Чек-листы и тест-кейсы

Чек-лист (checklist) — набор идей [тест-кейсов].

Чек-лист чаще всего представляет собой обычный и привычный нам список:

- в котором последовательность пунктов не имеет значения (например, список значений некоего поля);
- в котором последовательность пунктов важна (например, шаги в краткой инструкции);
- структурированный (многоуровневый) список, который позволяет отразить иерархию идей.

Важные свойства чек-листов

- Логичность
- Последовательность и структурированность
- Полнота и неизбыточность

Поскольку мы не можем сразу «протестировать всё приложение», нужно выбрать некую логику построения чек-листов, например:

- типичных пользовательских сценариев;
- различных уровней функционального тестирования;
- отдельных частей (модулей и подмодулей) приложения;
- отдельных требований, групп требований, уровней и типов требований;
- частей или функций приложения, наиболее подверженных рискам.

Разбить функции приложения можно, например, по степени их важности:

- **Базовые функции**, без которых существование приложения теряет смысл (т.е. самые важные — то, ради чего приложение вообще создавалось), или нарушение работы которых создаёт объективные серьёзные проблемы для среды исполнения. (См. «Дымовое тестирование»).
- **Функции, востребованные большинством пользователей** в их повседневной работе. (См. «Тестирование критического пути»).
- **Остальные функции** (разнообразные «мелочи», проблемы с которыми не сильно повлияют на ценность приложения для конечного пользователя). (См. «Расширенное тестирование»).

*Рассмотрим на примере тестового приложения из предыдущих лекций

Функции, без которых существование приложения теряет смысл

- Конфигурирование и запуск.
- Обработка файлов:

		Форматы входных файлов		
		TXT	HTML	MD
Кодировки входных файлов	WIN1251	+	+	+
	CP866	+	+	+
	KOI8R	+	+	+

- Остановка

Функции, востребованные большинством пользователей

- Конфигурирование и запуск:
  - С верными параметрами:
    - Значения SOURCE_DIR, DESTINATION_DIR, LOG_FILE_NAME указаны и содержат пробелы и кириллические символы (повторить для форматов путей в Windows и *nix файловых системах, обратить внимание на имена логических дисков и разделители имён каталогов (“/” и “\”)).
    - Значение LOG_FILE_NAME не указано.
  - Без параметров.
  - С недостаточным количеством параметров.
  - С неверными параметрами:
    - Недопустимый путь SOURCE_DIR.
    - Недопустимый путь DESTINATION_DIR.
    - Недопустимое имя LOG_FILE_NAME.
    - DESTINATION_DIR находится внутри SOURCE_DIR.
    - Значения DESTINATION_DIR и SOURCE_DIR совпадают.
- Обработка файлов:
  - Разные форматы, кодировки и размеры: [перечислить]
  - Недоступные входные файлы:
    - Нет прав доступа.
    - Файл открыт и заблокирован.
    - Файл с атрибутом «только для чтения».
- Остановка:
  - Закрытием окна консоли.
- Журнал работы приложения:
  - Автоматическое создание (при отсутствии журнала).
  - Продолжение (дополнение журнала) при повторных запусках.
- Производительность:
  - Элементарный тест с грубой оценкой.

Остальные функции и особые сценарии

- Конфигурирование и запуск:
  - Значения SOURCE_DIR, DESTINATION_DIR, LOG_FILE_NAME:
    - В разных стилях (Windows-пути + *nix-пути) — одно в одном стиле, другое — в другом.
    - С использованием UNC-имён.
    - LOG_FILE_NAME внутри SOURCE_DIR.
    - LOG_FILE_NAME внутри DESTINATION_DIR.
  - Размер LOG_FILE_NAME на момент запуска:
    - 2–4 ГБ.
    - 4+ ГБ.
  - Запуск двух и более копий приложения с:
    - Одинаковыми параметрами SOURCE_DIR, DESTINATION_DIR, LOG_FILE_NAME.
    - Одинаковыми SOURCE_DIR и LOG_FILE_NAME, но разными DESTINATION_DIR.
    - Одинаковыми DESTINATION_DIR и LOG_FILE_NAME, но разными SOURCE_DIR.
- Обработка файлов:
  - Файл верного формата, в котором текст представлен в двух и более поддерживаемых кодировках одновременно.
  - Размер входного файла:
    - 2–4 ГБ.
    - 4+ ГБ.

