**Домашнє завдання з пройденої теми:**

**Урок №6 Рівні та види тестування**

**1.Виконання завдання першого рівня**

**1.1** Складемо порівняльну таблицю функціонального, нефункціонального і пов'язаного зі змінами тестування.

| **Порівняння видів тестування** | **Функціональні види тестування** | **Нефункціональні види тестування** | **Пов'язане зі змінами види тестування** |
| --- | --- | --- | --- |
| **коли застосовується** | Функціональне тестування - це вид тестування в якому проводиться визначення, наскільки програмне забезпечення чи система відповідають заданим функціональним вимогам, описаним у специфікаціях. | Нефункціональне тестування - це вид тестування, направлений на перевірку нефункціональних особливостей програми, таких як зручність використання, сумісність, продуктивність. | Пов'язане зі змінами тестування - це такий вид тестування який запускається після проведення змін в програмі, таких як виправлення багу або дефекту. Програмне забезпечення проходить обов'язкову перевірку тестами для підтвердження того, що проблема була дійсно вирішена. |
| **що перевіряється** | Функціональне тестування виконує такі перевірки як:  -функціональні тести - грунтуються на функціях виконуваних системою і можуть проводитися на всіх рівнях тестування.  Тестування функціональності може проводитись в двох напрямках це вимоги та бізнес-процеси.  Потрібно скласти перелік того що буде піддаватися тестуванню.  -тестування безпеки - загальна стратегія перевірки системи безпеки яка грунтується на трьох основних принципів: конфіденційність, цілісність, доступність.  -тестування взаємовідносин - це функціональне тестування, що перевіряє здатність програми взаємодіяти з одним і більше компонентами або системами. | Нефункціональне тестування виконує такі перевірки як:  -Тестування продуктивності - дослідження показників швидкості реакції програми на зовнішній вплив при різної по характеру та інтенсивності навантажень. Розрізняють такі види:  -Навантажувальне тестування - дослідження здатності додатку зберігати задані показники якості при навантаженні в допустимих межах і деякому перевищенні цих меж;  -Тестування масштабованості - це дослідження здатності програми збільшувати показники продуктивності відповідно до збільшення кількості доданих нових ресурсів;  -Об'ємне тестування - дослідження продуктивності програми при обробці різних обсягів даних;  -Стресове тестування - дослідження поведінки програми при позаштатних змінах навантаження, що значно перевищують розрахунковий рівень;  -Конкурентне тестування - дослідження поведінки додатка в ситуації, коли йому доводиться обробляти велику кількість одночасно поступаючих запитів, що викликає конкуренцію між запитами за ресурси.  -Тестування встановлення - це тестування спрямоване на перевірку успішної інсталяції та налаштування, а також оновлення або видалення програмного забезпечення.  -Тестування зручності використання - тестування, спрямоване на дослідження того, наскільки кінцевому користувачеві зрозуміло, як працювати з продуктом, а також на те, наскільки йому подобається використовувати продукт.  -Тестування на відмову та відновлення - це дуже важливе тестування для систем, що працюють цілодобово.  -Конфігураційне тестування - це таке тестування яке спрямоване на перевірку роботи програмного забезпечення при різних конфігураціях системи. | Пов’язане зі змінами види тестування виконує такі перевірки як:  -Димове тестування - розглядається як короткий цикл тестів, що виконується для підтвердження того, що, після складання коду (нового або виправленого), додаток, що встановлюється, стартує і виконує основні функції.  -Регресійне тестування - спрямоване на перевірку змін, зроблених у додатку або навколишньому середовищі (лагодження дефекту, злиття коду, міграція на іншу операційну систему, базу даних, веб-сервер або сервер програми), для підтвердження того факту, що існуюча раніше функціональність працює як і раніше. Регресійними можуть бути як функціональні так і нефункціональні тести.  -Тестування збірки - це тестування, яке спрямоване визначення відповідності випущеної версії критеріям якості початку тестування.  -Санітарне тестування або перевірка узгодженості/справності - це тестування, яке достатнє для доказу того, що конкретна функція працює відповідно до заявлених у специфікації вимог. |
| **обмеження** | В функціональних тестах використовуються обмеження по:  *-Тестування вимог* використовує специфікацію функціональних вимог до системи як основу для дизайну створення Test-Cases.  *-Тестування* *бізнес-процесів* використовує їх знання, які описують сценарії щоденного використання системи.  Загальне тестування безпеки ґрунтується на трьох принципах:  -Конфіденційність – це приховування певних ресурсів інформації;  -цілісність, яка розділяється на два критерія довіру і пошкодження та відновлення;  -доступність – пред’являють вимоги о том, що ресурси повинні бути доступні авторизованим користувачам.  Тестування взаємовідносин – перевіряє здатність програми взаємодіяти з одним і більше компонентами | Тестування зручності використання ґрунтується на таких принципах:  -Правильність - це скільки помилок зробив користувач під час роботи з продуктом;  -Активізація в пам'яті – як багато користувачів пам'ятає роботу програми після припинення роботи з ним на тривалий період часу;  -Емоційна реакція – як користувач почуває себе після завершення завдання.  -Конфігураційне тестування має такі напрямки:  -проект із профілювання роботи системи - визначити оптимальну конфігурацію обладнання, що забезпечує необхідні характеристики продуктивності та часу реакції системи, що тестується;  -проект міграції системи з однієї платформи на іншу - перевірити об'єкт тестування на сумісність із оголошеним у специфікації обладнанням, операційними системами та програмними продуктами. | -Регресійне тестування має три основних типу:  -Регресія багів - це спроба довести, що виправлена помилка насправді невиправлена;  -Регресія старих багів - це спроба довести, що зміна коду чи даних зламало виправлення старих помилок, тобто старі баги почали знову відтворюватись;  -Регресія побічного ефекту - спроба довести, що нещодавня зміна коду або даних зламала інші частини програми, що розробляються. |
| **особливості** |  | Тестування зручності використання - може здійснюватися на різних рівнях розробки програмного забезпечення: модульному, інтеграційному, системному, приймайному.  -Тестування на відмову та відновлення - методика тестування полягає у симулюванні різних умов збою (наприклад: раптове відключення електропостачання на комп'ютері, відмова носіїв даних) та подальшому вивченні та оцінці реакції захисних систем. У процесі подібних перевірок з'ясовується, чи було досягнуто необхідного ступеня відновлення системи після виникнення збою. |  |

