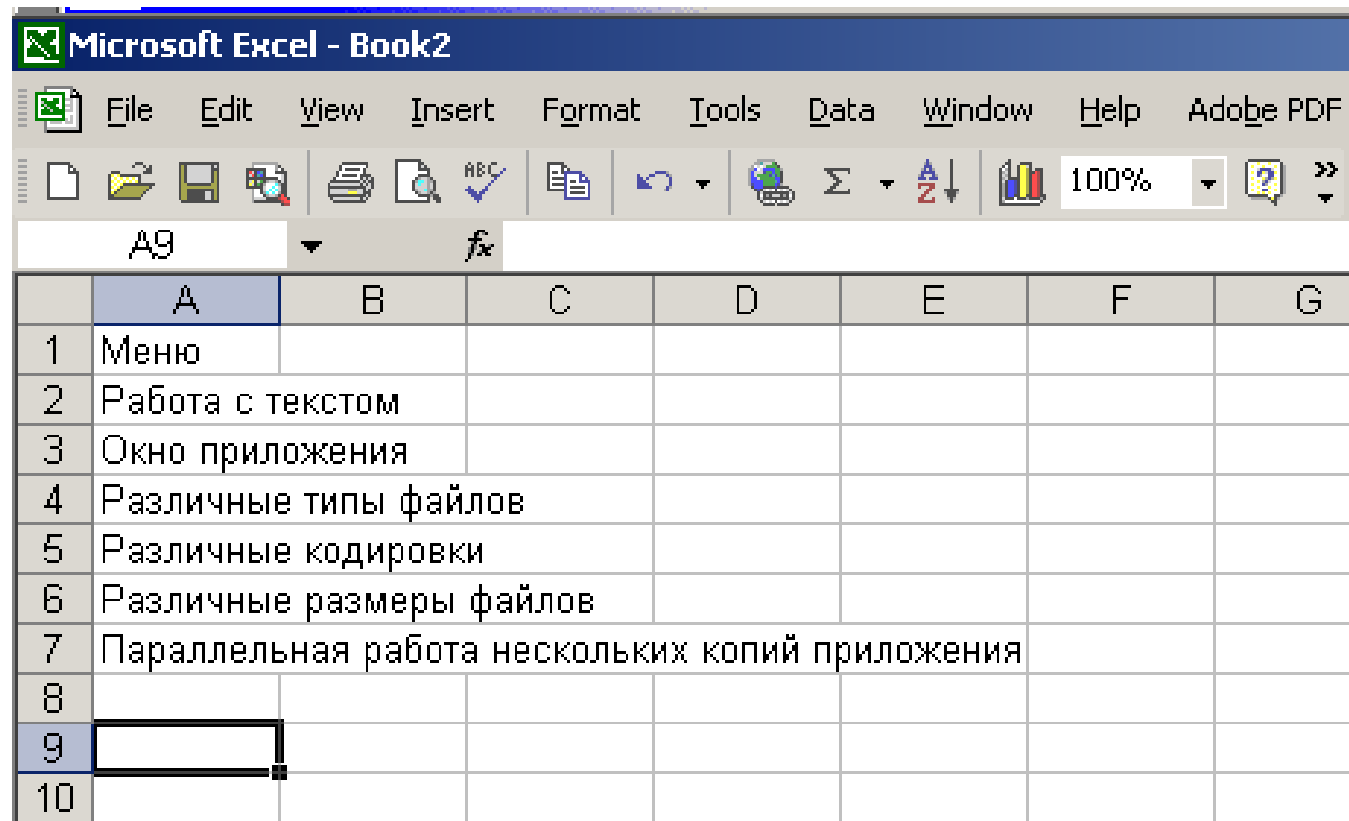


The screenshot shows a Microsoft Excel spreadsheet titled "Шаблон для разработки тестов v2". The spreadsheet is a table with 7 columns and 6 rows of data. The columns are labeled: Идентификатор, Ссылка на требование, Модуль, Подмодуль/экран, Описание теста, Ожидаемый результат, and Статус. The rows contain test cases ST\_001 through ST\_005. The status for all tests is "Не тестировано". The spreadsheet also shows the Excel interface with menus like File, Edit, View, Insert, Format, Tools, Data, Window, Help, and Adobe PDF. The status bar at the bottom indicates "Ready" and "Титульная страница".

	A	B	C	D	E	F	G
	Идентификатор	Ссылка на требование	Модуль	Подмодуль/экран	Описание теста	Ожидаемый результат	Статус ("не тестировано", "выполнено успешно", "выполнение завершилось ошибкой")
4							
5	ST_001	R1	Приложение не запущено		<b>Запустить приложение</b> 1. Выполнить команду notepad из командной строки	1. Появляется окно notepad с пустым файлом	Не тестировано
6	ST_002	R1, R16	Приложение		<b>Создать новый файл</b> 1. Выполнить последовательность команд "Файл" -> "Создать" с использованием меню	1. Создаётся новый файл (в рабочей области приложения пусто)	Не тестировано
7	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
8	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. Создаётся новый файл, введённый 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 сохранённый файл	Не тестировано
9	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

# Учимся составлять первый тест-кейс

Аналогичным образом начинаем и продолжаем работать с тестом критического пути:

