Тестування програмного забезпечення

Викладач: к.т.н. Шитікова Олена Вікторівна

Лекція 8
ФУНКЦІОНАЛЬНЕ ТЕСТУВАННЯ

Функціональне тестування

Функціональне тестування (functional testing) – вид тестування, при якому виявляється некоректна або неправильна робота функціоналу програми.

Тобто це процес перевірки відповідності поведінки системи тим функціональним вимогам, що були заявлені спочатку.

Функціональні вимоги визначають, що саме робить програмне забезпечення, які завдання вирішує.

Функціональне тестування

При ***функціональному тестуванні*** перевіряється коректність роботи системи, яка включає:

- перевірку кожної з функцій програми,
- адекватність збережених і вихідних даних,
- методи їх обробки,
- обробка даних, що вводяться,
- методи зберігання даних,
- методи імпорту та експорту даних тощо

Функції описуються в вимогах, функціональних специфікаціях або у вигляді випадків використання системи.

Функціональне тестування

Функціональне тестування займає 90% часу від усього тестування і може займати до 80% всього бюджету проєкту з тестування. Перевіряє функціональні вимоги: логіку і бізнес-правила додатків або системи.

Функціональні вимоги включають:

- функціональну придатність (suitability);
- точність (accuracy);
- здатність до взаємодії (interoperability);
- відповідність стандартам і правилам (compliance);
- захищеність (security).

Функціональне тестування

Цілями процесу функціонального тестування в області якості є:

• підтвердження якості програмного продукту, що відповідає показникам, встановлених замовником;

• виявлення та документування дефектів програмного продукту;

• прийняття об’єктивного рішення, зафіксованого в звіті про результати тестування, про можливість поставки програмного продукту замовнику.

Функціональне тестування

Зазвичай, функціональне тестування проводиться на двох рівнях:

- ***компонентне (модульне) тестування*** – вид тестування окремих компонентів програмного продукту, сфокусоване на їх специфіці, призначенні та функціональних особливостях;
- ***інтеграційне тестування*** – вид тестування, що проводиться після компонентного тестування і спрямований на виявлення дефектів взаємодії різних підсистем на рівні потоків управління та обміну даними.

Функціональне тестування

Підходи функціонального тестування:

- тестування вебформ;
- пошук функціональних багів;
- техніки тестування і еквівалентне розбиття;
- граничні сценарії;
- чекліст для перевірки функціоналу сайту;
- доповнення чекліста для перевірки функціоналу сайту;
- тестування без вимог;
- неформальні техніки тестування.

Тестування за ознакою позитивності сценаріїв

Позитивний тестовий сценарій (positive testing scenarios) використовує тільки коректні дані і перевіряє, що додаток правильно виконав функцію, тобто покликаний показати, що програма працює так, як треба, за умови, що користувач вносить валідні дані і не виходить за рамки передбаченого сценарієм поведінки.

Основною метою позитивного тестування є перевірка здатності системи працювати з виконанням тих функцій, для яких розроблялася.

Тестування за ознакою позитивності сценаріїв

Негативний тестовий сценарій (negative testing scenarios) призначений для перевірки сценаріїв, відмінних від коректного, вхідних даних з одним або більше невалідних параметрів

Основною метою негативного тестування є перевірка стійкості системи до впливів різного роду, валідації невірного набору даних, перевірка обробки виняткових ситуацій (як в реалізації самих програмних алгоритмів, так і логікою бізнес-правил).

Тестування за ознакою позитивності сценаріїв

В рамках негативного і позитивного тестування виконують ***димове тестування***, яке передбачає проведення коротких стадій тестів, які підтверджують, що система в цілому працездатна і виконує ключові функції.

У випадку, коли в ході димового тестування ніяких очевидних дефектів не виявляють, тест оголошується успішно пройденим.

Тестування за ознакою позитивності сценаріїв

План тестування:

1. Вивчення документації.

2. Димове тестування (smoke testing) – перший прогін програми, щоб зрозуміти чи працює вона взагалі.

3. Позитивне тестування – перевірка роботи програми при отриманні валідних вхідних даних.

