**Лабораторна робота №2**. **Процес тестування програмного забезпечення. Приймальні випробування (формування плану приймальних випробувань).**

**Мета:** Навчитися формувати план приймальних випробувань ПЗ

**Методичні рекомендації до виконання лабораторної роботи.**

1. Опрацювати теоретичні відомості.
2. Відповідно до проведеного в Практичній роботі №3 визначення вимог щодо власного проекту сформувати план приймальних випробувань ПЗ як послідовний перелік функціональних та не функціональних властивостей за ТЗ, що підлягають перевірці на приймальному тестуванні, та видів тестування щодо цих властивостей. Для цього заповнити таблицю .

|  |  |  |  |
| --- | --- | --- | --- |
| №№ пп | Властивість | Тип | Вид тестування |
| 1 | 2 | 3 | 4 |
|  |  |  |  |
|  |  |  |  |

Графа 3 містить ознаку типу властивості: Ф - функціональна, Н -нефункціональна.

Повинно бути **не менше** одної 3 функціональних та 6 нефункціональних властивостей для тестування.

1. Робота повинна бути виконана згідно критеріїв оформлення документації та повинна містити

* Назва лабораторної роботи.
* Прізвище, група
* Назва проекту.
* Результати роботи оформлюються у вигляді таблиці:

По закінченню практичну роботу потрібно здати на перевірку викладачеві, надіславши електронною поштою на адресу [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com) . Якщо викладач знаходить помилки чи неточності, він може повернути роботу на доопрацювання.

Файл з роботою повинен мати назву в такому форматі:

**ОPI<Номер групи><Номер лекції / практичної / лабораторної>[-<Номер завдання>][літера позначення типу роботи L – лекція, P –практична, R – лабораторна]<Прізвище англійською>**.. Наприклад, **РІ4101Р**buts.doc.

Не копіюйте фрагментів з різних інформаційних джерел, подумайте і викладіть свою точку зору. При наявності робіт-"близнюків" відповідь буде зараховуватися першому за часом надсилання.

Тему в заголовку листа записати

**ОПІ <Номер групи>-ЛР<Номер лабораторної>-<Прізвище >**

**Строк виконання цієї роботи ІПЗ-41 – 27.09.2021**

**ІПЗ-42 – 02.10.2021**

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

Тестування програмного забезпечення виконує дві базові функції: верифікацію та атестацію. **Верифікація** забезпечує відповідність результатів конкретної фази процесу розробки вимог даної та попередньої стадії. **Атестація** є гарантією того, що програмний продукт задовольняє системним вимогам.

Тестування - це одна з технік контролю якості, що включає в себе діяльність з планування робіт (Test Management), проектуванню тестів (Test Design), виконанню тестування (Test Execution) і аналізу отриманих результатів (Test Analysis).

Необхідними умовами для тестування є наявність :

• об'єкта тестування, доступного для проведення іспитів;

• виконавця(ів) (залежно від виду проведених іспитів їм може бути як людина, так і машина або комбінація людина + машина).

Достатніми умовами для тестування є наявність:

• об'єкта тестування, доступного для проведення іспитів;

• виконавця(ів) (залежно від виду діяльності на різних фазах їм може бути як людина, так і машина або комбінація людина + машина);

• плану тестування;

• тест кейсів / тестів;

• звіту, що підтверджує виконання задач і досягнення цілей, по тестуванню об'єкта.

Програмний продукт є якісним, коли:

* під час роботи користувача з програмним продуктом виникає невелика кількість відмов;
* програмний продукт надійний, а це означає, що його використання рідко викликало аварійні відмови;
* програмний продукт задовольняє вимогам більшості користувачів.
* Під плануванням випробувань розуміють об'єм випробувань, підходів, ресурсів та розкладів виконання запланованих дій.

Тестування проводиться для посвідчення того, що програмне забезпечення виконує те, що від нього очікує користувач. Існує два основних типи системних випробувань: функціональна перевірка та випробування робочих характеристик.

**Функціональна перевірка** не вимагає від тестувальника знань принципів роботи програмного продукту, але в той же час вона вимагає знання функціональних вимог, що пред'явлені до системи.

У межах **випробувань для визначення робочих характеристик** виконуються такі перевірки, як тестування в граничних режимах, навантажувальні випробування, контроль синхронізації та перевірки можливості відновлення.

Продукт користувачу для проведення приймальних випробувань може бути переданий. Якщо користувач належить тій компанії, що й розробники, то таке тестування часто називають **альфа тестуванням**. А якщо користувачі є замовники, готові працювати з програмним продуктом ще до його офіційної готовності, то таке тестування називається **бета тестуванням.**

Кінцевим типом приймальних випробувань є **установча перевірка**, за умовами якої завершена версія програмного продукту встановлюється у замовника з метою отримати від нього підтвердження, що програмний продукт відповідає вимогам та замовник згоден на його постачання.