Тест-кейс и его жизненный цикл: терминология

Тест (test) — набор из одного или нескольких тест-кейсов.

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

Под тест-кейсом также может пониматься соответствующий документ, представляющий формальную запись тест-кейса.

Высокоуровневый тест-кейс (high level test case) — тест-кейс без конкретных входных данных и ожидаемых результатов.

Низкоуровневый тест-кейс (low level test case) — тест-кейс с конкретными входными данными и ожидаемыми результатами.

Тест-кейс и его жизненный цикл: терминология

Спецификация тест-кейса (test case specification) — документ, описывающий набор тест-кейсов (включая их цели, входные данные, условия и шаги выполнения, ожидаемые результаты) для тестируемого элемента (test item, test object).

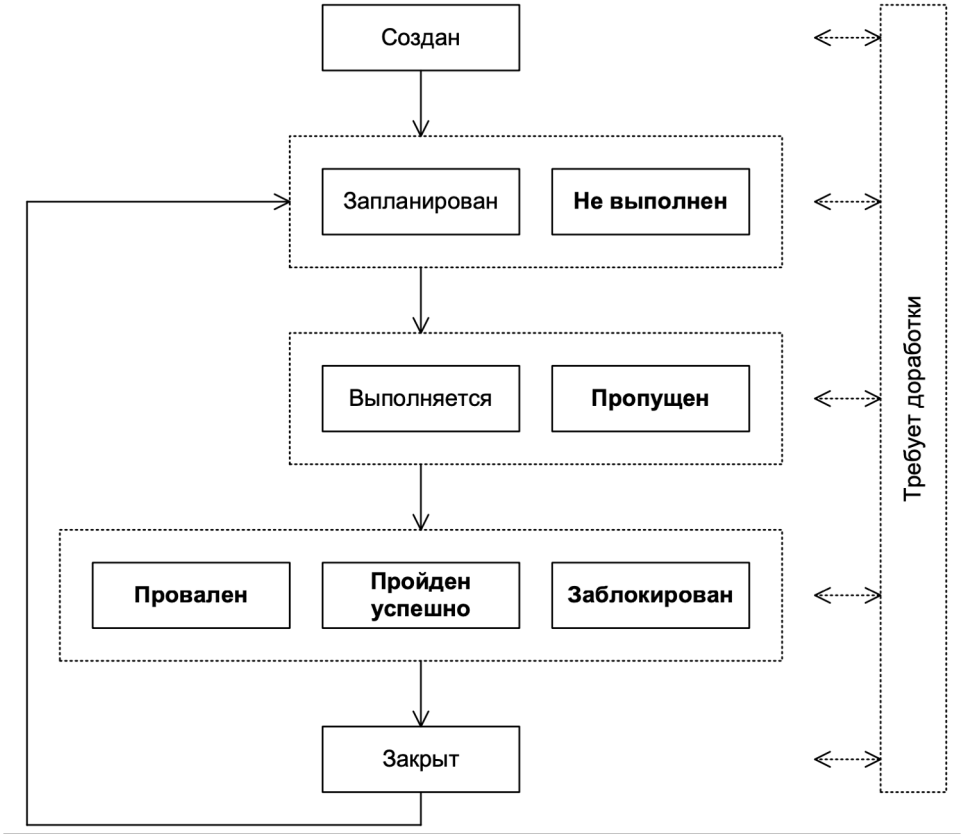
Спецификация теста (test specification) — документ, состоящий из спецификации тест-дизайна (test design specification), спецификации тест-кейса (test case specification) и/или спецификации тест-процедуры (test procedure specification).

Тест-сценарий (test scenario, test procedure specification, test script) — документ, описывающий последовательность действий по выполнению теста (также известен как «тест-скрипт»).