4. Негативне тестування

Тестування за ознакою позитивності сценаріїв

Найбільш поширені негативні тест-кейси:

- обов’язкове введення (Required Data Entry);
- типи даних полів (Field Type Test);
- розмір поля (Field Size Test);
- числові граничні значення (Numeric Bounds Test);
- граничні значення дати (Date Bounds Test);
- валідність дати (Date Validity);
- правила заповнення полів (Rules for filling fields);
- внутрішні одинарні лапки (Embedded Single Quote)

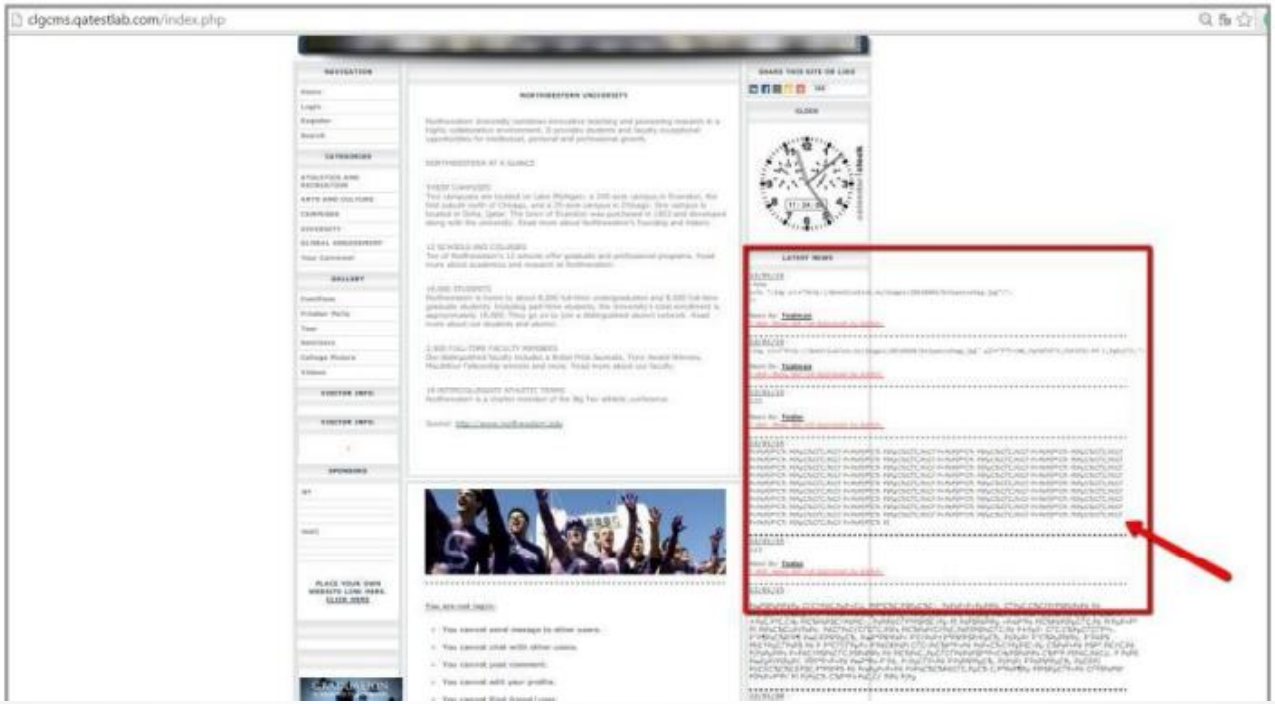
Перевірка модулів сайту

1. Реєстрація (registration, logging) – модуль дозволяє користувачам реєструватися на сайті для реалізації деяких подальших дій (наприклад, для доставки товару, відправки інформаційної розсилки, надання доступу до закритої інформації).

2. Авторизація (authorization) – підтвердження особи користувача в мережі. Для авторизації необхідно ввести логін та пароль.

