Задание 1. Напишите эссе на тему «Цикл разработки ПО. Роль меня, как тестировщика, в этом процессе».

Жизненный цикл ПО (SDLC- Software Development LifeCycle)

Первый этап - идея. Т.е. некоторый бизнес понимает, что у его пользователя (потребителя) есть некоторая потребность в продукте, этот продукт действительно выстрелит. Он собирает некоторые данные по тому, что он хочет всё-таки показать своему пользователю, и у него появляется идея.
Второй этап – разработка и сбор требований. С этой идеей он приходит в IT компанию и, если возникает договоренность о том, что этот продукт будут разрабатывать в этой компании, к делу подключаются бизнес□аналитики и начинают собирать требования. Они общаются с заказчиком, с конечным пользователем, формируют пул требований, которые нужно будет реализовать в конечном продукте.
Третий этап – дизайн. Дальше уже подключаются web-дизайнеры, которые на основании требований рисуют (если мы говорим о web-приложении), к примеру, мокапы, либо шаблоны того сайта, который хочет видеть наш заказчик.
Четвёртый этап – разработка. Если у заказчика нет вопросов к дизайну, тогда подключаются разработчики. Начинается непосредственно разработка.
Пятый этап – тестирование. Разработчики пишут код, который в дальнейшем попадает на тестирование к тестировщику.
Шестой этап – ввод в эксплуатацию. После того, как продукт протестирован, он передается в эксплуатацию конечного пользователя.
Седьмой этап – вывод из эксплуатации. Если бизнес понимает, что этот продукт больше не востребован, либо придумывает что-то новенькое и забивает на поддержку старого продукта, происходит вывод из эксплуатации.

Моя роль как тестировщика, заключается в том, чтобы как можно раньше начать тестировать продукт, т.е. на этапе разработки и сбора требований. Так же я буду тестировать продукт на каждом последующем этапе, чтобы сделать продукт качественным и уменьшить вероятность появления багов.

Задание 2. Представьте, что Вы – тестировщик. Вам поступила обратная связь от пользователя: «Я хотел авторизоваться в вашем приложении. Указал свои логин, пароль и нажал на кнопку [Войти], но программа выдала ошибку – error. Скриншот прилагаю.»

Необходимо ответить на сообщение пользователя следующее:

«Здравствуйте, спасибо за ваше обращение. Пожалуйста ответьте на вопросы и выполните необходимые действия, чтобы мы смогли лучше ознакомится с вашей проблемой. Прошли ли вы регистрацию в нашем приложении? Проверьте пожалуйста работу вашего интернета, обновите приложение.

Задание 3. Если автотест выполнился корректно, то он зеленый. Если автотест не зеленый, то он написан тестировщиком Джоном. Все автотесты либо выполнились корректно, либо красные. Если автотест красный, то он написан тестировщиком Кеном.
Выберите правильный ответ и подробно опишите свои рассуждения:
A. Все автотесты – красные
B. Все автотесты – написаны тестировщиком Кеном
C. Все автотесты – зеленые
D. Все автотесты – не зеленые
E. Нет правильного ответа

Ответ: Будем исходить из ответов:

Если выбрать ответ A, C и D, то в данных случаях нельзя утверждать, что все автотесты красные, так как не приведен результат выполнения автотестов.

Если выбрать вариант B (все автотесты написаны тестировщиком Кенном) – в задание указано, что автотесты так же могут быть написаны тестировщиком Джоном.

Ответ: E

Задание 4. Возраст матери и старшей дочери в сумме составляет 55 лет, при этом цифры в возрасте матери и дочери – одинаковы. В возрасте младшей дочери – те же цифры, которые будут в возрасте матери через количество лет, равное возрасту старшей дочери минус возраст младшей дочери сейчас. Сколько лет матери и дочерям?

Методом подбора (следуя из суммы возраста, которая равна 55) можно подобрать 2 комбинации, а именно (23; 32) и (41;14).

Методом исключения (исследуя возраст младшей дочери) комбинация (41;14) отпадает.