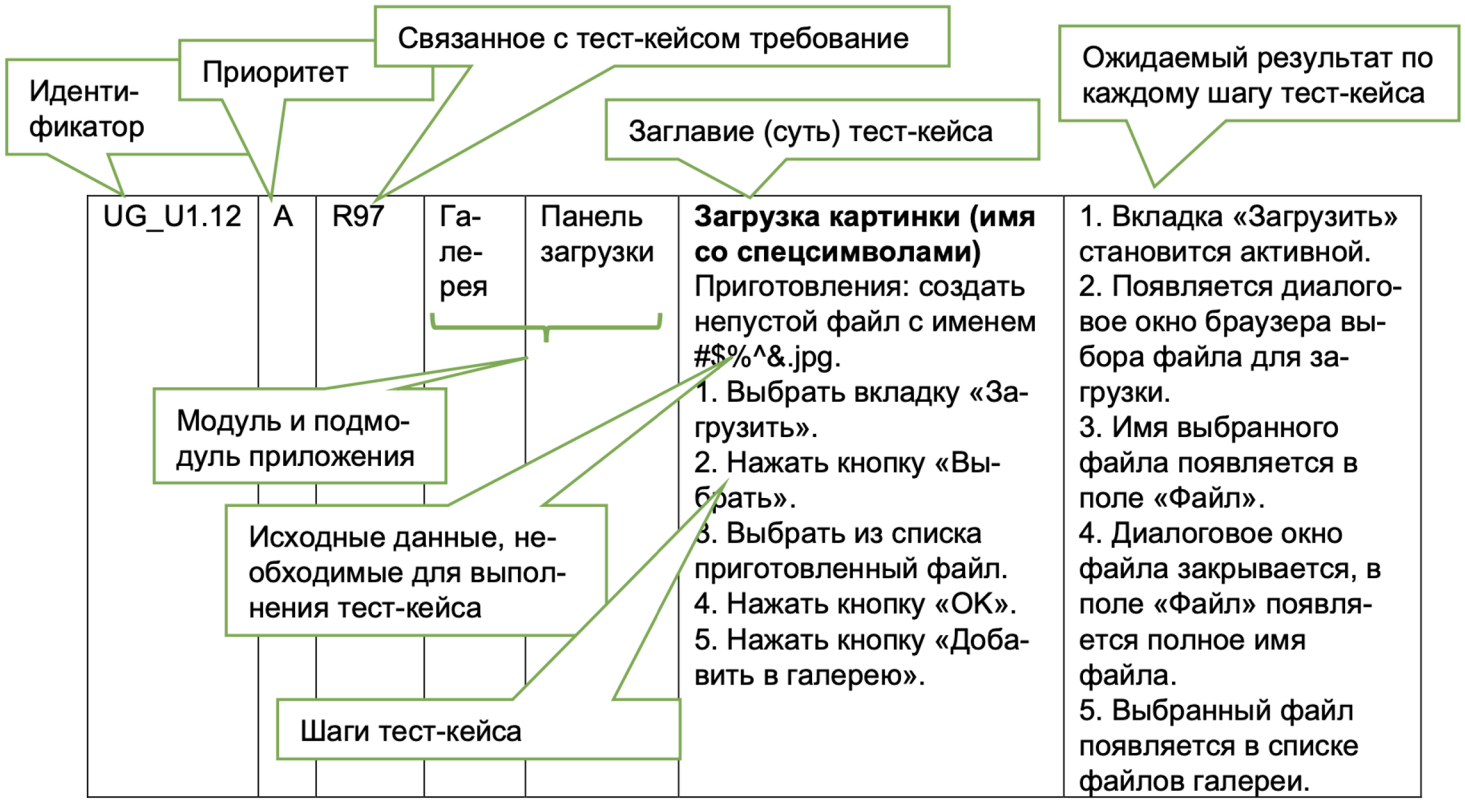
Цель написания тест-кейсов

- **Структурировать и систематизировать** подход к тестированию.
- Вычислять **метрики тестового покрытия** (test coverage metrics) и принимать меры по его увеличению.
- Отслеживать **соответствие текущей ситуации плану** .
- Уточнить **взаимопонимание** между заказчиком, разработчиками и тестировщиками.
- **Хранить информацию** для длительного использования и обмена опытом между сотрудниками и командами.
- Проводить **регрессионное** тестирование и **повторное** тестирование.
- Повышать **качество требований**.
- Быстро **вводить в курс дела** нового сотрудника, недавно подключившегося к проекту.