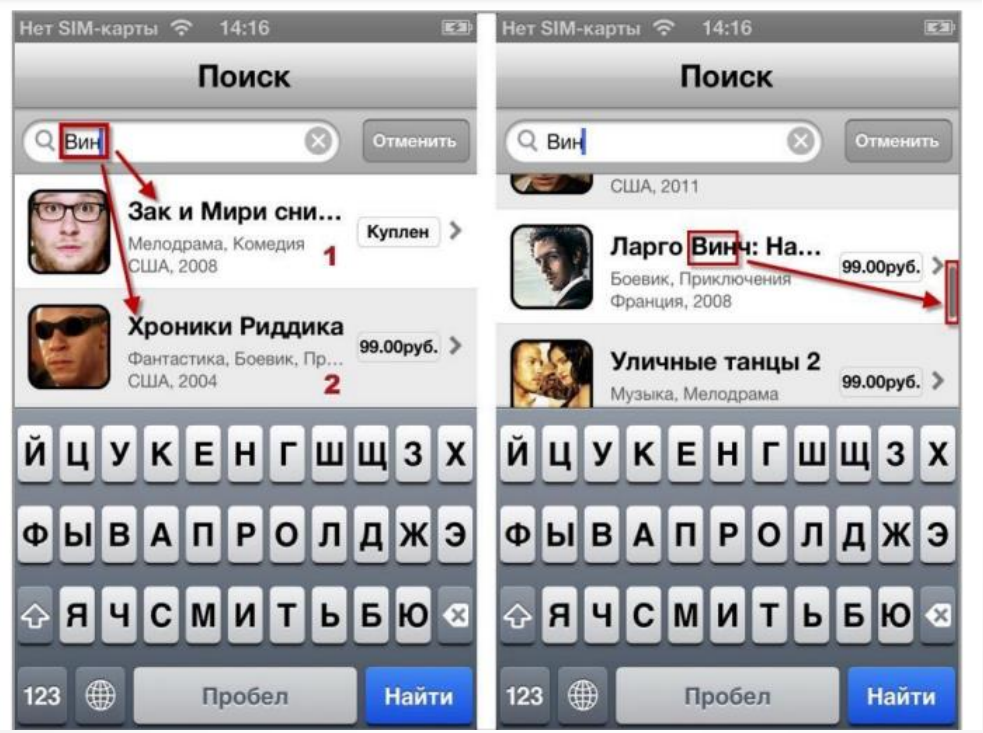
Перевірка модулів сайту

3. Новинний модуль (news) – призначений для публікації новин на сайті, забезпечує спрощення роботи як адміністраторів сайту по додаванню новин на сайт, так й засвоюваність інформації відвідувачами.



Перевірка модулів сайту

4. Пошук по сайту (search) – модуль дозволяє здійснювати пошук на всіх сторінках сайту. Результатом роботи є видача списку знайдених сторінок з уривками текстів та посиланнями, що містять пошуковий запит.



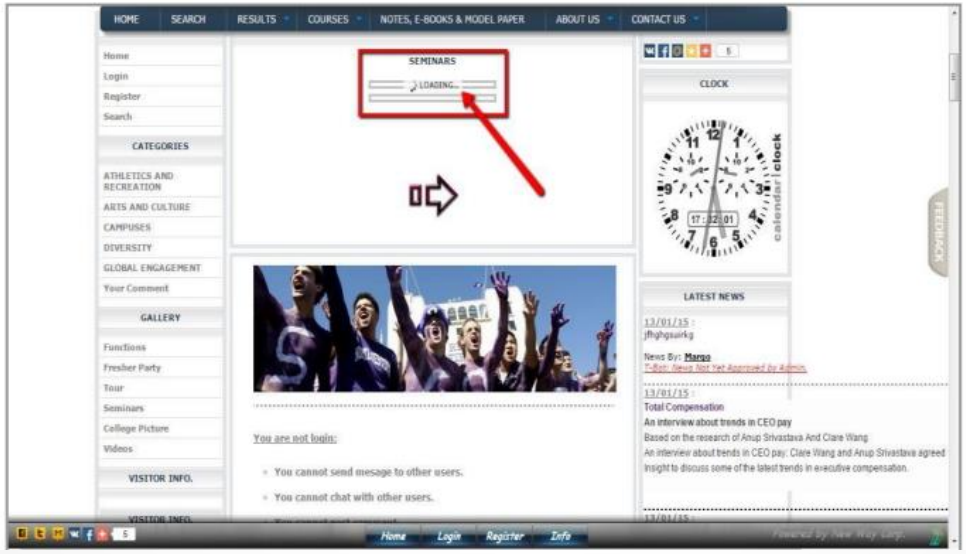
Перевірка модулів сайту

5. Зворотній зв'язок (feedback) – модуль дозволяє вивести текстову інформацію та форму з полями: ім'я, e-mail, повідомлення. Після натискання на кнопку «Відправити» введена інформація відправляється на e-mail користувача.

