

Г.Л. Євтушенко, О.І. Євтушенко, В.І. Кузнецов

**БАГАТОКРИТЕРІАЛЬНИЙ ПОРІВНЯЛЬНИЙ АНАЛІЗ  
ІНСТРУМЕНТІВ ДЛЯ АВТОМАТИЗАЦІЇ  
ТЕСТУВАННЯ ПРОГРАМНИХ ЗАСОБІВ**

*Анотація. В даній роботі проаналізовані найпопулярніші інструменти, що використовуються для автоматизації процесу тестування настільних та веб-додатків. Для побудови рейтингу розглянутих інструментів обрано метод аналізу ієрархій «у абсолютних вимірюваннях» (нормативний MAI, MAI-H).*

*Ключові слова: автоматизація тестування програмних засобів, багатокритеріальний аналіз, метод аналізу ієрархій у абсолютних вимірюваннях.*

Світ сучасних інформаційних технологій стрімко розвивається. Кожен день відбуваються розробка нових ідей, впроваджуються нові методи розробки різноманітних додатків. Пищуться тисячі рядків програмного коду, які полегшують наше життя. Проте, і вже створені проекти нікуди не зникають. Вони продовжують розвиватися, і вдосконалюватися. Адже без потрібного базису неможливо рухатися вперед.

Разом з тим неможливо уявити сучасний процес розробки чи то настільного, чи то веб-, чи то мобільного додатку без тестування. Адже у сучасному стрімкому, і інколи безжалільному світі ціна навіть невеликої помилки різко збільшується. Тому тестування – це необхідний етап процесу розробки, який завжди стоїть на сторожі якості програмного продукту. І для полегшення та прискорення процесу тестування часто його автоматизують.

Основна мета автоматизації тестування – скорочення витрат на випробування програми після її модернізації. Періодично повторювані однотипні перевірки забирають багато часу в циклі розробки. Автоматизація скорочує етап тестування і вивільняє головний ресурс компанії – робочий час фахівців. Інша, не менш очевидна перевага такого тестування – підвищення якості випробувань, що

гарантують надійність продукту. Адже збитки від дефектів, виявлені лише на стадії промислової експлуатації, можуть бути дуже високими, як і незадоволеність замовників [1].

В даній роботі виконаний багатокритеріальний порівняльний аналіз сучасних інструментів для автоматизації тестування настільних та веб-додатків. Для побудови рейтингу розглянутих інструментів обрано метод аналізу ієрархій «у абсолютних вимірюваннях» (нормативний MAI, MAI-H), який визнається точнішим за метод зважених сум (МЗС), який зазвичай використовується у складанні рейтингів. Разом з тим MAI-H не має обмежень щодо кількості об'єктів порівняння, як стандартний варіант MAI «у відносних вимірюваннях» (дескриптивний MAI, MAI-D).

MAI-H дозволяє порівнювати велику (у принципі – необмежену) кількість альтернатив; важливо також, що він надає можливість використання шкал інтенсивностей (лінгвістичних стандартів), які можна застосовувати для роздільної оцінки об'єктів (інструментів тестування у нашому випадку).

Одинадцять інструментів проаналізовані та оцінені за наступними критеріями (табл. 1):

1) Платформа, для тестування якої призначений той чи інший інструмент автоматизації (Windows, web, Android та ін.).

2) Мови програмування, на яких є можливість розробляти тестові випадки.

3) Ліцензія – варіант розповсюдження, за яким можна використовувати даний інструмент (безкоштовно чи платно).

4) Формат, у якому розповсюджується інструмент автоматизації, чи то фреймворк, який інтегрується у платформу для якої створений додаток, який перевіряється, чи самостійна середа розробки з власними додатковим функціоналом.

5) Складність – рівень підготовки та рівень вимог до інженеро-тестувальника. Рівень знання мови програмування та теоретичний базис тестування: Високий, середній та низький.

Для побудови рейтингу запропонованих інструментів автоматизації тестування програмних засобів застосовано метод аналізу ієрархій у абсолютних вимірюваннях (нормативний MAI, MAI-H) [13], який програмно реалізований у системі підтримки прийняття рішень NooTron [14].