Жизненный цикл тест-кейса



Атрибуты (поля) тест-кейса



UG_U1.12	A	R97	Галерея	Панель загрузки	**Загрузка картинки (имя со спецсимволами)** Приготовления: создать непустой файл с именем #$%^&.jpg. 1. Выбрать вкладку «Загрузить». 2. Нажать кнопку «Выбрать». 3. Выбрать из списка приготовленный файл. 4. Нажать кнопку «OK». 5. Нажать кнопку «Добавить в галерею».	1. Вкладка «Загрузить» становится активной. 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.

Общие рекомендации по написанию шагов тест-кейса

- начинайте с понятного и очевидного места, не пишите лишних начальных шагов (запуск приложения, очевидные операции с интерфейсом и т.п.);
- даже если в тест-кейсе всего один шаг, нумеруйте его (иначе возрастает вероятность в будущем случайно «приклеить» описание этого шага к новому тексту);
- если вы пишете на русском языке, используйте безличную форму (например, «открыть», «ввести», «добавить» вместо «откройте», «введите», «добавьте»), в английском языке не надо использовать частицу «to» (т.е. «запустить приложение» будет «start application», не «to start application»);
- соотносите степень детализации шагов и их параметров с целью тест-кейса, его сложностью, уровнем и т.д. — в зависимости от этих и многих других факторов степень детализации может варьироваться от общих идей до предельно чётко прописанных значений и указаний;
- ссылайтесь на предыдущие шаги и их диапазоны для сокращения объёма текста (например, «повторить шаги 3–5 со значением...»);
- пишите шаги последовательно, без условных конструкций вида «если... то...».

Общие рекомендации по написанию ожидаемых результатов

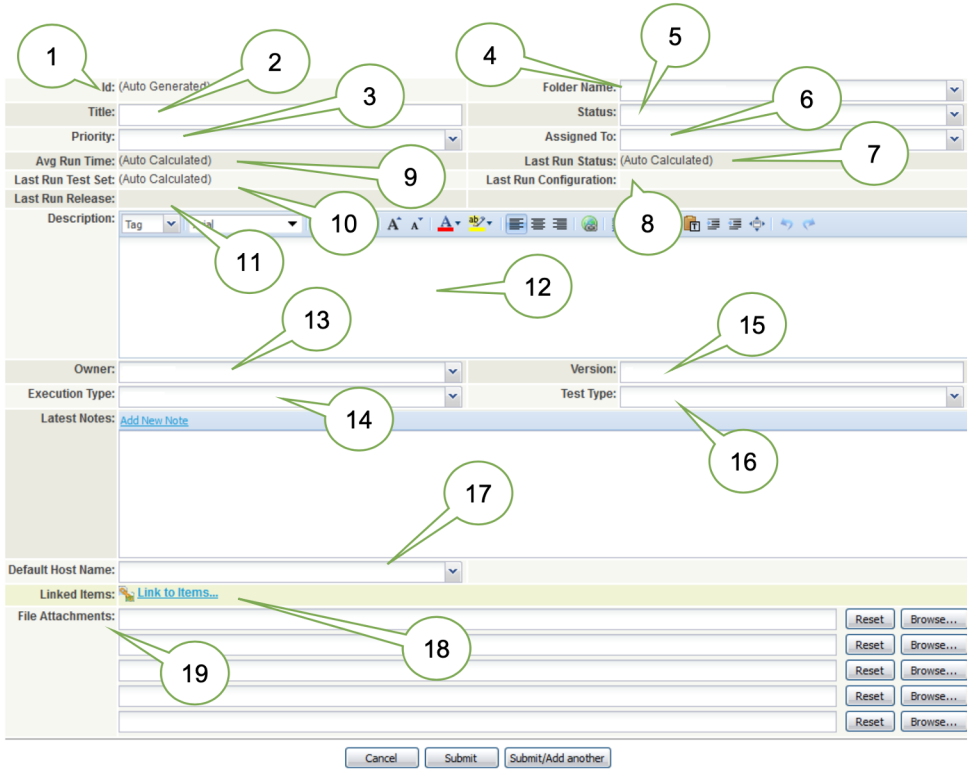
- описывайте поведение системы так, чтобы исключить субъективное толкование (например, «приложение работает верно» — плохо, «появляется окно с надписью...» — хорошо);
- пишите ожидаемый результат по всем шагам без исключения, если у вас есть хоть малейшие сомнения в том, что результат некоего шага будет совершенно тривиальным и очевидным;
- пишите кратко, но не в ущерб информативности;
- избегайте условных конструкций вида «если... то...».