**Сутність тестування**

*Тестування програмного забезпечення (Software Testing)* – це перевірка відповідності між реальною і очікуваною поведінкою програми, що здійснюється на кінцевому наборі тестів, обраних певним чином. (IEEE Guide to Software Engineering Body of Knowledge, SWEBOK, 2004) У більш широкому змісті, тестування - це одна з технік контролю якості, що включає в себе діяльність з планування робіт (Test Management), проектуванню тестів (Test Design), виконанню тестування (Test Execution) і аналізу отриманих результатів (Test Analysis).

*Верифікація (verification)* програми і її компонентів з метою визначення чи задовольняють результати поточного етапу розробки умовам, сформованим на початку цього етапу (IEEE). Тобто чи виконуються цілі, терміни, задачі з розробки проекту, визначені на початку поточної фази.

*Валідація (validation)* - це визначення відповідності розроблювального ПЗ очікуванням і потребам користувача, вимогам до системи. *План тестування (Test Plan)* - це документ, що описує обсяг робіт з по тестування, починаючи з опису об'єкта, стратегії, розкладу, критеріїв початку і закінчення тестування, до необхідного в процесі роботи устаткування, спеціальних знань, а також оцінки ризиків з варіантами їхнього рішення.

*Тест дизайн (Test Design)* - це етап процесу тестування ПЗ, на якому проектуються і створюються тестові випадки (тест кейси), відповідно до визначених раніше критерій якості і цілей тестування.

*Тестовий випадок (Test Case)* - це сукупність кроків, конкретних умов і параметрів, необхідних для перевірки реалізації функції або її частини, що тестуються.

*Звіт про помилку/Дефект Репорт (Bug Report)* - це документ, що описує ситуацію або послідовність дій, що призвели до некоректної роботи об'єкта тестування, із вказівкою причин і очікуваного результату.

*Тестове Покриття (Test Coverage)* - це одна з метрик оцінки якості тестування, що представляє із себе щільність покриття тестами вимог або коду, що виконується.

*Деталізація Тест Кейсів (Test Case Detalization)* - це рівень деталізації опису тестових кроків і необхідного результату, при якому забезпечується розумне співвідношення часу проходження до тестового покриття.

*Час Проходження Тест Кейса (Test Case Pass Time)* - це час від початку проходження кроків тест кейса до одержання результату тесту.

При проведенні приймальних випробувань виконують два класи тестування: функціональне та не функціональне тестування. Нижче надано перелік видів тестування по кожному класу.

|  |  |
| --- | --- |
| А Функціональне | Б Нефункціональне |
| Функціональні тести базуються на функціях і особливостях, а також взаємодії з іншими системами, і можуть бути представлені на всіх рівнях тестування: компонентному або модульному (Component/Unit testing), інтеграційному (Integration testing), системному (System testing) і приймальному (Acceptance testing). Функціональні види тестування розглядають зовнішнє поводження системи. Далі перераховані одні з найпоширеніших видів функціональних тестів: | Нефункціональне тестування описує тести, необхідні для визначення характеристик програмного забезпечення, що можуть бути вимірювані різними величинами. У цілому, це тестування того, "Як" система працює. Основні види нефункціональних тестів: |
| |  |  | | --- | --- | | А.1Функціональне тестування (Functional testing) | Б.1 Види тестування продуктивності: Б.1.1 Навантажувальне тестування (Performance and Load Testing) Б.1.2 Стресове тестування (Stress Testing) Б.1.3 Тестування стабільності або надійності (Stability / Reliability Testing) Б.1.4 Об'ємне тестування (Volume Testing) | | |  |  | | --- | --- | | Б.1 Види тестування продуктивності:  Б.1.1 Навантажувальне тестування (Performance and Load Testing)  Б.1.2 Стресове тестування (Stress Testing) Б.1.3 Тестування стабільності або надійності (Stability / Reliability Testing)  Б.1.3 Тестування стабільності або надійності (Stability / Reliability Testing)  Б.1.4 Об'ємне тестування (Volume Testing) | Б.2. Тестування установки (Installation testing | |
| А.2 Тестування безпеки (Security and Access Control Testing) | Б.2. Тестування установки (Installation testing ) |
| |  |  | | --- | --- | | А.3 Тестування взаємодії (Interoperability Testing):  А.3.1 Тестування сумісності (compatibility testing);  А.3.2 Інтеграційне тестування (integration testing). | Б.3 Тестування зручності користування (Usability Testing) | | Б.3 Тестування зручності користування (Usability Testing) |
|  | Б.4 Тестування на відмовлення і відновлення (Failover and Recovery Testing) |

*А.1 Функціональне тестування (Functional Testing)* розглядає заздалегідь зазначену поведінку і ґрунтується на аналізі специфікацій функціональності компонента або системи в цілому. Функціональні тести ґрунтуються на функціях, виконуваних системою, і можуть проводитися на всіх рівнях тестування (компонентному, інтеграційному, системному, приймальному). Як правило, ці функції описуються у вимогах, функціональних специфікаціях або у виді випадків використання системи (use cases). Тестування функціональності може проводиться в двох аспектах:

• вимоги;

• бізнеси-процеси.