6. Банер (banner) – графічне зображення рекламного характеру, що дозволяє на сторінках сайту в певних місцях створити рекламні місця. Банери розміщують для залучення потенційних клієнтів або для формування іміджу.

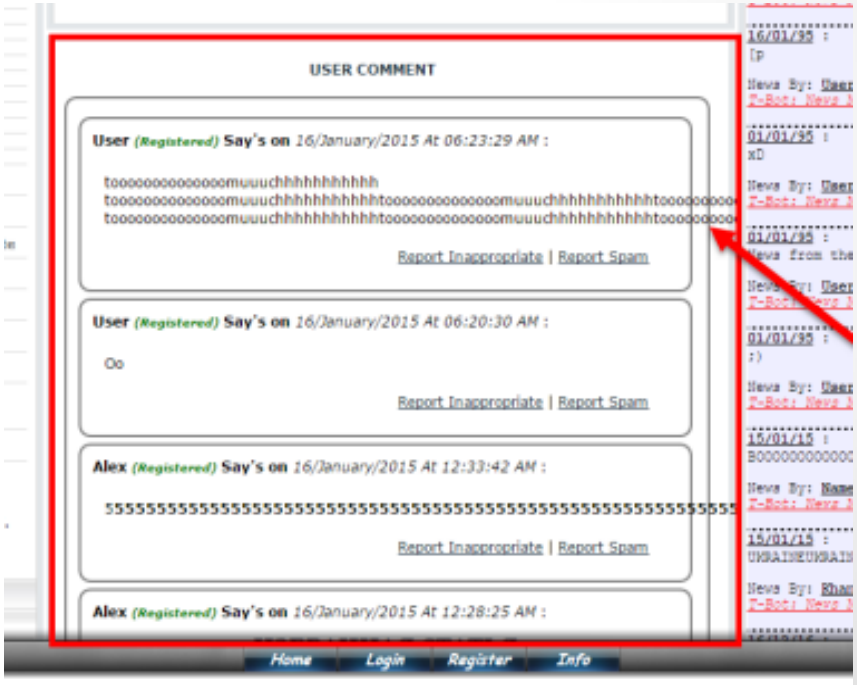
Перевірка модулів сайту

7. Фотоальбоми (photo album) – модуль, призначений для створення каталогу зображень, який дозволяє створювати зручні каталоги з категоріями зображень, фільтрами та іншими засобами, які спрощують навігацію і роботу з фотоальбомами.



Перевірка модулів сайту

8. Форум (forum) – важливий модуль, що дозволяє створювати форуми, відрізняється широкими функціональними можливостями і можливістю тонкої настройки, може використовуватися як для створення повноцінного форуму на сайті, так і для створення окремого сайту-форуму. Форум дозволяє створювати теми та вести їх обговорення.



Перевірка модулів сайту

9. Інтернет-магазин (online shop, e-shop) – один з найважливіших модулів системи, необхідний для створення повноцінного інтернет-магазину, володіє широкими настройками та відрізняється гнучким настроюванням.

10. Випадаючий список (drop-down list) – елемент списку одиночного вибору.

11. Коментарі в соціальних мережах – модуль для автоматичної публікації контенту, опублікованого на сайті, в соціальних мережах.

Перевірка модулів сайту

12. Відео – модуль, що дозволяє розміщувати на сайті власні відеофайли або розміщувати переналаштовані ліцензійні відео та канали.

13. Модуль розсилки (mailout module) – модуль розсилок дозволяє організувати розсилання повідомлень зареєстрованим відвідувачам сайту, існує можливість автоматичної відправки тематичних повідомлень на електронну пошту.

14. Форма (form) – набір полів для введення даних на вебсторінці. Дані відправляються на сервер, коли користувач завершує заповнення всіх полів і відправляє форму.

Валідація даних

Валідація (validation) – це процес перевірки даних на відповідність певним, заздалегідь відомим правилам (форматам, вимогам).

При тестуванні необхідно перевіряти узгодженість валідаторів вхідних даних з логікою обробки цих даних програмою, для тестування затвердження даних, треба розуміти, як вони повинні працювати, що можна вважати правильним, а що ні.

Невалідні дані, що не задовольняють певним обмеженням, можуть викликати збій у роботі програми.