Инструментальные средства управления тестированием

Общий набор функций, реализуемых такими инструментальными средствами:

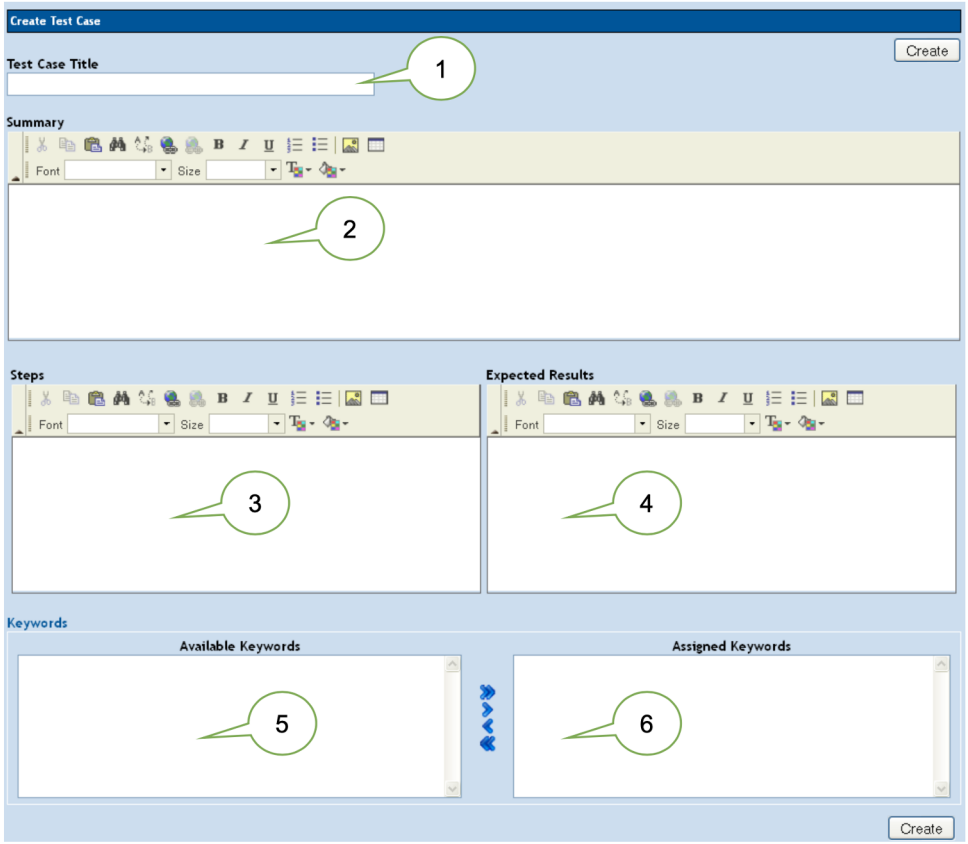
- создание тест-кейсов и наборов тест-кейсов;
- контроль версий документов с возможностью определить, кто внёс те или иные изменения, и отменить эти изменения, если требуется;
- формирование и отслеживание реализации плана тестирования, сбор и визуализация разнообразных метрик, генерирование отчётов;
- интеграция с системами управления дефектами, фиксация взаимосвязи между выполнением тест-кейсов и созданными отчётами о дефектах;
- интеграция с системами управления проектами;
- интеграция с инструментами автоматизированного тестирования, управление выполнением автоматизированных тест-кейсов.

