6.1 Сценарне тестування

Сайт: ІТ курси

Курс: QA manual v.5.0/2024 Книга: 6.1 Сценарне тестування Надруковано: Мусятовська Олена

Дата: субота 5 квітня 2025 02:19

Зміст

Scenario Testing

Scenario Testing: Examples

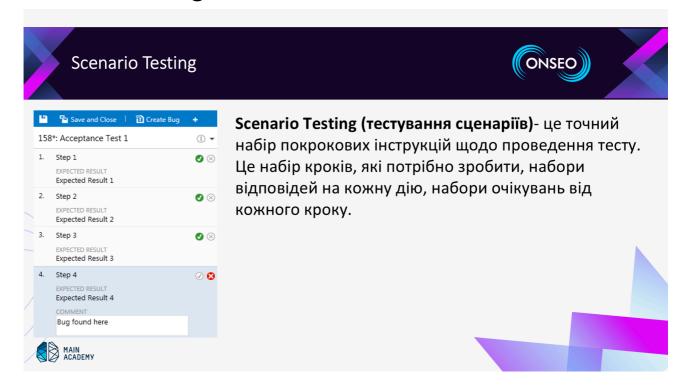
Переваги Scenario Testing

Протирічність Scenario Testing

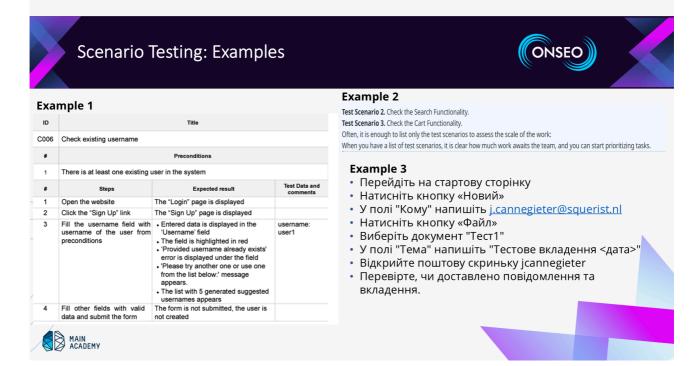
Scenario Testing: Недоліки

Підсумок

Scenario Testing



Scenario Testing: Examples



Переваги Scenario Testing

Переваги Scenario Testing



Процес створення сценаріїв забезпечує отримати кілька ключових переваг:

- Ретельно продуманий дизайн кожного тесту, його оптимізація
- Сценарії тестування можуть бути схвалені різними зацікавленими сторонами, такими як бізнес-аналітики, розробники, клієнти, щоб забезпечити ретельне тестування програми, що тестується. Це гарантує, що програмне забезпечення працює в найпоширеніших випадках використання.
- Можливість багаторазового використання
- Можливість використовувати сценарії, а точніше тест кейси що в ходять в сценарії як метрику щодо відсотків виконаних цих тестів.

Одна з найбільших **переваг** сценаріїв тестування - це **точність**. Але водночас час це найбільша **слабкість**.



Протирічність Scenario Testing

Протирічність Scenario Testing



Під час виконання сценарію ми звертаємо увагу на щось згадане і **не звертаємо** уваги на інші моменти. А зі сценарієм ви пропускаєте те саме щоразу, коли його запускаєте.

Це схоже на виконання тестування суто по інструкції росписаній в сценарії. Ви йдете чітко за описаним алгоритмом не відхиляючись від описаного. Тому, з часом починає спрацьовувати один з принципів тестування.

↑ тестування.

Який? =)



Scenario Testing: Недоліки

Scenario Testing: Недоліки



- Налагодження тестового сценарію є серйозною проблемою. Якщо в тестовому сценарії є помилка, іноді це може призвести до смертельних наслідків.
- Тестове обслуговування є дорогим. Навіть якщо в графічному інтерфейсі відбулися незначні зміни, тестовий сценарій потрібно перезаписати або замінити новим тестовим сценарієм.
- Типова ситуація, коли у нас є дизайнер тестів, який рідко виконує тести, але розробляє тести (з вимог,без досвіду тестування цього пз) для іншого тестувальника, який насправді їх виконує. Це дуже рентабельно. Але що, якщо програма має несподівані помилки?!



Підсумок

