

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА  
Факультет прикладної математики та інформатики  
Кафедра програмування



**ЛАБОРАТОРНА РОБОТА № 6**

Виконала:  
студентка групи ПМОм-11  
Кравець Ольга

Перевірив:  
Яцик І. М.

Львів 2024

**Тема:** Тестування ергономіки програмного забезпечення.

**Мета:** Ознайомитись з методами тестування ергономіки, порівняти різні методи між собою.

### **Хід роботи**

#### **Завдання 1. Переваги і недоліки функціонально тестування:**

##### ***Переваги:***

##### *а) Імітація реального використання системи.*

Функціональне тестування імітує фактичні сценарії використання програмного забезпечення, що дозволяє перевірити чи система працює відповідно до вимог замовника і вирішує поставлені завдання. Це гарантує, що функціональність, заради якої створювалась програма, повністю реалізована та відповідає очікуванням користувача.

##### *б) Можливість перевірки на всіх рівнях (модульне, інтеграційне, системне).*

Функціональне тестування може бути реалізоване на різних рівнях - від окремих компонентів (модульне тестування), до перевірки взаємодії між різними модулями (інтеграційне тестування) і перевірки роботи всієї системи (системне тестування). Це дає змогу покроково аналізувати працездатність кожного окремого елемента та його взаємодію з іншими компонентами.

##### *с) Виявлення невідповідностей між вимогами та реалізацією.*

Оскільки функціональне тестування орієнтоване на перевірку відповідності ПЗ вимогам, воно дозволяє швидко виявляти невідповідності між очікуваною та фактичною поведінкою системи.

d) *Можливість тестування без знання внутрішньої реалізації.*

Функціональне тестування може виконуватись за принципом “чорної скриньки”, коли тестувальник не має доступу до внутрішнього коду або структури програми. Це дозволяє зосередитись на поведінці системи з точки зору користувача, що часто є ключовим для замовників.

e) *Важливість для кінцевого користувача.*

Функціональне тестування надає впевненість у тому, що користувачі зможуть виконувати основні завдання в програмі, що є критичним для загальної якості продукту.

### ***Недоліки:***

a) *Можливе пропущення логічних помилок.*

Функціональне тестування здебільшого перевіряє відповідність вимогам, але не гарантує, що логіка програми абсолютно правильна. Наприклад, деякі внутрішні логічні помилки або недосконалості алгоритмів можуть залишитися непоміченими, якщо вони не впливають на зовнішні функціональні вимоги.

b) *Можливість надмірного тестування.*

Оскільки функціональне тестування може проводитись на всіх рівнях, є ризик надмірного тестування або дублювання тестів на різних етапах, що збільшує час і ресурси, необхідні для проведення повної перевірки. Це може стати проблемою, особливо в проектах з обмеженими ресурсами.

c) *Обмежена оцінка продуктивності та інших нефункціональних аспектів.*

Функціональне тестування зосереджене лише на тому, що система робить, але не перевіряє як вона це робить з точки зору

продуктивності, стійкості до стресових умов, безпеки і т.д. Такі аспекти можуть залишатися неперевіреними і викликати проблеми при реальній експлуатації.

d) *Не враховує можливі крайні випадки використання.*

Функціональне тестування може бути зосереджене на основних сценаріях використання, залишаючи поза увагою незвичні або рідкісні випадки, які можуть викликати несправності або неочікувану поведінку програми.

e) *Залежність від якості документації.*

Якість функціонального тестування залежить від чіткості та повноти вимог. Якщо вимоги до ПЗ не детально визначені або суперечливі, тестувальники можуть мати проблеми з розумінням того, що саме потрібно тестувати. Це може призвести до пропущених функцій або неправильних висновків про працездатність системи.

## **Завдання 2. Приклад виду функціонального тестування. Короткий опис його.**

*Тестування взаємодії* - вид тестування, націлений на оцінку якості взаємодії компонент-програмної системи або всієї програми з іншими компонентами або програмним забезпеченням.

Даний вид тестування став дуже актуальним із зростанням популярності мережевих додатків, основною запорукою успіху яких є швидка, надійна та безпечна взаємодія між іншим ПЗ в мережі.

Особливе значення тестування взаємодії має для таких областей мережевих технологій, як:

- ✓ Потоки даних (різні мультимедіа, мова, інформація).
- ✓ Вузькосмугові та широкосмугові мережі.
- ✓ Мобільні та фіксовані мережі.
- ✓ Мережі з тимчасовим поділом каналів та IP-мережі.

Суть даного виду тестування дозволяє включати в себе тестування сумісності та інтеграційне тестування, які також відповідальні за якість як взаємодії деяких компонентів в рамках системи, так і взаємодію ПЗ, що розробляється, з іншими додатками.

### **Завдання 3. Опис двох видів нефункціонального тестування.**

*Димове тестування (Smoke testing)* також є аналогом тестування складання (Build Verification Testing) - це тип тестування програмного забезпечення, який включає вичерпний набір тестів, спрямованих на перевірку роботи найбільш важливих, критичних функцій в системі. Результат цього тестування використовується для визначення того, чи досить стабільна збірка, щоб продовжити подальше тестування.

Димові тести здійснюються в тих випадках, коли тестувальники отримують нову версію (білд) програми на тестування, при цьому вважаючи її відносно нестабільною. На цьому етапі необхідно переконатися, що надзвичайно важливі функції AUT (Application Under Test) працюють згідно з очікуваннями.