QAComplete

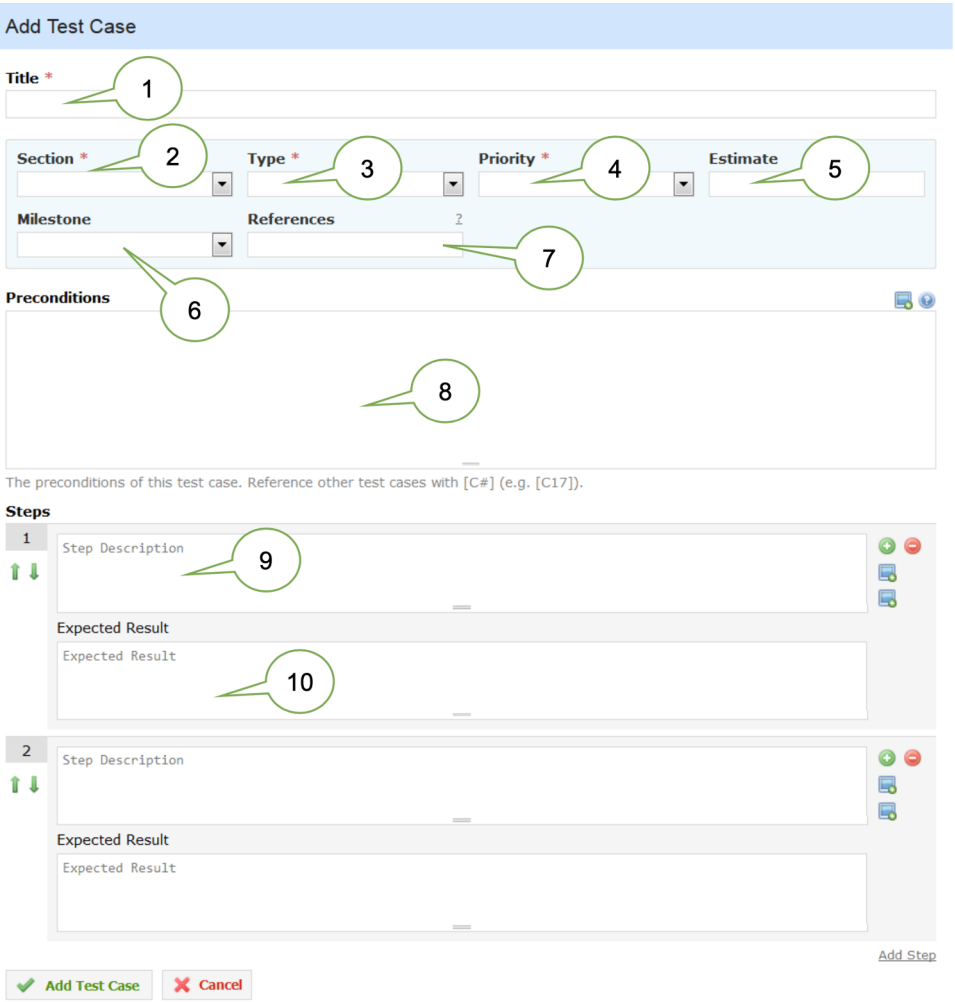

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
Id: (Auto Generated)
Folder Name:
Title:
Status:
Priority:
Assigned To:
Avg Run Time: (Auto Calculated)
Last Run Status: (Auto Calculated)
Last Run Test Set: (Auto Calculated)
Last Run Configuration:
Last Run Release:
Description:
Tag
Owner:
Version:
Execution Type:
Test Type:
Latest Notes:
Add New Note
Default Host Name:
Linked Items:
Link to Items...
File Attachments:
Reset
Browse...
Cancel
Submit
Submit/Add another


Actions
Step #
Critical
Steps
Expected Results
1
2
3

TestLink


Create Test Case
Create
Test Case Title
1
Summary
Font
Size
2
Steps
Expected Results
3
4
Keywords
Available Keywords
Assigned Keywords
5
6
Create

TestRail


Add Test Case
Title *
1
Section *
2
Type *
3
Priority *
4
Estimate
5
Milestone
References
7
6
Preconditions
8
The preconditions of this test case. Reference other test cases with [C#] (e.g. [C17]).
Steps
1
Step Description
9
Expected Result
Expected Result
10
2
Step Description
Expected Result
Expected Result
Add Step
Add Test Case
Cancel

Свойства качественных тест-кейсов

Правильный технический язык, точность и единообразие формулировок.

