

Домашнее задание к занятию

1.3. Тестирование ПО. Многообразие тестирования

Задание 1

Кейс 1

Допустим, вы получили задание протестировать новый лендинг от Альфа-Банка:*

https://alfabank.ru/get-money/credit-cards/100-days/step1-ab/?platformId=alfasite&utm_expId=4mvTmgXHQRKdmLtqOhnZuA.1&utm_referrer=

Делали его срочно, так что вы его видите в первый раз и к тому же на него почти нет документации. Какие техники вы примените и что именно из функционала лендинга вы будете проверять и как?

В первую очередь будем проводить функциональное тестирование дабы оценить основную функциональность на соответствие требованиям (те которые описаны в немногочисленной документации и соответствующие здравому смыслу). Способ выполнения - черный ящик (так как у нас нет доступа к коду), и проверять нам надо не только позитивные кейсы, но и негативные. Из функционала будем проверять заполнение полей ввода граничными значениями и средним. Вводом цифровых значений, пробела, значений содержащими “-” дефис. Их отправку на сервер. Так же проведём нефункциональное тестирование на usability и GUI. Закончим тестом безопасности. И не будем забывать про метод тестирования на основе опыта, так как с лендингом мы встречаемся часто и можем предположить его поведение и возможные проблемы.

Вы верно описали проверки. Как вы думаете, в рамках какого метода будут проводиться эти проверки, с учётом того, для нового лендинга почти нет документации и нужно провести тестирование в сжатые сроки?

В рамках метода исследовательского тестирования (ad-hoc). А мы тогда откажемся от метода чёрного ящика или оставим и нам это скажет что эквивалентные значения мы будем подбирать, а не брать из документации?

Кейс 2

Вам пришло задание на тестирование нового калькулятора ипотеки. Так как это важный функционал, у нас есть все нужное для тестирования - тестовые юзеры, информация про требования и тестовая среда, где мы можем все проверить до запуска продукта. Ваша задача - проверить весь функционал именно калькулятора:

<https://alfabank.ru/get-money/mortgage/programs/digital-ab/#calc> С применением каких техник вы стали бы это делать?

В этом случае так как у нас есть информация про требования, то будем использовать функциональное тестирование с четким соответствием по имеющимся требованиям. На основе имеющейся информации можем провести тестирование методом белого ящика.

Акцентирую ваше внимание на том, что нужно проверить именно калькулятор (есть форма для ввода данных) . Так же, обратите внимание, что у нас нет доступа к коду. Какие техники из рассмотренных в лекции подходят для этой проверки?

Ок, значит определение “всё нужное для тестирования” не включает в себя доступ к коду. Тогда мы используем метод черного ящика и совмещаем его с методом предположения об ошибках на основании информации про требования.

Кейс 3

Представим, что вы уже несколько лет тестируете приложение Альфа-Банка для iOS и вам пришла задача, в которой написано, что разработчики бэкенда переделали систему хранения данных о зарегистрированных пользователях и им важно знать, что ничего не сломалось на фронтенде приложения. Что и по каким методикам вы будете проверять при условии того, что у нас сжатые сроки по тестированию?

Здесь у нас было изменение по этому вид тестирования мы определяем как тест сборки для приемки в дальнейшее тестирование и последующего регрессионного тестирования. А так как над приложением мы уже работаем долго, то используем метод на основе опыта, а именно метод предположения об ошибках и быть может у нас даже есть чек-лист на данный сценарий.

Задание 2

Представим, что мы тестируем форму получения карты банка. В данном случае нам нужно применить знания, полученные на лекции для проверки ввода информации в поле "Имя"

Шаг 1 из 5: Заявка заполнена на 0%

Персональные данные



Мы гарантируем безопасность и сохранность ваших данных

Фамилия

Укажите точно как в паспорте

Имя

Укажите точно как в паспорте

Что нам известно по спецификации: Пользователь должен ввести данные, чтобы оформить заявку. Максимальная длина поля 30 символов, минимальная - 1.

Наша задача - проверить максимальное количество вариантов ввода данных, как валидных для системы, так и невалидных. Совет: задумайтесь над проверкой не только позитивных вариантов, но и негативных неописанных в спецификации, так как наша задача - не только проверить соответствие спецификации, но и изучить ситуации, не описанные в ней.

Для получения зачета нужно как минимум 12 вариантов ввода данных в поле имени.

Результат задания: список вариантов ввода имени.

1. Ничего (оставим поле пустым перед отправкой) (граничное длина 0 символов)
2. Ю (граничное длина 1 символ)
3. Як (граничное длина 2 символа)
4. Михаилмихаилмих (эквивалентное длина 15 символов)
5. МихаилмихаилмихМихаилмихаилми (граничное длина 29 символов)
6. МихаилмихаилмихМихаилмихаилмиха (граничное длина 31 символов)
7. Mikhail (проверка латиницы)
8. ᲛᲗᲗ (неюникодныe символы)
9. Амир-Сеит (Имена с тире)

- 10. NULL (проверка на экранирования входных данных)
- 11. X Æ A-12 (проверка на ввод цифр)
- 12. “ “ пробел
- 13. !";№;:?*_((спецсимволы)