

Написание тест-кейсов

Уровень

Связанное с тестом требование

Заглавие (суть) теста

Ожидаемый результат по каждому шагу

Идентификатор

Модуль и подмодуль

Исходные данные, необходимые для выполнения теста

Шаги

UG_U 1.12	A	R97	Галерея	Загрузка файла	Галерея, загрузка файла, имя со спецсимволами Приготовление: создать непустой файл с именем #\$\$%^&.jpg 1. Нажать кнопку «Загрузить картинку» 2. Нажать кнопку «Выбрать» 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
--------------	---	-----	---------	----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Преимущества

- «Планирование, и только потом – выполнение!» Тест-кейсы дают нам 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, мы можем в любой момент «вспомнить», что мы делали месяц, полгода, год назад.
- Тест-кейсы позволяют легко отслеживать прогресс (X% тестов выполнено, Y% тестов прошло (завалилось), Z% требований покрыто тестами).

Тест-кейсы могут быть:

- Простыми или сложными.
- Независимыми или связанными друг с другом.
- Позитивными или негативными.

Оба тест-кейса делают одну и ту же проверку. **Какой из них лучше?**

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

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

- 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.
- Если в тесте прописано много мелких деталей, возрастает время его создания и поддержки.
- Однако недостаток деталей может усложнить работу новичка.

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, 1.5, символов и спецсимволов	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.

Где в ниже перечисленном простые тест-кейсы, а где – сложные?

Набор 1:

1. Откройте файл «1.txt». Файл открыт.

Набор 2:

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.

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.

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 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.

Хороший тест-кейс

- Обладает высокой вероятностью обнаружения ошибки.
- Исследует соответствующую («ту, которую надо») область приложения.
- Выполняет какие-то интересные действия.
- Не выполняет ненужных действий.
- Является не слишком простым, но и не слишком сложным.
- Не является избыточным по отношению к другим тестам.
- Делает обнаруженную ошибку очевидной.
- Позволяет легко диагностировать ошибку.

Тестовый сценарий – 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.

Microsoft Excel - Шаблон для разработки тестов v2							
File Edit View Insert Format Tools Data Window Help Adobe PDF Type a question for help							
B9 R23							
	A	B	C	D	E	F	G
4	Идентификатор	Ссылка на требование	Модуль	Подмодуль/экран	Описание теста	Ожидаемый результат	Статус ("не протестировано", "выполнено успешно", "выполнение завершилось ошибкой")
5	ST_001	R1	Приложение не запущено		Запустить приложение 1. Выполнить команду notepad из командной строки	1. Появляется окно notepad с пустым файлом	Не протестировано
6	ST_002	R1, R16	Приложение		Создать новый файл 1. Выполнить последовательность команд "Файл" -> "Создать" с использованием меню	1. Создаётся новый файл (в рабочей области приложения пусто)	Не протестировано
7	ST_003	R34, R75.7	Приложение		Ввести текст 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 протестировано
8	ST_004	R23	Приложение	Работа с файлами	Сохранить файл 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" Какой каталог для сохранения должен отображаться по умолчанию? 3. Имя файла отображается в строке ввода 4. Диалоговое окно "Сохранить файл" исчезает, на диске в указанном каталоге появляется сохранённый файл	Не протестировано
9	ST_005	R45, R57, R92	Приложение	Работа с файлами	Распечатать файл 1. Выполнить последовательность команд "Файл" -> "Печать" с использованием меню 2. Следовать инструкциям	1. Открывается диалог "Печать документа" Какова реакция приложения на отсутствие в системе установленных принтеров?	Не протестировано

Титульная страница | Smoke test | Critical Path test

Ready