| 1. Дай визначення терміну “тестування”. |
| --- |
| Відповідь: Тестування це процес, який допомагає знайти помилки беги та дефекти в програмному забезпеченні, шляхом його ретельної перевірки вимог та реальних результатів. |
| 1. Назви принципи тестування, які ти знаєш. Опиши детально на вибір **два з них**. |
| Відповідь: Тестування показує дефект.  Кластеризація дефектів. Всі основні баги можуть знаходиться в одному блоці проекту.  Ефект пестицида, Тести потрібно змінювати, щоб знайти нові дефекти.  Вичерпне тестування неможливе. Відсутність помилок неможлива, вони є завжди. |
| 1. У чому різниця між QA та QC? |
| Відповідь: Основна різниця в принципах тестування. QA це правильність за нормами прогармування QC дотримання вимог. |
| 1. Перелічи основні етапи розробки програмного забезпечення у хронологічній послідовності. |
| Відповідь: Збір інформації планування вимог та документації розробка коду  тестування готового коду  реліз та обслуговування проекту |
| 1. Назви 4 особливості SCRUM методології. |
| Відповідь: Створення спринтів для роботи.  Щоденні зустрічі.  Стоврення робочих групу. Використовується беклог для створення планування роботи. |
| 1. Перелічи всі рівні тестування, які ти знаєш.   6.1 Дай коротке (1-2 речення) визначення **кожному** з них. |
| Відповідь: Модульне найперший етап тестування окремих блоків та модулів. Проводиться, щоб зайти помилки на ранніх етапах. Інтеграційне тестування різних модулів. Проводиться для розуміння чи вірно взаємодіють вони між собою. Системне тестування проводиться на готовому продукті перед завершення проекту. Для того, щоб побачити як система функціонує повністю, та виправити помилки. Приймальне тестування яке проводить замовник, для розуміння чи відповідає продукт вимогам. |
| 1. Назви відомі тобі види тестування.   7.1 Деталізуй (1-2 речення) визначення **двох** з них. |
| Відповідь: Функціональне, тестування придназначене для тестування функціоналу сайта. Тестування продуктивності, тестування на максимальне навантаження системи.  Тестування на безпеку. Допомагає зрозуміти наскільки захищена система.  Регресійне тестування. перевірка того, як система реагує на зміни. Нефункціональне тестування, тестування продуктивності продукта. |
| 1. Уяви, що ти щойно вніс/внесла зміни в файл *readme.md* в локальному репозиторії, який прив’язаний до віддаленого репозиторію. Напиши команди, які дозволять відправити ці зміни до віддаленого репозиторію. |
| Відповідь:  git add  git commit -m  git push |
| 1. Дай визначення терміну “баг”.   9.1 Назви основні етапи життєвого циклу багу з коротким (1-2 речення) описом **кожного** етапу. |
| Відповідь:  Баг - це збій програми, під час якого система поводитися не відповідно своїм цілям та задачам. Наприклад, при відкриванні поля користувача сайт фрізиться.  Життєвий цикл  Новий  Відкритий Закритий  може бути ще виправлення багу та його повторна перевірка. |
| 1. У формі для логіну на сайт (головна сторінка сайту), в якості логіну використовується номер телефону; про що повідомляється відповідним плейсхолдером. Існує юзер, який вже зареєстрований на цьому сайті. При логіні, зареєстрований користувач вводить в поле логіну латинські літери, потім свій актуальний пароль і після натискання кнопки “Увійти” отримує помилку про невірний логін або пароль.   Створи баг-репорт (якщо баг присутній) для описаного вище кейсу. |
| Відповідь:  Баг репорт не потрібен. В поле логін було введено не валідні дані. Тому юзер отримава помилку.  Назва. Перевірка валідних значень поля “Логін” під час входу в акаунт. . кроки: 1. На головній сторінці сайту відкрити поле входу в аккаунт.  2. В поле логін введіть тестові літери латинські 3. В поле введіть валідне значення очікуваний результат після натискання кнопки “Увійти” отримує помилку про невірний логін або пароль. фактичний результат після натискання кнопки “Увійти” отримує помилку про невірний логін або пароль. Все вірно так як, система повинна пропустити значення логіки тільки валідні Важливо 1-2-3 неважливо Северіти 3 приоріти 3 |
| 1. Напиши 1 позитивний тест-кейс (з усіма його атрибутами) для тестування функціоналу з попереднього завдання. |
| Відповідь: Назва. Вхід в систему під валідними даними. precondition .   Логін - 093 123-12-09 пароль - gfhjkm  Кроки: 1. На головній сторінці сайту відкрити поле входу в аккаунт.  2. В поле логін ведіть тестові значення логіну 093 123-12-09. 3. В поле пароль ведіть gfhjkm Очікуваний результат - Ви увійшли в систему |
| 1. Які техніки тест-дизайну ти знаєш? Наведи приклади використання (1-2 речення) **двом з них**. |