- пишите лаконично, но понятно;
- используйте безличную форму глаголов (например, «открыть» вместо «откройте»);
- обязательно указывайте точные имена и технически верные названия элементов приложения;
- не объясняйте базовые принципы работы с компьютером (предполагается, что ваши коллеги знают, что такое, например, «пункт меню» и как с ним работать);
- везде называйте одни и те же вещи одинаково (например, нельзя в одном тест-кейсе некий режим работы приложения назвать «графическое представление», а в другом тот же режим — «визуальное отображение», т.к. многие люди могут подумать, что речь идёт о разных вещах);
- следуйте принятому на проекте стандарту оформления и написания тест- кейсов (иногда такие стандарты могут быть весьма жёсткими: вплоть до регламентации того, названия каких элементов должны быть приведены в двойных кавычках, а каких — в одинарных)

Свойства качественных тест-кейсов

Баланс между специфичностью и общностью.

Шаги	Ожидаемые результаты
Конвертация из всех поддерживаемых кодировок Приготовления: • Создать папки C:/A, C:/B, C:/C, C:/D. • Разместить в папке C:/D файлы 1.html, 2.txt, 3.md из прилагаемого архива. 1. Запустить приложение, выполнив команду «php converter.php c:/a c:/b c:/c/converter.log». 2. Скопировать файлы 1.html, 2.txt, 3.md из папки C:/D в папку C:/A. 3. Остановить приложение нажатием Ctrl+C.	1. Отображается консольный журнал приложения с сообщением «текущее_время started, source dir c:/a, destination dir c:/b, log file c:/c/converter.log», в папке C:/C появляется файл converter.log, в котором появляется запись «текущее_время started, source dir c:/a, destination dir c:/b, log file c:/c/converter.log». 2. Файлы 1.html, 2.txt, 3.md появляются в папке C:/A, затем пропадают оттуда и появляются в папке C:/B. В консольном журнале и файле C:/C/converter.log появляются сообщения (записи) «текущее_время processing 1.html (KOI8-R)», «текущее_время processing 2.txt (CP-1251)», «текущее_время processing 3.md (CP-866)». 3. В файле C:/C/converter.log появляется запись «текущее_время closed». Приложение завершает работу.

Шаги	Ожидаемые результаты
Конвертация из всех поддерживаемых кодировок 1. Выполнить конвертацию трёх файлов допустимого размера в трёх разных кодировках всех трёх допустимых форматов.	1. Файлы перемещаются в папку-приёмник, кодировка всех файлов меняется на UTF-8.

Свойства качественных тест-кейсов

Почему плоха излишняя специфичность (тест-кейс 1):

- при повторных выполнениях тест-кейса всегда будут выполняться строго одни и те же действия со строго одними и теми же данными, что снижает вероятность обнаружения ошибки;
- возрастает время написания, доработки и даже просто прочтения тест-кейса;
- в случае выполнения тривиальных действий опытные специалисты тратят дополнительные мыслительные ресурсы в попытках понять, что же они упустили из виду, т.к. они привыкли, что так описываются только самые сложные и неочевидные ситуации.

Почему плоха излишняя общность (тест-кейс 2):

- тест-кейс сложен для выполнения начинающими тестировщиками или даже опытными специалистами, лишь недавно подключившимися к проекту;
- недобросовестные сотрудники склонны халатно относиться к таким тест-кейсам;
- тестировщик, выполняющий тест-кейс, может понять его иначе, чем было задумано автором (и в итоге будет выполнен фактически совсем другой тест- кейс).

Свойства качественных тест-кейсов

Шаги	Ожидаемые результаты
Конвертация из всех поддерживаемых кодировок Приготовления: • Создать в корне любого диска четыре отдельные папки для входных файлов, выходных файлов, файла журнала и временного хранения тестовых файлов. • Распаковать содержимое прилагаемого архива в папку для временного хранения тестовых файлов. 1. Запустить приложение, указав в параметрах соответствующие пути из приготовления к тесту (имя файла журнала — произвольное). 2. Скопировать файлы из папки для временного хранения в папку для входных файлов. 3. Остановить приложение.	1. Приложение запускается и выводит сообщение о своём запуске в консоль и файл журнала. 2. Файлы из папки для входных файлов перемещаются в папку для выходных файлов, в консоли и файле журнала отображаются сообщения о конвертации каждого из файлов с указанием его исходной кодировки. 3. Приложение выводит сообщение о завершении работы в файл журнала и завершает работу.