Проверяем комбинацию (32;12):

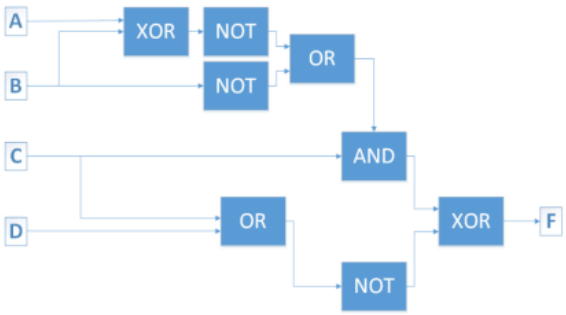
32 – возраст матери

23 – возраст средней дочери

14 – возраст младшей сестры

Ответ: (32;12;14)

Задание 5. Дана логическая схема. При каких комбинациях A, B, C и D функция F принимает истинное значение? Распишите, как пришли к такому решению?



Начнем с конца: если F => True значит на оператор XOR (стоящий перед функцией) должно приходить (0;1) или (1;0). Начнем с первой комбинации (0;1), где ноль – выход с оператора AND, а единица – выход с оператора NOT.

Рассмотрим нижнюю часть схемы (3 и 4 коэффициенты) с оператором NOT и будем двигаться с конца в начало. Так как на выходе оператора NOT – единица, следовательно, на входе – ноль, а значит на выходе оператора OR – ноль. Если на выходе ноль, значит на входе должны быть биты равные (0;0). Следовательно, получаем набор с известными 3 и 4 битами, а именно (X,У,0,0).

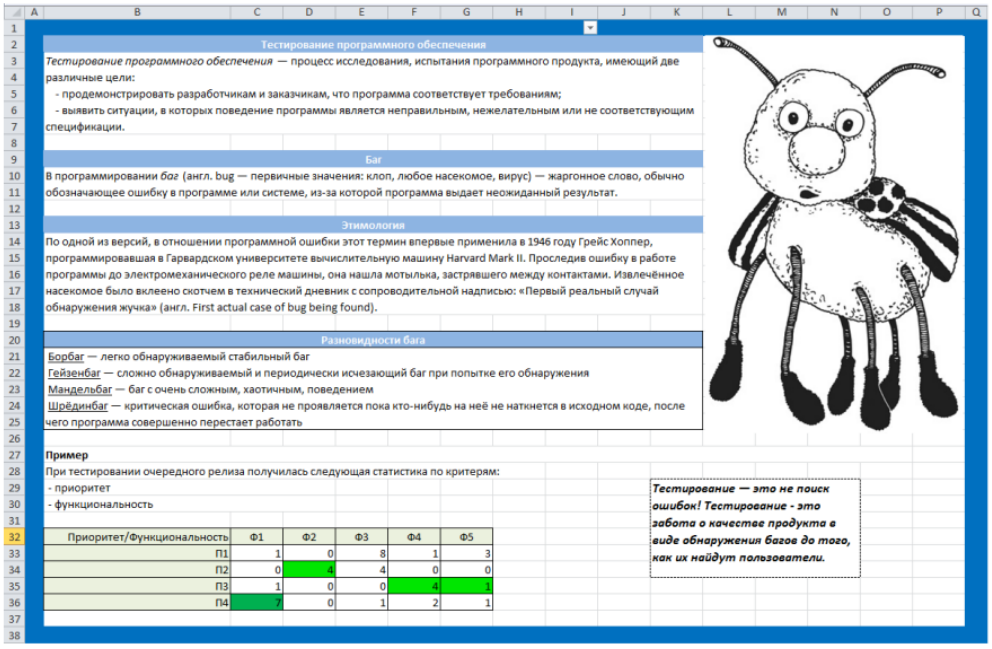
Далее перейдем к рассмотрению верхней части схемы с 1 и 2 битами. Так как на выходе с оператора AND – ноль, а 3 коэффициент равен 0, следовательно, на другом входе должны быть 0, либо 1. Рассмотрим первый случай, когда на выходе с оператора OR к оператору AND – ноль. Значит на входе в оператор OR должна быть пара (0;0). Отсюда следует, что бит B = единица. Другой вход на оператор OR тоже ноль, значит выход с оператора XOR = 1, отсюда следует что имеется две пары битов (0;1) и (1;0). Так как бит B = 1, значит вторая пара не имеет смысла. Исходя из вышесказанного получаем первый набор битов, при которых функция F принимает истинное значение, (0;1;0;0).