Тестування в перспективі «вимоги» використовує специфікацію функціональних вимог до системи як основу для дизайну тестових випадків (Test Cases). У цьому випадку необхідно зробити перелік того, що буде тестуватися, а що ні, визначають пріоритетність вимог на основі ризиків (якщо це не зроблено в документі з вимогами), а на основі цього переліку пріоритетів формуються тестові сценарії (test cases). Це дозволяє сфокусуватися при тестуванні на важливішому функціоналі. Тестування в перспективі «бізнеси-процеси» використовує знання цих самих бізнесів-процесів, що описують сценарії щоденного використання системи. У цій перспективі тестові сценарії (test scripts), як правило, ґрунтуються на випадках використання системи (use cases).

*А.2 Тестування безпеки (Security and Access Control Testing)* – це перевірка безпеки системи, а також аналіз ризиків, пов'язаних із забезпеченням цілісного підходу до захисту додатка, атак хакерів, вірусів, несанкціонованого доступу до конфіденційних даних. Тестування безпеки може виконуватися як автоматизовано так і в ручну, включаючи перевірку як позитивних, так і негативних тестових випадків. Ґрунтується на трьох основних принципах - *це конфіденційність, цілісність і доступність* (confidentiality, integrity, availability)

*А.3 Тестування взаємодії (Interoperability Testing)* – це функціональне тестування, що перевіряє здатність додатка взаємодіяти з одним і більш компонентами або системами, що включає в себе тестування сумісності (compatibility testing) і інтеграційне тестування (integration testing).

*Б.1 Види тестування продуктивності (Performance testing)* Задачею *тестування продуктивності (Performance testing)* є визначення масштабованості додатка під навантаженням, при цьому відбувається:

• вимір часу виконання обраних операцій при визначених інтенсивностях виконання цих операцій;

• визначення кількості користувачів, що одночасно працюють з додатком;

• визначення границь прийнятної продуктивності при збільшенні навантаження (при збільшенні інтенсивності виконання цих операцій);

• дослідження продуктивності на високих, граничних, стресових навантаженнях.

*Б.1.1 Навантажувальне тестування (Load vs Performance Testing).*

В англомовній термінології ви можете так само знайти ще один вид тестування - Load Testing - тестування реакції системи на зміну навантаження (у межах припустимого). Load і Performance переслідують одну й теж саму мету: перевірка продуктивності (часів відгуку) на різних навантаженнях. Власне тому їх не розділяють.

*Б.1.2 Стресове тестування (Stress Testing)* дозволяє перевірити наскільки додаток і система в цілому працездатні в умовах стресу і також оцінити здатність системи до регенерації, тобто до повернення до нормального стану після припинення впливу стресу.

Стресом у даному контексті може бути підвищення інтенсивності виконання операцій до дуже високих значень або аварійна зміна конфігурації сервера.

Також однієї з задач при стресовому тестуванні може бути оцінка деградації продуктивності, у такий спосіб мети стресового тестування можуть перетинатися з цілями тестування продуктивності.

*Б.1.3 Тестування стабільності або надійності (Stability / Reliability Testing)* – це перевірка працездатності додатка при тривалому (багатогодинному) тестуванні із середнім рівнем навантаження. Часи виконання операцій можуть грати в даному виді тестування другорядну роль.

При цьому на перше місце виходить відсутність витоків пам'яті, перезапусків серверів під навантаженням і інші аспекти, що впливають саме на стабільність роботи.

*Б.1.4 Об'ємне тестування (Volume Testing)* -це одержання оцінки продуктивності при збільшенні обсягів даних у базі дані додатки, при цьому відбувається:

• вимір часу виконання обраних операцій при визначених інтенсивностях виконання цих операцій;

• може вироблятися визначення кількості користувачів, що одночасно працюють з додатком

*Б.2 Тестування установки (Installation Testing)* направлено на перевірку успішної інсталяції і настроювання, а також відновлення або видалення ПЗ. Інсталяція відбувається автоматично вручну та за допомогою візардів.

*Б.3 Тестування зручності користування (Usability Testing)* - це метод тестування, спрямований на встановлення ступеня зручності використання, навчання, зрозумілості і привабливості для користувачів розроблювального продукту в контексті заданих умов.

**Зміст тестового плану**

Вступ (Introduction) подається в довільній формі

Мета (Purpose)

• Виявлення існуючих проектів і програмних компонентів, які необхідно перевірити.

• Перелік вимог для проведення випробування.

• Рекомендації щодо опису стратегії тестування.

• Визначення необхідних ресурсів і забезпечення оцінки випробувань.

• Перелік тестових елементів проекту.

Довідкова інформація (Background**)**

Опис елементів тестування (компоненти, додатки, системи тощо).

Інформація, щодо основних функцій і можливостей, архітектури, короткої історії проект

Галузь застосування (Scope), описують:

• Типи, наприклад, групи, інтеграцію, систему, етапи тестування (функціональність і п

• Перелік особливостей функцій, що будуть протестовані.

• Перелік пропозицій, що можуть вплинути на проектування, розробку тестування.

• Перелік усіх ризиків й непередбачуваних обставин, що можуть вплинути на розробку

• Перелік обмежень, які можуть вплинути на проектування, розробку тестування.

Визначення проекту (Project Identification)