**1.2** Пояснення, в чому різниця між регресією та ретестингом.

Мета регресійного тестування - переконатися, що свіжі зміни в коді не надали побічних ефектів на функціональність, котра була вже розроблена, натомість виконання Ре-тесту перевіряє що дефект насправді виправлений.

Регресією підтверджується, що свіжі зміни в коді або додатку в цілому не вплинули на вже існуючу функціональність чи продукт в цілому, Ре-тест перевіряє ще раз і підтверджує факт того, що раніше не пройдені тест-кейси проходять після того, як дефекти виправлені.

Ре-тест проводиться на виправленій збірці з використанням тих самих даних, на тому ж оточенні, але з різним набором вхідних даних в тому випадку як регресія виконується за будь-якої модифікації або змін у вже існуючому проекті.

Ре-тестінг перевіряє той факт що дефект усунений, а регресійне тестування не проводить усунення дефектів.

Тест-кейси регресійного тестування можуть бути отримані з функціональних вимог або специфікацій, мануалів, і проводяться незалежно від того, що виправили розробники в Ре-тестінгу використовується той же тест-кейс, який виявив дефект.

**2.Виконання завдання другого рівня**

2.1 Як ти вважаєш, чи можливе для продукту проведення тільки функціонального тестування, без перевірки нефункціональних вимог ?

На мою думку для випуску якісного програмного продукту виконання тільки функціонального тестування без перевірки вимог нефункціонального тестування неможливе. Тому що випускаючи програму неможливо оцінити якість її роботи тільки виходячи з вимог вказаних в специфікації. В край необхідно провести такі тести як тестування продуктивності, Тестування зручності і використання, тестування на відмову та відновлення та інше. Якщо наприклад виходить програмний продукт який дуже навантажує внутрішні ресурси вашого комп'ютера то він просто не зможе нормально функціонувати так як не адаптований (під ресурси, платформу використання так і припустимо збільшення об'єму інформації) і тому вкрай необхідно виконувати тестування продуктивності. Стосовно тестування зручності як на мою думку виходить другим важливим фактором після тестування продуктивності. Так як припустимо не в тому місці розташовуються кнопки, не працює належним чином адаптивність під різні пристрою чи в котрий раз потрібно заходити в багато випадаючих списків щоб застосувати якусь функцію, а не вивести її на панель швидкого доступу. Такий вид як тестування на відмову та відновлення необхідний припустимо для того чи має програмне забезпечення автозбереження в певний проміжок часу, чи коректно запускається вона після непередбачуваних дій (відключення електроенергії), чи має функцію відновлення попередніх файлів та всіх застосованих функцій.

2.2 Як ти розумієш необхідність проведення smoke (димового) тестування? Чи завжди воно є доречним?

Димове тестування зазвичай використовується тільки в тих випадках коли необхідно максимально швидко дати загальні відповіді стосовно роботи програмного забезпечення. Воно є доречним коли на перевірку відсутній час на більш деталізоване тестування. Припустимо при встановленні программа корректно аналізує здатність платформи та операційної системи, не видає помилку під час встановлення, всі внутрішні критично важливі модулі працюють при застосуванні, відсутнє блокування програми при застосуванні будь якого параметру.