Перейдем к рассмотрению второго случая, когда на другом входе в оператор AND не ноль, а единица. Так как на выходе OR (к оператору AND) единица, значит на входе мы имеем три комбинации (1;0), (0;1), (0;0). Исследуем первую (1;0), B = 1 (проходя через оператор NOT, получаем на выходе 0). Так как на выходе оператора XOR должен быть ноль (так как на воде в OR – единица), то бит A должен быть равен единице. Отсюда получаем второй набор битов, при которых функция F принимает истинное значение (1;1;0;0).

Проделывая все манипуляции со всевозможными комбинациями получаем десять комбинаций при которых функция F принимает истинное значение:

0	0	1	0
0	0	1	1
1	1	1	0
1	1	1	1
1	0	1	0
1	0	1	1
1	0	0	0
0	1	0	0
0	0	0	0
1	1	0	0

Задание 6. На картинке ниже имеются различия. Укажите найденные различия, выделив их на картинке.



Тестирование программного обеспечения

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий две различные цели:

- продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям;
- выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

Баг

В программировании *баг* (англ. bug — первичные значения: клоп, любое насекомое, вирус) — жаргонное слово, обычно обозначающее ошибку в программе или системе, из-за которой программа выдает неожиданный результат.

Этимология

По одной из версий, в отношении программной ошибки этот термин впервые применила в 1946 году Грейс Хоппер, программировавшая в Гарвардском университете вычислительную машину Harvard Mark II. Проследив ошибку в работе программы до электромеханического реле машины, она нашла мотылька, застрявшего между контактами. Извлечённое насекомое было вклеено скотчем в технический дневник с сопроводительной надписью: «Первый реальный случай обнаружения жучка» (англ. First actual case of bug being found).

Разновидности бага

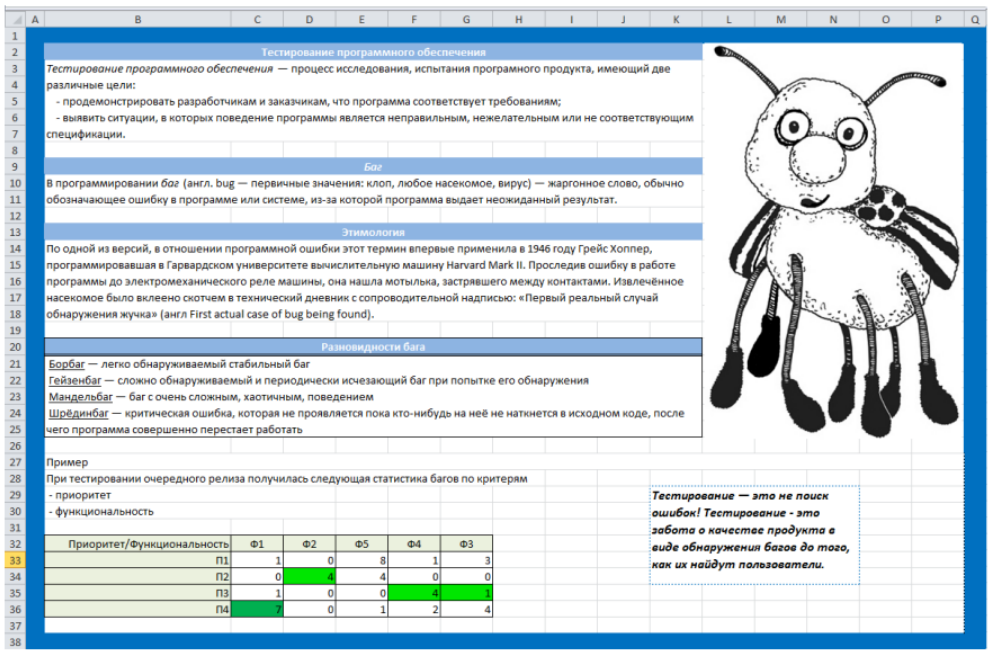
Борбаг — легко обнаруживаемый стабильный баг
Гейзенбаг — сложно обнаруживаемый и периодически исчезающий баг при попытке его обнаружения
Мандельбаг — баг с очень сложным, хаотичным, поведением
Шрёдинбаг — критическая ошибка, которая не проявляется пока кто-нибудь на неё не наткнется в исходном коде, после чего программа совершенно перестает работать