Концепція цього виду тестування полягає в тому, щоб виявити серйозні проблеми якомога раніше і відхилити цей білд або повернути його на доопрацювання на ранньому етапі тестування. Це необхідно, щоб не поглиблюватися в складні

тести і не витрачати час на явно неякісне програмне забезпечення.

Метою димового тестування є перевірка надійності і стабільності програмного забезпечення загалом, щоб дозволити подальше проведення ретельнішого тестування.

Переваги димового тестування:

- ✓ відразу знаходяться проблеми інтеграції;
- ✓ тестування розкриває проблеми на ранній стадії;
- ✓ забезпечує певний рівень впевненості в тому, що зміни в програмному забезпеченні не зробили несприятливої дії на основні області системи;
- ✓ збій при димовому тестуванні призводить до негайної відмови від цієї збірки програмного забезпечення.

*Санітарне тестування* або перевірка працездатності (Sanity testing) - це дуже коротка поглиблена перевірка певної функціональності комп'ютерної програми, системи, веб-сайту, розрахунків і так далі, яка проводиться з метою переконатися, що система працює належним чином. Цей вид тестування часто виконується до початку повного циклу регресійного тестування, але після проведення димового тестування.

Санітарне тестування, як правило, проводиться, коли виправлена яка-небудь незначна помилка в системі або є невелика зміна у функціональності.

Після внесення змін до коду нова збірка програмного забезпечення стає доступною для тестувальників. Після встановлення збірки тестувальники можуть виконати швидко перевірку зміненої функціональності замість повного регресійного тестування, якому потрібно приділити більше часу.

Якщо виправлення помилки і зміни не працюють належним чином, тоді тестувальники не повинні приймати збірку і, відповідно, будь-які помилки, виявлені на такій ранній стадії, врятовують від втрати часу на тестування недопрацьованої збірки.

Декілька важливих особливостей санітарного тестування:

- є поверхневим тестуванням з концентрацією на детальному тестуванні деяких вибраних функцій;
- є підмножиною регресійного тестування;
- виконується тоді, коли у тестувальників відсутня достатня кількість часу для детального тестування;
- перевірка працездатності зазвичай не документується;
- є коротким, швидким тестуванням для того, щоб переконатися, що зміни працюють так, як очікувалося, і відповідно до технічних документів;
- тестування працездатності виконується для перевірки виправлення незначних помилок і незначних функціональних змін.

Переваги санітарного тестування:

- ✓ фокусується тільки на певних функціях програмного забезпечення;
- ✓ не виходить на рівень структури проекту, тому розробникам дуже важко зрозуміти, як виправити проблеми, виявлені під час перевірки працездатності;
- ✓ виконується тільки для деяких обмежених функцій, тому, якщо є які-небудь проблеми з іншими функціями, їх буде складно відловити;
- ✓ перевірка працездатності, як правило, не документується, тому ті ж самі перевірки в майбутньому можуть бути випущені.

### *Порівняльна таблиця димового і санітарного тестування*

<i>Smoke testing (Димове тестування)</i>	<i>Sanity testing (Перевірка працездатності, санітарне тестування)</i>
Виконується на початкових збірках програмного продукту перед регресійним тестуванням	Проводиться на збірках, які успішно пройшли димові випробування і перед циклом регресійного тестування
Тестована збірка може бути стабільною або нестабільною	Тестована збірка відносно стабільна
Мотивом є перевірка стабільності нової зібраної версії продукту в цілому	Основною метою є перевірка раціональності системи в деталях, щоб приступити до ретельного тестування
Збій при тестуванні призводить до негайної відмови від цієї збірки програмного забезпечення	Збій в перевірці працездатності поміщає збірку програмного забезпечення в список відхилених
Може виконуватися як розробниками, так і тестувальниками	Зазвичай проводиться тестувальниками
Включає документацію і роботу за сценарієм	Не робить акценту на якій-небудь документації і роботу за сценарієм
Можна розглядати як загальний вигляд тестування, що поверхнево покриває усі основні функції без глибокого тестування	Спеціалізована або детальніше орієнтована методика тестування, спрямована на певну функціональність або функцію
Може бути виконано як автотестами, так і вручну	Зазвичай виконується вручну
Економить зусилля команди тестувальників і мінімізує час роботи над дефектним складанням програмного забезпечення	Зберігає час в умовах, коли його недостатньо для максимально великого регресійного тестування



**Висновки:** під час виконання лабораторної роботи я ознайомилася з функціональним тестуванням та його переваги і недоліки. Функціональне тестування дозволяє перевірити відповідність системи вимогам користувачів, імітуючи реальні сценарії використання. Також розглянула методи нефункціонального тестування, зокрема димове та санітарне тестування. Ці методи допомагають виявляти серйозні проблеми на ранніх етапах, забезпечуючи стабільність системи перед більш детальним тестуванням. Загалом, я переконалася у важливості комбінації функціонального та нефункціонального тестування для досягнення високої якості програмного забезпечення. Виконувати цю лабораторну роботу було знаттєлюбно.

#### **Список використаних джерел:**

1. [Функціональне тестування](#)
2. [Різниця між функціональним і нефункціональним тестуванням](#)
3. [Види тестування ПО. Функціональне тестування \(Functional Testing\). Тестування безпеки \(Security and Access Control Testing\). Тестування взаємодії \(Interoperability Testing\)](#)
4. [Тестування взаємодії](#)
5. [Різниця між димовим та санітарним тестуванням](#)