Свойства качественных тест-кейсов

Баланс между простотой и сложностью.

Преимущества простых тест-кейсов:

- их можно быстро прочесть, легко понять и выполнить;
- они понятны начинающим тестировщикам и новым людям на проекте;
- они делают наличие ошибки очевидным (как правило, в них предполагается выполнение повседневных тривиальных действий, проблемы с которыми видны невооружённым взглядом и не вызывают дискуссий);
- они упрощают начальную диагностику ошибки, т.к. сужают круг поиска.

Преимущества сложных тест-кейсов:

- при взаимодействии многих объектов повышается вероятность возникновения ошибки;
- пользователи, как правило, используют сложные сценарии, а потому сложные тесты более полноценно эмулируют работу пользователей;
- программисты редко проверяют такие сложные случаи (и они совершенно не обязаны это делать).

Свойства качественных тест-кейсов

Шаги	Ожидаемые результаты
Запуск приложения 1. Запустить приложение.	1. Приложение запускается.

Шаги	Ожидаемые результаты
Повторная конвертация Приготовления: • Создать в корне любого диска три отдельные папки для входных файлов, выходных файлов, файла журнала. • Подготовить набор из нескольких файлов максимального поддерживаемого размера поддерживаемых форматов с поддерживаемыми кодировками, а также нескольких файлов допустимого размера, но недопустимого формата. 1. Запустить приложение, указав в параметрах соответствующие пути из приготовления к тесту (имя файла журнала — произвольное).	2. Файлы постепенно перемещаются из входной в выходную папку, в консоли и файле журнала появляются сообщения об успешной конвертации файлов. 3. Файлы постепенно перемещаются из входной в выходную папку, в консоли и файле журнала появляются сообщения об успешной конвертации файлов. 5. Файлы постепенно перемещаются из входной в выходную папку, в консоли и файле журнала появляются сообщения об успешной конвертации файлов допустимого формата и сообщения об игнорировании файлов недопустимого формата.
2. Скопировать в папку для входных файлов несколько файлов допустимого формата. 3. Переместить сконвертированные файлы из папки с результирующими файлами в папку для входных файлов. 4. Переместить сконвертированные файлы из папки с результирующими файлами в папку с набором файлов для теста. 5. Переместить все файлы из папки с набором файлов для теста в папку для входных файлов. 6. Переместить сконвертированные файлы из папки с результирующими файлами в папку для входных файлов.	6. Файлы постепенно перемещаются из входной в выходную папку, в консоли и файле журнала появляются сообщения об успешной конвертации файлов допустимого формата и сообщения об игнорировании файлов недопустимого формата.

Свойства качественных тест-кейсов

Шаги	Ожидаемые результаты
Много копий приложения, конфликт файловых операций Приготовления: • Создать в корне любого диска три отдельные папки для входных файлов, выходных файлов, файла журнала. • Подготовить набор из нескольких файлов максимального поддерживаемого размера поддерживаемых форматов с поддерживаемыми кодировками. 1. Запустить первую копию приложения, указав в параметрах соответствующие пути из приготовления к тесту (имя файла журнала — произвольное). 2. Запустить вторую копию приложения с теми же параметрами (см. шаг 1). 3. Запустить третью копию приложения с теми же параметрами (см. шаг 1). 4. Изменить приоритет процессов второй (“high”) и третьей (“low”) копий. 5. Скопировать подготовленный набор исходных файлов в папку для входных файлов.	3. Все три копии приложения запускаются, в файле журнала появляются последовательно три записи о запуске приложения. 5. Файлы постепенно перемещаются из входной в выходную папку, в консоли и файле журнала появляются сообщения об успешной конвертации файлов, а также (возможно) сообщения вида: a. “source file inaccessible, retrying”. b. “destination file inaccessible, retrying”. c. “log file inaccessible, retrying”. Ключевым показателем корректной работы является успешная конвертация всех файлов, а также появление в консоли и файле журнала сообщений об успешной конвертации каждого файла (от одной до трёх записей на каждый файл). Сообщения (предупреждения) о недоступности входного файла, выходного файла или файла журнала также являются показателем корректной работы приложения, однако их количество зависит от многих внешних факторов и не может быть спрогнозировано заранее.