

## ІНТЕЛЕКТУАЛЬНИЙ МОДУЛЬ ДЛЯ ПІДВИЩЕННЯ ЯКОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

<sup>1</sup>Вінницький національний технічний університет

### **Анотація**

*У роботі проаналізовано актуальність розробки інтелектуального модуля для підвищення якості програмного забезпечення. В даній роботі описується процес тестування для виявлення фактів розбіжностей між фактичною поведінкою програми та її вимогами. На основі проведеного аналізу створено UML-діаграму класів інтелектуального модуля.*

**Ключові слова:** інтелектуальний модуль, якість програмного забезпечення, тестування.

### **Abstract**

*In this article the relevance of the development of an intellectual module to improve software quality is analyzed. This paper describes the testing process for identifying discrepancies between the actual behavior of the program and its requirements. The UML diagram of intelligent module classes was created on the basis of the conducted analysis.*

**Keywords:** intelligent module, software quality, testing.

### **Вступ**

Користувачі сучасних інформаційних систем зазвичай взаємодіють з ними через спеціальні програмні модулі, показники якості яких залежать від рівня якості всього програмного забезпечення. Застосування стандартів для створення правильної та надійної архітектури, розробки та інтеграції програмних систем підвищує шанси на успішне створення системи та її безперебійну роботу, але складність системи при їх інтеграції може значно зрости [1].

Метою роботи є виявлення критеріїв якості програмного забезпечення та її досягнення за допомогою тестування. На основі проведеного аналізу необхідно створити UML-діаграму класів інтелектуального модуля.

### **Основна частина**

Якість програмного забезпечення – це набір його характеристик, які стосуються можливості задовольнити заявлені або очікувані потреби як користувачів, так і розробників та замовників. Існують наступні критерії якості програмного забезпечення [2]:

- просте у використанні;
- має хорошу продуктивність;
- відсутні помилки;
- програма не псує дані користувача при збоях;
- можливість використання на різних платформах;
- здатність працювати 24 години на добу;
- легко додаються нові можливості;
- задоволення потреб користувача;
- документованість.

Насправді, критеріями якості програмного забезпечення можна назвати ще багато факторів, серед яких зручність перевірки та відповідність стандартам, точність обчислень, ступінь стандартності інтерфейсів, функціональна повнота, стійкість до помилок, стандартність форматів даних, ефективність роботи, розширюваність, модульність, простота роботи, зручність навчання, захищеність, адекватність, економічна ефективність та ін.

Для досягнення якості програмного забезпечення найбільш дієвий спосіб – проведення тестування. Тестування – це процес пошуку програми на наявність помилок [3]. Помилки або

дефекти в програмі означають дефекти в розробці програмного продукту, що призводять до невідповідності очікуваним результатам при запуску та експлуатації розробленого програмного продукту. Іншими словами, завдання процесу тестування в основному полягає у виявленні фактів розбіжностей між фактичною поведінкою програми та її вимогами.

На основі аналізу розглянутої інформації було створено UML-діаграма класів інтелектуального модуля для підвищення якості програмного забезпечення (рис. 1).