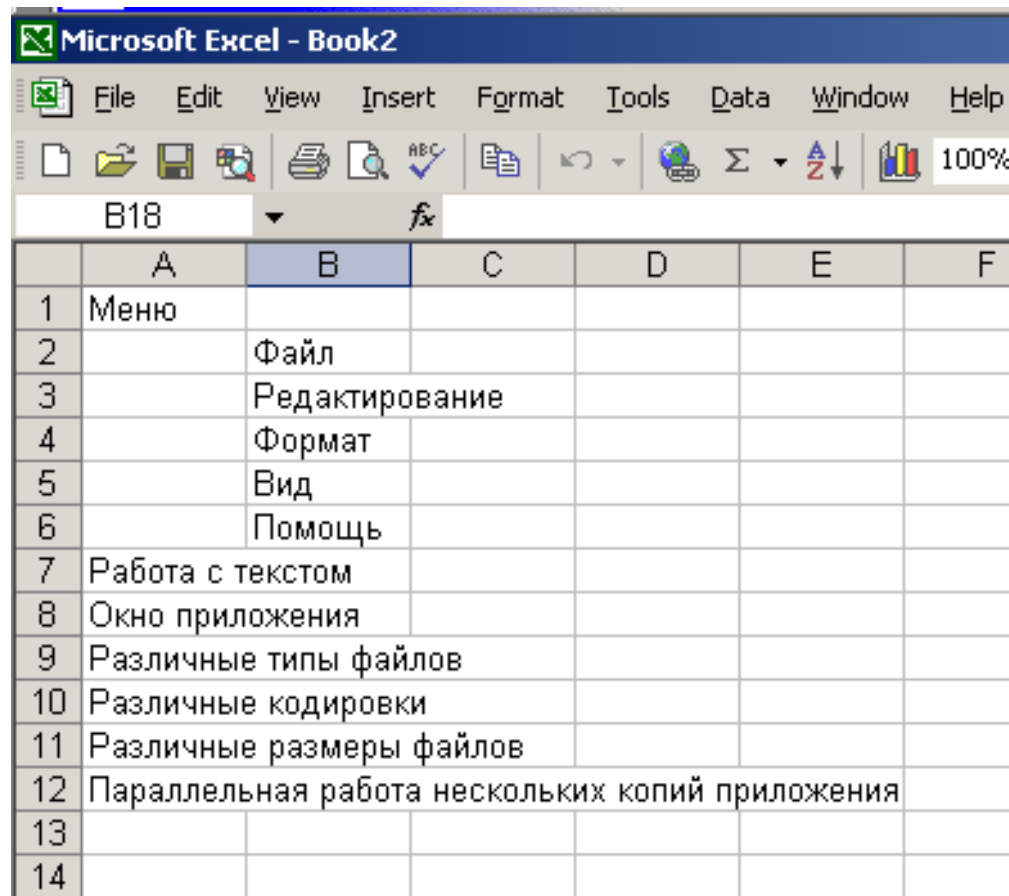


The screenshot shows the Microsoft Excel interface with a menu bar (File, Edit, View, Insert, Format, Tools, Data, Window, Help, Adobe PDF) and a toolbar. The active cell is A9. The table below represents the test case data:

	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							

# Учимся составлять первый тест-кейс

- Детализируем чек-лист:

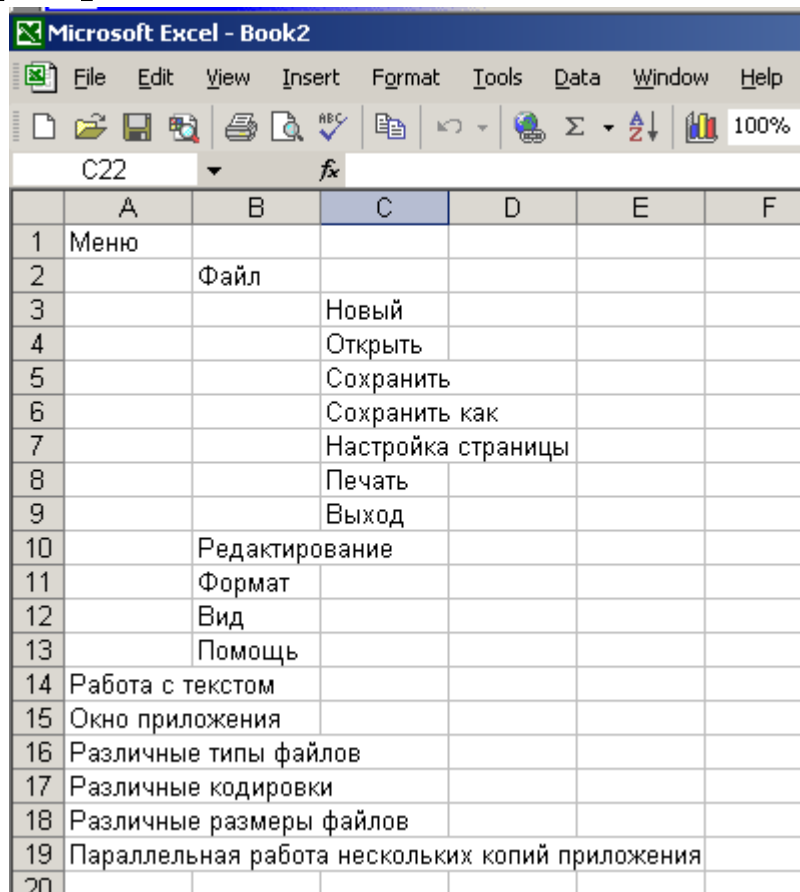


The screenshot shows a Microsoft Excel window titled "Microsoft Excel - Book2". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Window, and Help. The toolbar contains icons for file operations, editing, and formatting. The active cell is B18. The worksheet contains a table with 14 rows and 6 columns (A-F). The table lists various menu items and their sub-items, which serve as test cases.

	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						

# Учимся составлять первый тест-кейс

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 window titled "Microsoft Excel - Book2". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Window, and Help. The toolbar contains various icons for file operations and editing. The active cell is C22. The spreadsheet data is as follows:

	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						



# Контрольные вопросы

1. Определение тест-кейса
2. Свойства тест-кейса.
3. Описать основные разделы тест-кейса.
4. Тестовый набор (сценарий).