Таблиця 1

Порівняльний аналіз інструментів автоматизації програмних засобів

Інструменти автоматизації	Платформи	Мови	Ліцензія	Формат	Складність
.Net UI Automation [2]	Windows	C#	Безкоштовна	Фреймворк	Середня
White project [3]	Windows	C#	Безкоштовна	Фреймворк	Середня
SilkTest [4]	Windows, UNIX, Linux, web	4test, Java, C#	Платна	Середа розробки	Висока
Coded UI Test [5]	Windows	C#	Платна	Фреймворк	Середня
Test Complete [6]	Windows, web, Android, iOS	VBScript, JScript, C++Script, C#Script, Python, VB	Платна (є безкоштовна версія)	Середа розробки	Висока
UFT [7]	Windows, web, Android, iOS	C#, Java	Платна (є безкоштовна версія)	Середа розробки	Висока
Ranorex [8]	Windows, web, Android, iOS	C#, VB.NET	Платна (є безкоштовна версія)	Середа розробки	Середня
Selenium [9]	Web	Java, C#, JS, Phyton, PHP, Ruby	Безкоштовний	Фреймворк	Середня
Katalon Studio [10]	Web, Windows, Linux, Android, iOS	Java, C#, JS, Phyton	Платна (є безкоштовна версія)	Фреймворк / Середа розробки	Середня
Watir [11]	Web	Ruby	Безкоштовний	Фреймворк	Середня
TestPlant eggPlant [12]	Web, Windows, Linux, Android, iOS	Java, C#, JS, Phyton, PHP, Ruby	Платна (є безкоштовна версія)	Середа розробки	Середня

На першому етапі у МАІ-Н необхідно визначити пріоритети критеріїв відносно цілі за допомогою матриці парних порівнянь (рис.1).

Название		Кр1	Кр2	Кр3	Кр4	Кр5	ЛПр.
<b>Кр1</b>	Платформи	1	1	4	3	5	0.352
<b>Кр2</b>	Мови	1	1	4	3	5	0.352
<b>Кр3</b>	Ліцензія	1/4	1/4	1	2	3	0.132
<b>Кр4</b>	Формат	1/3	1/3	1/2	1	3	0.110
<b>Кр5</b>	Складність	1/5	1/5	1/3	1/3	1	0.054
		Dim	Lam	CI	CR		
		5.000	5.189	0.047	0.042		

Рисунок 1 – Матриця парних порівнянь критеріїв

Отже, найголовнішими критеріями виявилися – платформа та мови програмування.

Далі необхідно визначити шкалу інтенсивностей для кожного критерію, або – загальну для всіх критеріїв, як у нашому випадку (рис. 2).

	Название	Инт5.1	Инт5.2	Инт5.3	Инт5.4	ЛПр.
Инт5.1	Высокий	1	2	3	4	0.467
Инт5.2	Вище середнього	1/2	1	2	3	0.277
Инт5.3	Середній	1/3	1/2	1	2	0.160
Инт5.4	Низький	1/4	1/3	1/2	1	0.096

Dim	Lam	CI	CR
4.000	4.031	0.010	0.011

Рисунок 2 – Матриця парних порівнянь шкали інтенсивностей критеріїв

На наступному етапі задаються альтернативи (у нашій задачі – це інструменти автоматизації тестування ПЗ) та їхні оцінки за критеріями у шкалі інтенсивностей (рис. 3). Після чого розраховуються глобальні пріоритети альтернатив та будується рейтинг (рис. 4).

№	Альтернативы	Кр1	Кр2	Кр3	Кр4	Кр5
A1	.Net UI Automation	Середній	Середній	Високий	Середній	Середній
A2	White project	Середній	Середній	Високий	Середній	Середній
A3	SilkTest	Вище середнього	Вище середнього	Низький	Вище середнього	Високий
A4	Coded UI Test	Середній	Середній	Низький	Середній	Середній
A5	Test Complete	Високий	Високий	Середній	Вище середнього	Високий
A6	UFT	Високий	Вище середнього	Середній	Вище середнього	Високий
A7	Ranorex	Високий	Середній	Середній	Вище середнього	Середній
A8	Selenium	Низький	Високий	Високий	Середній	Середній
A9	Katalon Studio	Високий	Вище середнього	Середній	Високий	Середній
A10	Watir	Низький	Низький	Високий	Середній	Середній
A11	TestPlant eggPlant	Високий	Високий	Середній	Вище середнього	Середній

Рисунок 3 – Оцінки альтернатив по критеріям

З отриманого вище рейтингу можна зробити наступні висновки:

1. Найбільшу кількість балів отримали інструменти, які дозволяють охопити найбільшу кількість платформ, а також дають широкий вибір підтримки мов програмування.

2. За результатами отриманого рейтингу розглянуті інструменти автоматизації утворюють 5 класів за пріоритетністю (див. рис. 4): 1 – Test Complete, TestPlant eggPlant; 2 – Katalon Studio, UFT; 3 – Selenium, Ranorex, SilkTest; 4 – .Net UI Automation, White project; 5 – Watir, Coded UI Test.

