

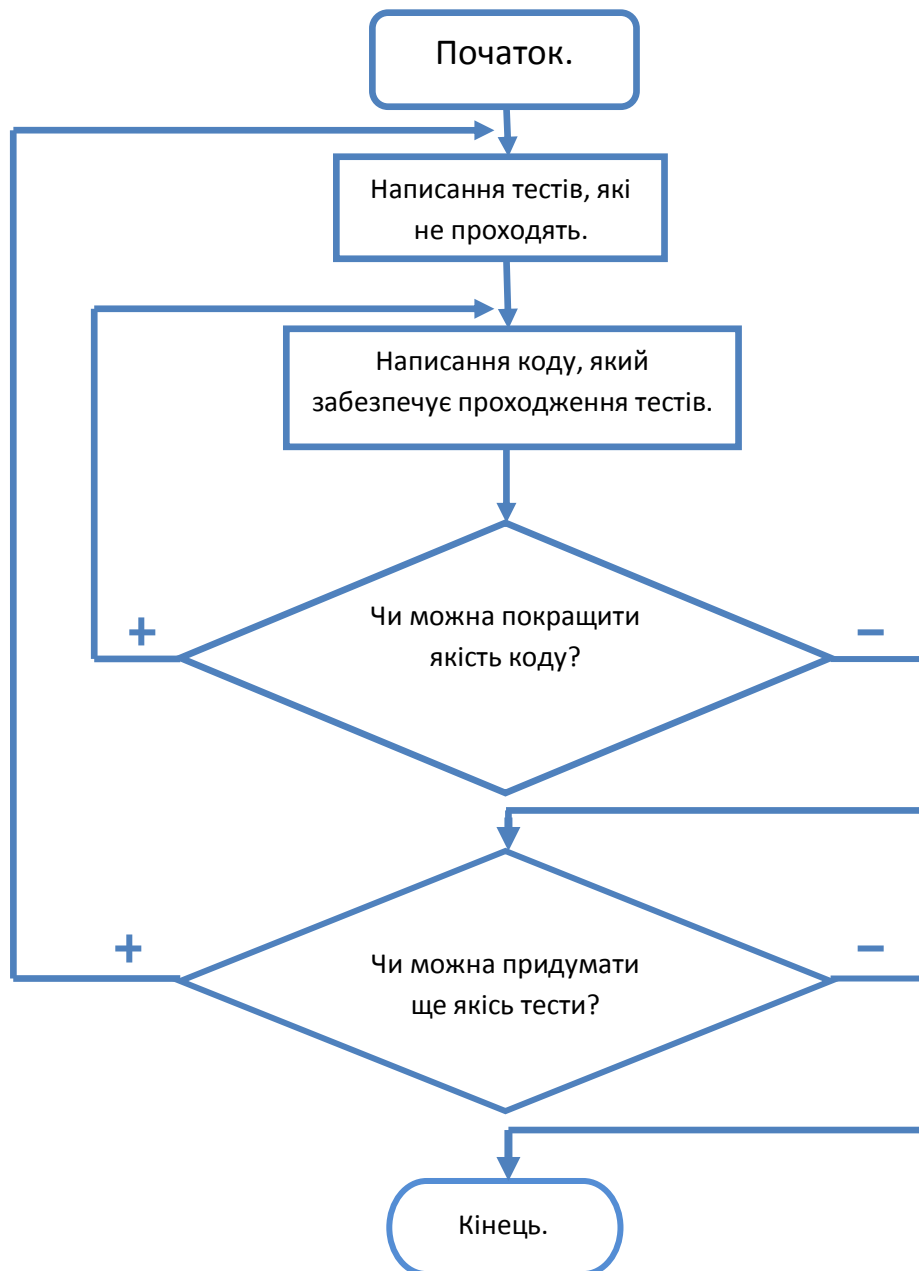
## **ВИКОРИСТАННЯ СУЧАСНОЇ ТЕХНІКИ TEST DRIVEN DEVELOPMENT (TDD) ДЛЯ НАПИСАННЯ ЯКІСНОГО ПРОГРАМНОГО КОДУ**

*Лупей Максим Іванович*

*e-mail: [maximlp1990@gmail.com](mailto:maximlp1990@gmail.com)*

Сучасна розробка програмного забезпечення являє собою складний процес, який повинен пройти кожен розробник для підтвердження правильності функціонування свого коду. Існує досить багато технік написання програмного коду. Однією з них є Test Driven Development [1-4], скорочена назва якої – TDD. Дану техніку можна розглядати як алгоритм, якому потрібно слідувати для написання якісного програмного коду. Під якістю розуміють сукупність властивостей, що визначають ступінь придатності програмного забезпечення для використання за призначенням, а також виділяються наступні характеристики якості програмного продукту: функціональність, надійність функціонування, зручність використання, раціональність, супроводжуємість та масштабуємість.