| Відповідь: Діаграма рішень. Показує які тести ми використали і які покрили, та як діяти далі.  Таблиця рішень. Таблиця показує вимоги до проекту та назви тестів. За допомогою цієї таблиці бачимо, які тести, що покривають. Дає змогу не робити лишню роботу. |
| 1. Уяви, що в тебе на проєкті є задача на тестування системи бонусів для користувачів. Ця система складається з кількох модулів: калькуляція бонусів, калькуляція знижок, система зберігання даних. Яку техніку або техніки естимації ти застосуєш для оцінки часу на тестування даної системи? Відповідь обґрунтуй (1-2 речень). |
| Відповідь: Щоб зрозуміти, як правильно рахує система я б створив таблицю рішень, щоб наші вимоги не пересікалися між собою і не сперечалися між собою. Так ми будемо розуміти які умови не повинні перетинатися, та які не можуть взаємодіяти разом. Наприклад для новий клієнт не може отримати скидку постійного клієнта. |
| 1. У базі даних є таблиці:  *Users1* з полями *Name, Age, Address Users2* з полями *Name, Phone, Number* Напиши запити, які повертатимуть такі дані:   А. всі дані з таблиці Users1;  Б. мінімальний вік користувачів;  В. адреси користувачів з іменем Nina;  Г. імена, адреси та номери телефонів усіх користувачів, вік яких більше за середній вік поміж всіх користувачів. |
| Відповідь: select \* from *Users1;*  select min( Age) from Users1;  select *Address from* Users1; where name = “Nina”:  select u1.Name, u1.Address u2.Phone, u1.Age From User1 u1  join User2 u2 on u1.Name = u2.Name  where u1.Age > (select avg(Age) from Users1) |
| 1. Опиши, що таке API та в яких випадках ми його тестуємо. |
| Відповідь: API - це частина програмного протоколу, яку ми можемо додати на наш проект, Яка виконує функції проекту з якого ми додали АРІ. Наприклад посилання на гугл мапс, або форма реєстрації через гугл почту. Ми тестуємо API на та як воно взаємодіє з системою нашого проекта.  Перевірку правильності обробки та передачі інформації API тестування може включати тестування функцій, методів, протоколів. |
| 1. Опиши, що таке JSON та його структуру. |
| Відповідь: JSON- це мова програмування, яка застосовується для передачі даних та створення запитів. Має просту структуру яка складається з двох частин. Грубо кажучи назва та елементи. Наприклад  “name“: Ben” “Age” : 24  Також вона может групуватися в більш складні запити і більш детально описувати об'єкт. |
| 1. З чого складається запит до API? Опиши **кожен** елемент. |
| Відповідь: Із запиту до серверу тобто get, delete, puch …. Також з URI посилання на сервер, на який ми надсилаємо запити Тіло запиту. В Боді ми можемо розписати за допомогою мови програмування . . |
| 1. Назви та коротко опиши основні методи запитів до API. |
| Відповідь: Це основні запити для створення запитів. Подивимся на принципі звичайного юзера. Post- створення юзера  GET пошук юзера  Put - редагування юзера Delete- видалення юзера Patch - внесення деяких змін шляхом завантаження нового файлу. Як типу пач для ігор) |
| 1. Тестувальник має в роботі мобільний застосунок. Точно відомо, що в сотій версії застосунку є баг. Тестувальник має доступ до всіх версій, починаючи з 1 і єдине, що може зробити — встановлювати їх на тестовий девайс. Як знайти версію, в якій цей баг вперше з'явився? |
| Відповідь: 1. перевію документацію та коментарі інших працівників.  Якщо у нас є можливість завантажувати всі системи. Я б спочатку провів тетс в 51 версії. Якщо його там не буде бага, значить проведу тест в 76 версії.  Якщо він там буде, баг буде в 76 версії. Тоді я розумію, що в проміжку між 51 та 76 версіями вперше виник баг. Значит беремо середнє між ними значення 64. І таким чином зменшують поле ймовірних випадків до мінімуму. Поки не знайдемо версію з якої все почалось.  Але ліпше вести документацію. |
| 1. Питання з зірочкою:   Які навички треба QA для того, щоб побудувати процес забезпечення якості на проєкті з 4 людей — QA, програміст, ПМ та дизайнер? Проєкт, що розробляється — це сайт з однією сторінкою, де знаходиться форма для опитування. |
| Відповідь: QA повинен володіти такими навичками:  1. Знання розробки ПЗ. 2. Вміти тестувати пз та вииктористувати вірну техніку тестування і слідкувати принципам тестування. 3.Потрібно вірно вести документацію. Вірно заповнювати чек лист, тест кейси та багрепорти. Хоч і сайт на одун сторінку, але вони потрібні. 4. Також допоможуть навички роботи з інструментами тестування, що провести навантажувальне тестування, перевірити як відправляються запити на сервіс. Так, як у нас є форма заповнення поля з даними і вона кудась відправляється та зберігається. 5. Зання бізнес ідеї, данного проєкту, що вірно побудувати тестування. |