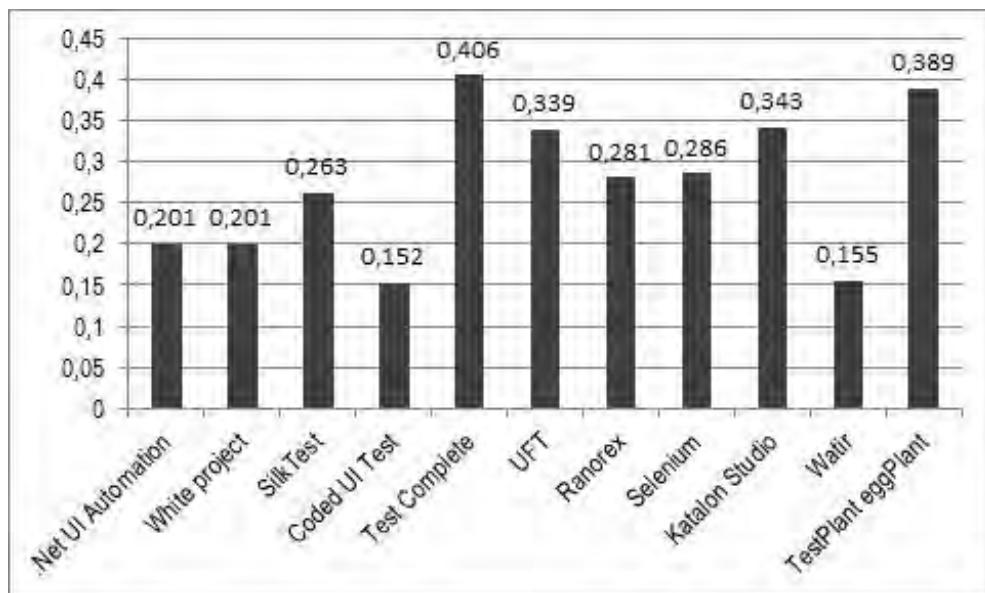


Рисунок 4 – Рейтинг інструментів автоматизації тестування програмних засобів

В процесі виконання роботи проаналізовані найпопулярніші інструменти, що використовуються для автоматизації процесу тестування. Також на основі описаних критеріїв було складено рейтинг на основі методу аналізу ієархій у абсолютних вимірюваннях із застосуванням СППР NooTron.

Можна зробити загальний висновок, що проблематика автоматизації тестування програмних продуктів є вельми актуальну. Тестування займає важливе місце в життєвому циклі додатків. Розвивається його інструментарій. Без професійного і якісного тестування неможливо створити якісний та успішний програмний продукт.

Результати проведеного аналізу можуть бути використані для обґрунтування вибору інструменту автоматизації тестування програмного засобу у конкретному проекті.

## ЛІТЕРАТУРА

1. Криспин Л., Грегори Дж. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд. – М.: «Вильямс», 2010. – С. 464.
2. UI Automation Overview [Електронний ресурс] – Електрон. дані – Режим доступу <https://docs.microsoft.com/en-us/dotnet/framework/ui-automation/ui-automation-overview/>, вільний. – Загол. з екрану.
3. White project Overview [Електронний ресурс] – Електрон. дані – Режим доступу <http://white.teststack.net/docs/getting-started/>, вільний. – Загол. з екрану.
4. SilkTest [Електронний ресурс] – Електрон. дані – Режим доступу <https://www.microfocus.com/products/silk-portfolio/silk-test/>, вільний. – Загол. з екрану.
5. Coded UI Test [Електронний ресурс] – Електрон. дані – Режим доступу <http://www.dotnetcurry.com/visualstudio/702/codedui-test-cuit-using-visual-studio-2010/>, вільний. – Загол. з екрану.
6. Test Complete Test [Електронний ресурс] – Електрон. дані – Режим доступу <https://smartbear.com/product/testcomplete/overview/>, вільний. – Загол. з екрану.
7. Unified Functional Testing Test [Електронний ресурс] – Електрон. дані – Режим доступу <https://software.microfocus.com/en-us/products/unified-functional-automated-testing/overview/>, вільний. – Загол. з екрану.
8. Ranorex Test [Електронний ресурс] – Електрон. дані – Режим доступу <https://www.ranorex.com/>, вільний. – Загол. з екрану.
9. Что такое Selenium? [Электронный ресурс] // Selenium/Webdriver. Автоматизация тестирования веб-приложения через браузер. – Режим доступа: <http://selenium2.ru/>
10. Katalon Studio [Електронний ресурс] – Електрон. дані – Режим доступу <https://www.katalon.com/>, вільний. – Загол. з екрану.
11. Watir [Електронний ресурс] – Електрон. дані – Режим доступу <http://watir.com/>, вільний. – Загол. з екрану.
12. TestPlant eggplant [Електронний ресурс] – Електрон. дані – Режим доступу <https://www.testplant.com/>, вільний. – Загол. з екрану.
13. Саати Т.Л. Принятие решений при зависимостях и обратных связях: Аналитические сети / Т.Л.Саати. – М.: Изд-во ЛКИ, 2008. – 360 с.
14. Система поддержки принятия решений NooTron [Електронний ресурс] – Електрон. дані – Режим доступу: <http://nootron.net.ua>, вільний. – Загол. з екрану.