

## КОМПЛЕКС АВТОМАТИЗОВАНИХ ТЕСТІВ GUI ГРАФІЧНОГО РЕДАКТОРА В ЗАДАЧІ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ УПРАВЛІННЯ СИСТЕМАМИ КОНДИЦІОНУВАННЯ

Салавага О.Ю.

*Тернопільський національний економічний університет, магістрант*

### I. Постановка проблеми

Автоматизоване тестування програмного забезпечення (ПЗ) – частина процесу тестування на етапі контролю якості під час розробки програмного забезпечення [1]. Для кінцевих користувачів найбільш важливою частиною взаємодії з ПЗ є GUI (Graphical User Interface – графічний користувацький інтерфейс). Перевірка усіх елементів інтерфейсу користувача є досить комплексною роботою і потребує створення певного набору тестів, особливо в умовах наявності великої кількості змін та корективів у ПЗ [3, 4].

Об'єктом дослідження у даній роботі є програмне забезпечення для управління системами кондиціонування, до складу якого входить не тільки GUI, а й підсистеми для проведення розрахунків обладнання, фізичних показників, експорту схем в такі формати як XLS, DXF та PDF. В задачі проектування та в ході регулярного нарощення функціональності програмного продукту, тобто додавання нових функцій існує необхідність підтримки відповідного високого рівня якості в умовах обмеження часовими, трудовими та матеріальними ресурсами. Звідси випливає потреба пошуку методу оптимізації роботи із автоматизованими тестами.

### II. Мета роботи

Метою даної роботи є розробка комплексу гнучких та легко підтримуваних автоматизованих тестів для перевірки GUI програмного забезпечення, що служить середовищем для управління системами кондиціонування.

### III. Особливості реалізації комплексу автоматизованих тестів

Стандартна імплементація автоматизованих тестів базується на використанні шляхів до об'єктів, що змушує модифікувати вміст кожного сценарію у випадку зміни назв елементів програми. Це є досить не раціонально, особливо, коли тестів у наявності десятки, а то й сотні. У зв'язку з даною проблемою запропоновано використання вказівників на об'єкти, які збираються в межах одного сценарію, що значно пришвидшує та полегшує роботу розробника. При використанні такого методу реалізації значно оптимізується та пришвидшується робота розробника тестів.

Кожен тест є окремим модулем і його робота не залежить від попереднього, що дозволяє відтворити цілий комплекс тестів в межах одного запуску (рис. 1).

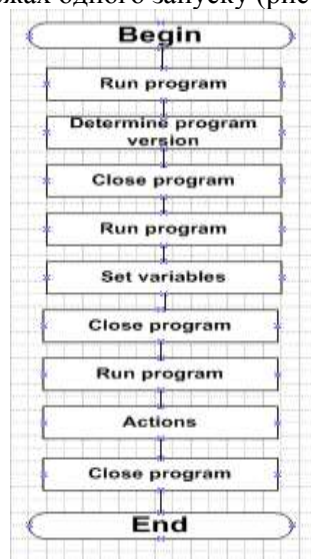


Рисунок 1 - Основний алгоритм задачі

Кожен модуль несе в собі перевірку лише одного елемента GUI, щоб можна було конкретно визначити проблемну ділянку у функціональності. Після створення ряду сценаріїв будується дерево проекту, куди додаються тести у певній послідовності (рис. 2).

