**Лабораторна робота №1** Оцінка якості програмного забезпечення

**Мета:** Оцінити якість загально доступного програмного забезпечення (ПЗ). Створити пакет документів, що включає вимоги, тестові сценарії, звіт дефектів.

**Постановка задачі**.

**1)** Завантажити та встановити загально доступне програмне забезпечення або вами розроблене ПЗ, наприклад, до курсової роботи. .

**2)** Знайти будь який дефект або недолік інтерфейсу користувача у програмі.

**3)** Специфікувати вимоги і тестовий сценарій, що покривають знайдену проблему.

**Завдання до лабораторної роботи**

1. Ознайомитись з теоретичною частиною та рекомендаціями до виконання роботи.
2. Специфікувати вимоги для обраної частини програми.
3. Визначити набір тестових сценаріїв, які покривають вимоги, з метою подальшого використання для тестування програмного забезпечення та пошуку дефектів в подальших лабораторних роботах.
4. Оформите звіт, в якому надайте:

* Відомості про обрану програму (наприклад, скриншот, короткий опис, призначення тощо);
* Перелік вимог до програми;
* Визначення тестового сценарію для визначення виконання вимоги.

Файл зі звітом надати з іменем у форматі :

**QA<Номер групи><Номер лекції / лабораторної>[-<Номер завдання>][літера позначення типу роботи L – лекція, R – лабораторна]<Прізвище англійською>**. Наприклад, **QA4101R**buts.doc.

1. Файл зі звітом відішліть на електронну адресу викладача.

**Строк виконання цієї роботи ІПЗ-31 25.09.2023**

**ТЕОРЕТИЧНА ЧАСТИНА**

**Якість програмного забезпечення**

Як перевірити, що вимоги визначені досить повно і описують усе, що очікується від майбутньої програмної системи? Це можна зробити, простеживши, чи всі необхідні аспекти якості ПЗ відбиті в них. Саме поняття якісного ПЗ відповідає уявленню про те, що програма досить успішно справляється з всіма покладеними на неї задачами та не приносить проблем ні кінцевим користувачам, ні їхньому начальству, ні службі підтримки, ні фахівцям по продажах. Та й самим розробникам створення якісної програми приносить набагато більше задоволення.

Якщо попросити групу людей висловити свою думку щодо того, що таке якісне ПЗ, можна одержати наступні варіанти відповідей:

* + Його легко використовувати
  + Воно демонструє гарну продуктивність.
  + У ньому немає помилок
  + Воно не псує користувацькі дані при збоях
  + Його можна використовувати на різні платформах
  + Воно може працювати 24 години на добу й 7 днів в тиждень.
  + У нього легко додавати нові можливості
  + Воно задовольняє потреби користувачів
  + Воно добре документоване

Все це дійсно має безпосереднє відношення до якості ПЗ. Але ці відповіді виділяють характеристики, важливі для конкретного користувача, розробника або групи таких осіб. Для того щоб задовольнити потреби всіх сторін (кінцевих користувачів, замовників, розробників, адміністраторів систем, у яких воно буде працювати, організацій-регуляторів та ін.), для досягнення міцного положення розроблюваного ПЗ на ринку та підвищення потенціалу його розвитку необхідне врахування всієї сукупності характеристик ПЗ, що є важливими для всіх зацікавлених осіб

Наведені вище відповіді показують, що якість ПЗ може бути описана великим набором різнорідних характеристик. Такий підхід до опису складних понять називається **холістичним** (від грецького слова ολόκληρος - ціле). Він не дає єдиної концептуальної основи для розгляду питань, що зачіпаються, яку дає цілісна система уявлень (наприклад, Ньтонівська механіка у фізиці або класичній теорії обчислюваності на основі машин Тьюринга), але дозволяє, принаймні, не упустити нічого істотного.

Характеристики якості ми розглядали в курсі основ програмної інженерії і заглибимось в наступній лекції.

Характеристики та атрибути якості ПЗ дозволяють систематично описувати вимоги до нього, визначаючи, які властивості ПЗ за даною характеристикою хочуть бачити зацікавлені сторони. **Таким чином, вимоги повинні визначати наступне**.

ПЗ має , наприклад:

* *дозволяти клієнтові оформити замовлення й забезпечити їхню доставку;*
* *забезпечувати контроль якості будівництва й відслідковувати проблемні місця;*
* *підтримувати потрібні характеристики автоматизованого процесу виробництва, запобігаючи аварії й оптимальним способом використовуючи наявні ресурси*

**Бути надійним**, наприклад:

* *працювати 7 днів у тиждень і 24 години на добу;*
* *допускається непрацездатність протягом не більше 3 годин у рік;*
* *ніякі уведені користувачами дані при відмові не повинні губитися*

Бути зручним для користування, наприклад:

* *користувач повинен, знаючи назву товару й маючи середні навички роботи в Інтернет, знаходити потрібний йому товар за не більш ніж 2 хв.;*
* *інженер не професіонал з комп’ютерних технологій повинен протягом одного дня вміти розібратися в 80% функцій системи*

**Бути ефективним**, наприклад:

* *підтримувати обслуговування до 10000 запитів у секунду;*
* *час відгуку на запит при максимальному завантаженні не повинен перевищувати 3 с;*
* *час реакції на зміну параметрів процесу виробництва не повинен перевищувати 0.1 с;*
* *на обробку одного запиту не повинне витрачатися більше 1 MB оперативної пам'яті*

**Зручним для супроводу,** наприклад:

* *додавання в систему нового виду запитів не повинне вимагати більше 3 людино-днів;*
* *додавання підтримки нового етапу процесу виробництва не повинне коштувати більше $20000.*

**Бути пристосованим**, наприклад:

* *ПЗ повинне працювати на операційних системах Linux, Windows XP і MacOS X;*
* *ПЗ повинне працювати з документами у форматах MS Word і HTML;*
* *ПЗ повинне зберігати файли звітів у форматах MS Word 20\*\*, MS Excel 20\*\*, HTML, RTF та у вигляді звичайного тексту;*
* *ПЗ повинне сполучатися з існуючою системою запису даних про замовлення.*

**Питання для обговорення:**

**1.** Сертифікація програмного забезпечення в Україні.

**2.** Значення і вартість якості.

**3.** Навести рівні цілісності програмного забезпечення.

**4.** Пояснити значення і вартість якості (Value and Costs of Quality) при розробці програмного забезпечення.

**5.** Проаналізувати процес підвищення якості (Quality Improvement).

**6.** Надати визначення поняттю метрики як основи вимірювання.