|  |
| --- |
| **ЗАДАНИЕ № 1** |
| **3 принцыпа тестирования.** |
|  |
| **Тестирование зависит от контекста.** |
| Тестирование выполняется по-разному в зависимости от контекста. Например, программное обеспечение управления производством, в котором критически важна безопасность, тестируется иначе, чем мобильное приложение электронной коммерции. Предположим, нам нужно проверить интернет магазин. Мы будем сосредоточены на функционале платежей, правильности работы корзины и данных для ввода для платежа. Если же мы тестируем медицинский сайт поликлиники, то тут мы будет делать акцент на данных (клиентах), корректность выполнения запросов и ответов базы данных. |
|  |
| **Исчерпывающее тестирование не возможно.** |
| Полное тестирование с использованием всех комбинаций вводов и предусловий физически невыполнимо. Например: Есть поле для ввода — без ограничений и валидации — мы можем писать туда что угодно: числа, символы, буквы любого алфавита, emoji , да еще и текст любой длины. Таким образом количество колитчество проверок будет равно бесконечности, что и подтверждает невозможность исчерпывающего тестирования. Именно для упрощения таких ситуаций необходимо учитывать тест-анализ, анализ рисков и приоритезацию проверок, а так же применение определенных методов тестирования, которые позволят сократить количество тестов к минимуму не ухудшая качество выпускаемого продукта. |
|  |
| **Раннее тестирование сохраняет время и деньги.** |
| К примеру:  Первый вариант: дефект найденный в требованиях к ПО исправлен сразу и не попал на релиз. Таким образом мы экономим время команды и деньги клиента. Во время исправленная ошибка стоит дешевле и минимум времени, на этом этапе возможно взаимодействие не всей команды, а (допустим) тестировщик и разработчик.  Второй вариант: дефект найден клиентом. Отсюда : затраты на дополнительное время, которое оплачивается (соответственно денежные расходы заказчика). Это собрания, совещания и т.д., уже приводит к взаимодействию в работе большего количество людей команды, а так же ущерб репутации, отказ клиентов от покупки продукта. Отсюда следует : раннее тестирование экономит деньги и время! |
|  |
| **ЗАДАНИЕ № 4** |
| **Здание 10 этажей и подвал.** |
| Применяем технику эквивалентное разбиение. Разбиваем этажи на классы: четные и нечетные. Парковку принимаем за четный этаж. Из этого следует, что нужно проверить любой этаж из нечетных и один из четных. На четный этаж лифт не должен идти. Но мы знаем что из всех четных этажей лифт идет на 2-ой этаж (исключение), ход лифта на него мы проверим отдельно. |
|  |
| **ЗАДАНИЕ № 5** |
| **Привести пример бага S2 и S3** |
| S2. Баг не влияет на критический функционал, но создает неудобства при пользовании, есть возможность альтернативного обхода. Интернет магазин, добавлен товар в корзину. Для покупки товара нужно зайти в корзину и нажать кнопку оплатить. Кнопка не работает. Но при повторном нажатии на товар, который уже добавлен в корзину, открывается форма для оплаты. Это препятствует функционированию продукта и усложняет удобство для пользователя, поэтому серьезность дефекта будет высокой. Но так как есть альтернатива, то баг можно рассматривать с низким приоритетом. |
|  |
| S3. Баг влияет на критический функционал или критические данные. Дефекты которые нужно исправлять, но можно не сразу. Пример: приложение отчетности. Подаются квартальные и годовые отчеты. Возникает ошибка при подаче годового отчета - Это ошибка высокой степени серьезности, часть основной бизнес логики работает некорректно, но с низким приоритетом, так как на данный момент функция формирования годовой отчетности не актуальна. Этот баг может быть исправлен на следующем этапе. Такие ошибки обычно влияют на функциональность, наблюдаются только при использовании некоторых входных параметров. |
|  |
| **ЗАДАНИЕ № 6** |
| **Метрики** |
| Сайт: 30 задач (кейсов), 20 багов (3 нашли пользователи, 1 разраб.), 30 часов работы |
|  |
| Покрытие тестами: а=30 (общее кол-во тестов) / в=30 (общее кол-во требований) = 1 (\*100%) = 100% Если принять 30 задач за 30 выполненных кейсов, получаем 100 % покрытие тестами, что есть отлично, ничего менять и добавлять ненужно |
|  |
| Коэффициент ошибок пропущенных на продуктив: а=3 (баги найденные пользователями) / в=20 (общее количество багов) = 0,15 Полученный коэффициент больше чем 0,1. В нашем случае каждый десятый и даже больше баг не был обнаружен и был выпущен на продуктив. Качество тестирования очень низкое, необходимо пересмотреть работу, возможно нужно добавить кейсов или увеличить время работы. |
|  |
| Коэффициент повторно открытых дефектов: а=1 (кол-во повторно обнаруженных, предположим это баг который нашел разработчик) / в=20 (общее кол-во ошибок) = 0,05 Коэффициент удовлетворительный. Чем ближе к 0, тем меньше при разработке повторяются старые дефекты |
|  |
| Время затраченное командой на исправление дефектов: а=30 часов (общее время) / в=20 (кол-во ошибок) = 1,5 Исходя из этого мы видим, среднее время затраченное на исправление = 1,5 часа |