Здравствуйте, меня зовут Куликов Петр Александрович, и это решение моего тестового задания на позицию «Тестировщик» в ООО «Сигнатек», расположенного по адресу: <https://talantix.ru/ats/testTasks/7e730fc06f9f40a68c1738cd7dffb9cf>

Формулировка задания скудная, и по-хорошему, я не могу приступить к тестированию ПО, не задав уточняющие вопросы о спецификации ПО, требований к нему, по выделенным срокам, приоритетам и требованиям к тестированию. Свое решение я буду строить из предположения, что я уточнил требования у аналитика/заказчика или прочитал их в документах. Даже если Вы работаете по Agile, документация все равно должна быть, пусть и не исчерпывающая.

И начну с первого же предложения в тестовом задании: «Исследовать эту программу на наличие ошибок.». В архиве присутствует две программы: «tri-bach2.exe» и «tri-bach2.xex». Исполняемые файлы для Windows и Xbox(как я понял), соответственно. Какую из двух программ тестировать не указано, поэтому предположим, что я уточнил требование, и нужно тестировать программу «tri-bach2.exe» и её графический, а не консольный интерфейс.

Немаловажный вопрос заключается в том, что это за программа и для чего она сделана(особенно, если поднимать вопросы о юзабилити-тестировании и тестировании стабильности), и в спецификации должен быть дан на это ответ. К этому вопросу я вернусь позже.

Начну со смоук-тестов, исходя из требований, указанных в текстовом файле.

Сделаем вид, что я уточнил у аналитика/заказчика требования, что программа должна запускаться на моем ПК с моей сборкой, что консольное окно программы открывается и сворачивается, и показывается графический интерфейс, который соответствует мокапу.

Смоук-тесты проверяют, что программа запускается, что открывается графическое и консольное окна, что графический интерфейс соответствует ожидаемому, что программа правильно работает с валидными данными для равностороннего, равнобедренного и разностороннего треугольников, что программа закрывается. Сами тест-кейсы находятся в прикрепленной excel-таблице.

Проводить функциональное тестирование самой программы я буду используя таблицу решений и разбивая значения переменных на классы эквивалентности. К сожалению, в спецификации указано только то, что это 3 положительные значения, но не указано, могут ли они быть только целыми, или могут быть десятичными, с какой точностью десятичными, и какое у них может быть максимальное значение, какая должна быть точность у треугольника, изображенного в Picturebox.

Допустим, я уточнил требования у заказчика/аналитика, и они оказались следующими:

1. Точность значений сторон – 1 знак после запятой. Диапазон значений «такой-то»(см ниже)
2. Изображение в Picturebox носит исключительно иллюстративный характер и не отражает абсолютную точность (надо подумать, как это правильно сформулировать)
3. Изображение треугольника в Picturebox должно соответствовать по цвету и форме указанным данным.
4. В случае невозможности построения треугольника, поле Picturebox должно оставаться пустым.

В случае «таких-то» допустимых значений, я должен проверить значения на границах диапазонов, но, поскольку у меня их нет, я этого делать не буду. Пальцем тыкать в небо и проводить неосмысленные тесты нет смысла. На настоящем проекте должны быть указаны значения, в пределах которых программа должна работать.

В эксель-файле на листе «таблица решений» предоставлены, по сути, тест-кейсы. По сути, первые 6 тестов можно было объединить в три, но, поскольку текстбокс «Side B» забагован, я решил разделить тесты.

Негативных тестов вышло больше, чем позитивных. По-хорошему, так не должно быть, и в этот список тестов стоит добавить ещё позитивных тестов на границах значений.

В эксель-файле на листе «баг-репорт» представлен баг-репорт. По-хорошему, баг-репорты стоит вносить в баг-трекинговую программу. Поэтому количество полей может отличаться от Ваших ожиданий. Замечу, что все найденные баги связаны с визуальным отображением. В зависимости от спецификации, им могут быть присвоены разные степени серьезности.

На этом с функциональным тестированием я закончу. Конечно, ещё много тестов можно придумать, но для них нужно собрать требования.

Про нефункциональное тестирование тоже много вопросов:

1. Соответствует ли дизайн мокапу?
2. Для кого предназначается ПО?
3. Какие требования к безопасности?
4. Какие требования к производительности?
5. Какие требования к инсталляции?
6. Какие требования к юзабилити-тестированию?
7. Какие требования к локализации?

Без ответов на эти вопросы, «другое» тестирование невозможно. Дизайн для меня – сойдет. Для детей, которые не знают английский язык – не подойдет. К тому же, некоторых может смутить консольное окно

Про безопасность – заметил, что файл логов создается не в корневой папке, а на уровень выше. К тому же, будь у меня открыта эта программа на диске С, он бы не смог создаться из-за недостаточных прав пользователей.

Чтобы провести нефункциональное тестирование, нужно гораздо-гораздо больше документации.

**Заключение о стабильности работы программы:**

Как и гласит принцип кластеризации дефектов, могу сказать, что бОльшая часть дефектов находится в логике, связанной с Textbox “Side B” и с Picturebox. Программа к релизу не готова, а силы тестирования стоит направить на тесты Textbox “Side B”.