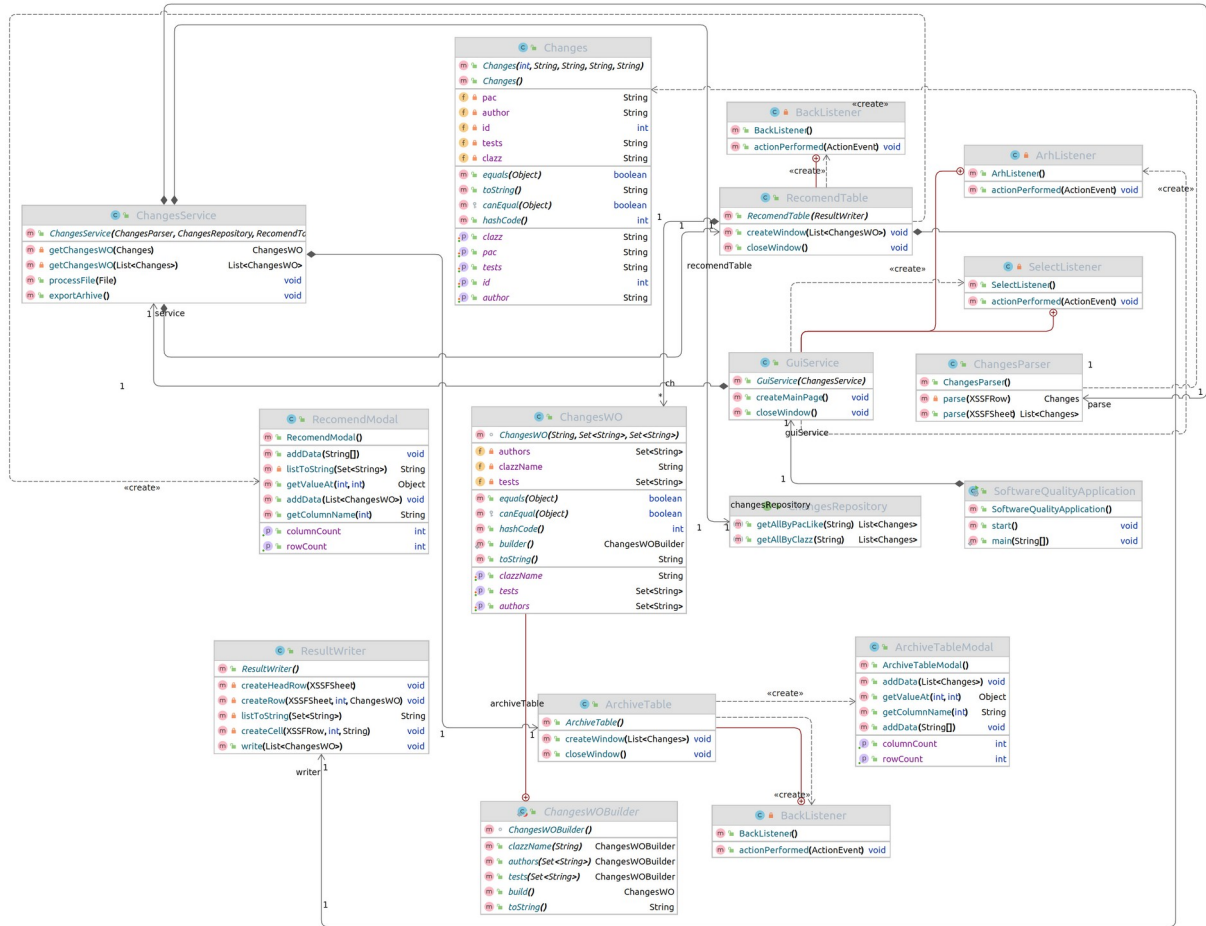


Рисунок 1 – UML-діаграма класів інтелектуального модуля

Для забезпечення тестування було застосована технологія BDD [4]. BDD має багато різних фреймворків. Програмний фреймворк – це набір готових до використання програмних рішень, що включають базовий дизайн, логіку та функціональність системи чи підсистеми. Відповідно, програмна структура може також включати утиліти, певні бібліотеки коду, скрипти і взагалі все, що полегшує створення та комбінування різних компонентів великого програмного забезпечення.

Програмну реалізацію інтелектуального модуля для підвищення якості програмного забезпечення здійснено мовою програмування Java, тому доцільно розглянути фреймворки, що працюють з обраною мовою програмування.

Cucumber - це behavior driven development(BDD), з відкритим вихідним кодом який працює з Ruby, Java, .NET, Flex або веб-додатками, написаними будь-якою мовою.

Instinct – це Behaviour Driven Development (BDD) структура для Java. Натхненна RSpec, Instinct забезпечує гнучку анотацію контекстів, специфікацій та акторів; має підтримку плагіна IntelliJ IDEA.

JBehave – це фреймворк на основі Java, створений для надання стимулу до співпраці між розробниками, бізнес-партнерами та іншими членами команди через автоматизовані сценарії. JNarrate призначений для написання специфікацій поведінки в Java. Narrative є основою для побудови тестів з BDD у вільній Java.

Easub – це BDD-фреймворк для платформи Java. Easub прагне включити виконувачу, але читабельну документацію.

## Висновки

У ході проведеного аналізу було доведено актуальність створення інтелектуального модуля для покращення якості програмного забезпечення. У результаті аналізу визначено критерії якості програмного забезпечення та обґрунтовано доцільність проведення тестування для визначення якості програмного забезпечення. На основі проведеного аналізу було створено UML-діаграма класів інтелектуального модуля, розглянуто фреймворки, що полегшують створення надійної системи тестування.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. The Art of Software Testing / Glenford J. Myers, Revised and Updated by Tom Badgett, Todd M.Thomas, Corey Sandler. - 2nd ed. - Hoboken, New Jersey.: John Wiley & Sons, Inc., 2004 - 234 р..
2. Андон Ф.І., Коваль Г.І., Коротун Т.М. Основи інженерії якості програмних систем. 2-ге видання. - К.: Академперіодика, 2007. - 672 с.
3. Що таке тестування програмного забезпечення та яке його значення: [Електронний ресурс]. Режим доступу: <https://www.quality-assurance-group.com/shho-take-testuvannya-programnogo-zabezpechennya-ta-yake-jogo-znachennya/> – Назва з екрану.
4. Нікітченко Я.Ю., Резанова В.Г. Автоматизоване тестування Веб-застосунку з використанням BDD-технологій.

**Толстая Дар'я Петрівна** – студентка групи 1КН-18б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, email: [dasha.tolstaya18@gmail.com](mailto:dasha.tolstaya18@gmail.com).

**Белзетський Руслан Станіславович** – канд. техн. наук, доцент кафедри інтеграції навчання з виробництвом, Вінницький національний технічний університет, e-mail: [belzetskiy@vntu.edu.ua](mailto:belzetskiy@vntu.edu.ua).

**Tolstaya Daria P.** – Department Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, email: [dasha.tolstaya18@gmail.com](mailto:dasha.tolstaya18@gmail.com).

**Belsetskiy Ruslan S.** — Cand. Sc. (Eng), Assistant Professor, Head of Training and Production Integration Department, Vinnytsia National Technical University, Vinnytsia.