Пример

При тестировании очередного релиза получилась следующая статистика по критериям:
- приоритет
- функциональность

Приоритет/Функциональность	Ф1	Ф2	Ф3	Ф4	Ф5
П1	1	0	8	1	3
П2	0	4	4	0	0
П3	1	0	0	4	1
П4	7	0	1	2	1

Тестирование — это не поиск ошибок! Тестирование - это забота о качестве продукта в виде обнаружения багов до того, как их найдут пользователи.



Тестирование программного обеспечения

Тестирование программного обеспечения — процесс исследования, испытания програмного продукта, имеющий две различные цели:

- продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям;
- выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

Баг

В программировании *баг* (англ. bug — первичные значения: клоп, любое насекомое, вирус) — жаргонное слово, обычно обозначающее ошибку в программе или системе, из-за которой программа выдает неожиданный результат.

Этимология

По одной из версий, в отношении программной ошибки этот термин впервые применила в 1946 году Грейс Хоппер, программировавшая в Гарвардском университете вычислительную машину Harvard Mark II. Проследив ошибку в работе программы до электромеханического реле машины, она нашла мотылька, застрявшего между контактами. Извлечённое насекомое было вклеено скотчем в технический дневник с сопроводительной надписью: «Первый реальный случай обнаружения жучка» (англ First actual case of bug being found).

Разновидности бага

Борбаг — легко обнаруживаемый стабильный баг
Гейзенбаг — сложно обнаруживаемый и периодически исчезающий баг при попытке его обнаружения
Мандельбаг — баг с очень сложным, хаотичным, поведением
Шрёдинбаг — критическая ошибка, которая не проявляется пока кто-нибудь на неё не наткнется в исходном коде, после чего программа совершенно перестает работать

Пример

При тестировании очередного релиза получилась следующая статистика багов по критерям
- приоритет
- функциональность

Приоритет/Функциональность	Ф1	Ф2	Ф5	Ф4	Ф3
П1	1	0	8	1	3
П2	0	4	4	0	0
П3	1	0	0	4	1
П4	7	0	1	2	4

Тестирование — это не поиск ошибок! Тестирование - это забота о качестве продукта в виде обнаружения багов до того, как их найдут пользователи.

Ответ:

Тестирование программного обеспечения

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий две различные цели:

- продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям;
- выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

Баг

В программировании *баг* (англ. bug — первичные значения: клоп, любое насекомое, вирус) — жаргонное слово, обычно обозначающее ошибку в программе или системе, из-за которой программа выдает неожиданный результат.

Этимология

По одной из версий, в отношении программной ошибки этот термин впервые применила в 1946 году Грейс Хоппер, программировавшая в Гарвардском университете вычислительную машину Harvard Mark II. Проследив ошибку в работе программы до электромеханического реле машины, она нашла мотылька, застрявшего между контактами. Извлечённое насекомое было вклеено скотчем в технический дневник с сопроводительной надписью: «Первый реальный случай обнаружения жучка» (англ First actual case of bug being found).

Разновидности бага

Бербаг — легко обнаруживаемый стабильный баг

Гейзенбаг — сложно обнаруживаемый и периодически исчезающий баг при попытке его обнаружения

Мандельбаг — баг с очень сложным, хаотичным, поведением

Шрёдинбаг — критическая ошибка, которая не проявляется пока кто-нибудь на неё не наткнется в исходном коде, после чего программа совершенно перестает работать

Пример

При тестировании очередного релиза получилась следующая статистика багов по критериям

- приоритет
- функциональность

Приоритет/Функциональность	Ф1	Ф2	Ф5	Ф4	Ф3
П1	1	0	8	1	3
П2	0	4	4	0	0
П3	1	0	0	4	1
П4	7	0	1	2	4

Тестирование — это не поиск ошибок! Тестирование - это забота о качестве продукта в виде обнаружения багов до того, как их найдут пользователи.