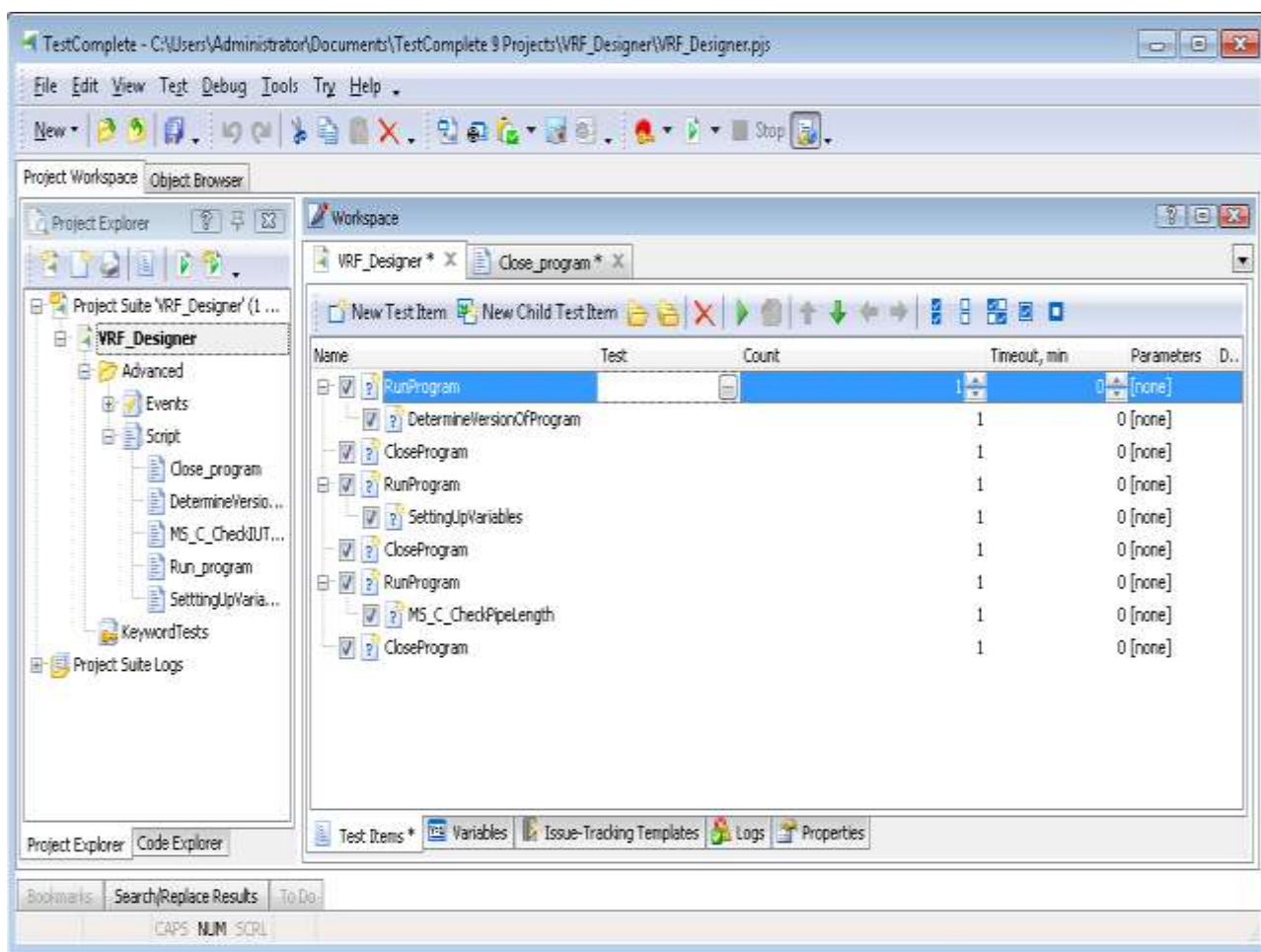


Рисунок 2 – Архітектура комплексу автоматизованих тестів в середовищі TestComplete 9

Така архітектура реалізується з допомогою середовища для автоматизованого тестування TestComplete 9, яке було обрано згідно до поставлених вимог [2, 5]. Вибір відбувався відповідно до компонентів програмної системи, тестування якої необхідно забезпечити, та їх реалізації мовою програмування C#.

### Висновок

Використання обраного підходу та засобу для розробки і виконання комплексу автоматизованих тестів дозволило створити оптимізовану систему сценаріїв із значною економією часу та ресурсів на її впровадження і підтримку, що пришвидшило та полегшило процес тестування графічного користувачького інтерфейсу програмного забезпечення для управління системами кондиціонування.

### Список використаних джерел

1. Савин Р. Тестирование Дот Ком или Пособие по жестокому обращению с багами в интернет-стартапах. — М.: Дело, 2007. — 312 с.
2. Учебник по TestComplete Режим доступу: <http://tctutorial.ru>
3. Dustin E. Automated software testing: introduction, management and performance / Dustin E, Rashka J., Paul J. — AddisonWesley, 1999. — 575 p.
4. Kolawa, Adam; Huizinga, Dorota (2007). Automated Defect Prevention: Best Practices in Software Management. Wiley-IEEE Computer Society Press.
5. TestComplete 9 Documentation Режим доступу: <http://support.smartbear.com/viewarticle/32760/>