Основою техніки TDD є пошук наборів тестів, які не проходять у поточному варіанті програмного коду і тим самим потребують його вдосконалення. Розпишемо детальніше алгоритм написання програмного коду за TDD. На першому кроці визначаємо, який функціонал повинен нести наш майбутній програмний код. Далі розробляємо тести, які не проходять у даному функціоналі. Наступним кроком буде написання коду (внесення змін у поточний функціонал) для забезпечення успішного проходження тесту. Удосконалюємо роботу та загальний вигляд коду. Далі проводимо дослідження, чи можна написати ще тести, які будуть покривати даний функціонал. Якщо такі тести існують то переходимо до початку алгоритму. В іншому випадку алгоритм закінчено. На малюнку 1 приведено даний алгоритм у вигляді блок-схеми.



Мал. 1. Блок-схема алгоритму написання програмного коду за TDD

Отже, ми отримаємо добре функціонуючий код, який дійсно працює належним чином, та набори тестів до цього коду. Зауважимо, що отримані набори тестів можна запускати у будь-який момент часу для модульного та регресивного тестування роботи даного програмного коду, що являє собою одним із найбільших плюсів техніки TDD. Існують спеціальні програми-аналізатори такі як dotCover, codeCover, GroboCodeCoverage, та багато інших, які аналізують код і показують відсоток покриття програмного коду тестами. Покриття коду тестами

на 80% рахується непоганим показником. Проте деякі компанії вимагають 95% і більше. Програмний код, добре покритий тестами, - це запорука стабільності системи. Адже у наш час, з розвитком технології SaaS (software as a service) та web-технологій, програмний код завжди змінюється і пристосовується до потреб користувачів. При написанні нового, або зміні існуючого коду, при хорошому покритті системи наборами тестів, небажані дефекти виявляються на початковій стадії, що дає можливість відразу їх виправити на етапі тестування. Якщо в системі з'являються дефекти, то компанія може втратити дуже багато грошей. Для того, щоб не втрачати користувачів та гроші при небажаній роботі свого програмного продукту, та запобігти виникненню дефектів, сучасні компанії приділяють велику увагу підбору правильної техніки забезпечення якості розроблюваного або супроводжуемого продукту.



Мал. 2. Аналіз використання різних технік при розробці сучасного програмного забезпечення [5]

Використання сучасної техніки Test Driven Development дозволяє проводити модульне тестування та тестування функціональності; виявляти помилки в

процесі написання програмного коду; використовувати написані тести в регресивному тестуванні. TDD міняє мислення програміста при написанні коду. Адже замість того, щоб сісти і писати несистематизований набір коду для програмного забезпечення, який згодом може призвести до дефектів у системі, програміст починає думати над бажаним функціоналом і формує набори тестів для вдосконалення програмного коду. Коли тести сформовані, залишається внести зміни в програмний код, щоб він задовольняв написані тести. На даний момент, у великої кількості потужних компаній по розробці програмного забезпечення, техніки TDD та регресивне тестування підтримується на рівні менеджменту. На малюнку 2 наведено результати дослідження, які техніки використовуються командами програмістів при розробці сучасного програмного забезпечення. Із наведених результатів видно, що регресивне тестування для перевірки роботи програмного продукту використовує 71% команд програмістів, а техніку TDD при написанні коду – 53% команд.

## **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. <http://www.agiledata.org/> - Introduction to Test Driven Development (TDD) - 20.11.2012 [Електронний ресурс] - Режим доступу: <http://www.agiledata.org/essays/tdd.html>
2. Crispin, L. Agile Testing: A Practical Guide for Testers and Agile Teams / Lisa Crispin, Janet Gregory. – Addison-Wesley, 2008. – 533 с.
3. Beck, K. Test Driven Development: By Example/ Kent Beck. – Addison-Wesley Professional; 1 edition (November 18, 2002). - 240 с.
4. <http://www.nunit.org> – Unit - testing framework. Quick start – 20.11.2012 [Електронний ресурс] – Режим доступу: <http://www.nunit.org/index.php?p=quickStart&r=2.6.2>
5. <http://www.ambysoft.com> - Surveys Exploring The Current State of Information Technology Practices - 20.11.2012 [Електронний ресурс] – Режим доступу: <http://www.ambysoft.com/surveys>