Валідація даних

Проблеми, що виникають при відсутності валідації:

1. Неможливість відновитися після збою. Не завжди програма здатна «повернути все назад». Можливо, в процесі роботи програма виконала якісь незворотні дії – видалила файл, відправила дані по мережі, надрукувала щось на принтері та інше. Але навіть якщо відновлення в принципі можливо, алгоритм відновлення може теж містити помилки, і це іноді призводить до зовсім сумних наслідків.

Валідація даних

2. Додаткове навантаження на систему.

Відновлення після збою – це зайва робота. Вся робота, яка була виконана до моменту збою – теж зайва. А це означає додаткове навантаження на систему, якої можна уникнути, якщо заздалегідь перевірити дані.

З іншого боку, валідація - це теж додаткове навантаження, причому відновлення доводиться робити лише зрідка, а перевірку треба виконувати кожен раз.

Валідація даних

3. Ін’єкції не викликають збоїв.

Одним з основних способів експлуатації вразливостей в програмах полягає в тому, щоб обійти валідатори, тобто передати дані, які валідатор визнає коректними. Але при цьому ці дані інтерпретуються неналежним чином та зловмисник може отримати несанкціонований доступ до даних або деякими можливостям програми. Якщо валідації немає взагалі, завдання зловмисника максимально спрощується.

Валідація даних

4. Складність ідентифікації, причини, проблеми.

Якщо виключення вилетіло звідкись із глибини програми, визначити причини його виникнення не просто. І навіть якщо це можливо, може виявитися нелегко пояснити користувачеві, що збій викликано даними, які він ввів деякий час тому в зовсім іншому місці програми. Та якщо перевірка виконана одразу ж після введення даних, жодних складнощів з ідентифікацією джерела проблеми не виникає.

Валідація даних

Валідація даних здійснюється наступним чином:

1. Посимвольна перевірка. Як правило, такі перевірки виконуються в інтерфейсі користувача під час введення даних.

2. Перевірка окремих значень. Для інтерфейсу користувача це перевірка значення в окремому полі, виконуватися вона може як під час введення даних (перевіряється неповне значення, яке введено до справжнього моменту), так і після завершення введення, коли поле втрачає фокус.

Валідація даних

3. Сукупність вхідних значень. Можна припустити, що в програму спочатку передаються деякі дані, після чого подається сигнал, який ініціює їх обробку. У цей момент можна виконати так звані «семантичні» перевірки, націлені на валідацію не тільки окремих значень, але і взаємозв'язків між ними, взаємних обмежень.

Для програмного інтерфейсу цей різновид валідації передбачає перевірку набору вхідних параметрів процедури, які викликаються. Для випадку отримання даних з файлу – це перевірка всіх прочитаних даних.

Валідація даних

4. Перевірка стану системи після обробки даних. Якщо валідацію безпосередньо вхідних даних виконати не вдається – можна спробувати їх обробити, але залишити можливість повернути все до вихідного стану. Такий механізм часто називається транзакційним.

Транзакція – це послідовність дій, які або всі завершуються успішно, або відбувається якийсь збій при виконанні окремої дії, і тоді скасовуються результати всіх попередніх дій цього ланцюжка.

Приклад функціонального чекліста

Шапка

- *Іконка і назва на вкладці*
- *Логотип*
- *Live Chat*
- *Twitter*
- *Facebook*
- *Like*

Фотогалереї

- *Перелистування фото*
- *Коректний зум (якщо є)*
- *Коректний поворот фотографій (якщо є)*
- *Коректне відображення списку фотографій і самих фото*

Реєстрація/Авторизація

- *Реєстрація користувача*
- *Авторизація користувача*
- *Анонімний користувач*
- *Відновлення пароля*
- *Редагування облікового запису*

Статті

- *Шаринг статей у всіх запропонованих соц. мережах*
- *Відображення фото*
- *Коректність переходу за посиланнями*
- *Коректність повернення на головну сторінку*

Приклад функціонального чекліста

Особистий кабінет

- *Редагування анкети*
- *Можливість видалення анкети*
- *Відображення статуса користувача*
- *Відображення статуса підписки*
- *Можливість відмінити/підтвердити підписку*
- *Вихід користувача із особистого кабінету*

Пошук

- *Пошук за новинами, розділами*
- *Перехід за посиланнями*
- *Відображення дат новин*

Зворотній зв'язок

- *Робота при валідному заповненні полів*
- *Робота при невалідному заповненні полів*