KIỂM THỬ TỰ ĐỘNG WINDOWS FORM VỚI CODED UI TEST

● TRẦN VĂN THỌ

TÓM TẮT:

Kiểm thử tự động giao diện người dùng (UI Automation) là việc sử dụng các công cụ để thực hiện các test case, cho phép nhập dữ liệu kiểm thử vào hệ thống kiểm thử, so sánh kết quả mong đợi với kết quả thực tế và tạo ra các báo cáo kiểm thử chi tiết. Một trong số những công cụ kiểm thử UI Automation trên Windows Form là Coded UI Test (CUIT). CUIT sử dụng Visual Studio IDE để viết các kịch bản, vì việc ghi chép có thể được thực hiện bằng Visual Studio. Các kiểm thử này liên quan đến kiểm thử chức năng của các điều khiển UI (User Interface). CUIT kiểm thử các chức năng của toàn bộ ứng dụng bao gồm cả giao diện người dùng trên Windows Form.

Từ khóa: UI Automation, kiểm thử UI Automation, Windows Form, kiểm thử tự động, kiểm thử tự động giao diện người dùng.

1. Đặt vấn đề

Coded UI Test (CUIT) là một công cụ cho phép tự động hoá việc kiểm thử giao diện người dùng (Testing Tool User Interface Automation) điển hình tương tự như các công cụ kiểm thử tự động Selenium, QTP (Quick Test Professional),... CUIT đã được giới thiệu bởi Microsoft cùng với Visual Studio 2010. Nó tích hợp với Team Foundation Server.

CUIT đặc biệt hữu ích cho các bài kiểm thử chức năng của toàn bộ ứng dụng, giúp người dùng kiểm tra, rà soát lại các chức năng của các phần giao diện, giúp đảm bảo sự ổn định của ứng dụng. CUIT sẽ rất hữu dụng khi cần rà soát lại các chức năng hay tính logic trong giao diện người dùng. Ngoài ra, nó cũng được sử dụng để tự động hoá những phần hay phải kiểm tra bằng tay. CUIT sử dụng Visual Studio IDE để viết các kịch bản, vì việc ghi chép có thể được thực hiện bằng Visual Studio. CUIT là các bài kiểm thử tự động thúc đẩy ứng dụng của chúng ta thông qua giao diện người dùng (UI).

Trong bài báo này, tác giả giới thiệu về công cụ kiểm thử tự động giao diện người dùng Coded UI Test (CUIT) trong Visual Studio của Microsoft.

2. Phương pháp nghiên cứu

2.1. Kiểm thử giao diện đồ họa người dùng

2.1.1. Khái niệm

Kiểm thử giao diện đồ họa người dùng (Graphical User Interface Testing - GUI Testing) là:

- Một quá trình kiểm thử GUI của ứng dụng, xác định các lỗi xảy ra trong ứng dụng ở giai đoạn thiết kế.
- Được thực hiện để xác minh chức năng của GUI theo thông số kỹ thuật và phụ thuộc vào công nghệ được sử dụng.
- Đánh giá các tính năng điều khiển như menu, nút, biểu tượng, hộp văn bản, danh sách, hộp thoại, bố trí, màu sắc, kích thước phông chữ, định dạng văn bản,...
- Được thực hiện thủ công hoặc tự động với sự trợ giúp của các công cụ thường được thực hiện bởi công ty bên thứ ba thay vì nhà phát triển hoặc người dùng.
- Được sử dụng để thực hiện giá trị của từng đối tượng GUI và thực hiện các sự kiện GUI như nhấn

phím hoặc bấm chuột.

2.1.2. GUI Testing cần kiểm thử gì trong quá trình thực hiện kiểm thử

Dưới đây là danh sách gợi ý những gì cần kiểm thử trong quá trình thực hiện kiểm thử GUI Testing:

- Xác thực màn hình.
- Kích thước và vị trí của các phần tử GUI.
- Hình ảnh rõ ràng và được căn chỉnh.
- Điều hướng (liên kết).
- Phông chữ và căn chỉnh văn bản.
- Trường ngày và số.
- Điều kiện khả năng sử dụng và tính toàn vẹn của dữ liệu.
- Thông báo lỗi.
- Các trường bắt buộc.
- Mâu thuẫn của chữ viết tắt.
- Thanh tiến độ (Progress bars)
- Phím tắt.
- ...

2.1.3. Phương pháp tiếp cận để kiểm thử GUI

1. Kiểm thử thủ công: Người kiểm thử sẽ áp dụng kiến thức của họ và kiểm thử màn hình đồ họa theo yêu cầu kinh doanh.

2. Kiểm thử Record và Replay: Điều này được thực hiện bằng cách sử dụng các công cụ tự động hóa, các hành động Record và Replay của công cụ đó. Sau đó, thực thi các bước được ghi lại trên ứng dụng đang được kiểm thử trong khi Record/ Replay.

3. Kiểm thử dựa trên mô hình: Kiểm thử dựa trên mô hình được thực hiện theo hành vi của hệ thống. Các mô hình này có thể được phân thành 3 loại như sau:

- Mô hình dựa trên sự kiện: Dựa trên các sự kiện GUI sẽ xảy ra ít nhất một lần.
- Mô hình dựa trên trạng thái: Dựa trên trạng thái GUI được thực hiện ít nhất một lần.
- Mô hình miền: Dựa trên miền và chức năng của ứng dụng.

Với 3 mô hình trên yêu cầu cũng cần phải được theo sau:

- Xây dựng mô hình.
- Chỉ định đầu vào cho mô hình.
- Xác định kết quả mong đợi.
- Thực hiện kiểm thử.
- So sánh kết quả thực tế và mong đợi.
- Quyết định hành động trong tương lai sẽ được thực hiện.
- Công cụ kiểm thử giao diện người dùng hàng đầu.

2.1.4. Một số công cụ kiểm thử giao diện người dùng

Dưới đây là danh sách một số công cụ tốt nhất đang được sử dụng để kiểm thử GUI: (Bảng 1)

2.2. Công cụ kiểm thử Coded UI Test

Bảng 1. Một số công cụ tốt nhất được sử dụng để kiểm thử GUI

STT	GUI Testing	Logo
1	Abbot Java GUI Test Framework tool	Abbot Java GUI Test Framework
2	RAPISE by Inflectra testing tool	Rapise®
3	AutoIt UI testing	AUTOIT
4	CubicTest tool	CUBICTEST
5	eggPlant UI automation testing tool	eggplant BUILT BY
6	FitNesse testing tool	FITNESSE
7	Ascentialtest testing tool	AscentialTest Zeenyx
8	iMacros testing tool	iMacros
9	Maveryx user interface testing tool	

STT	GUI Testing	Logo
10	Ranorex Studio testing tool	
11	RIATest tool	
12	SilkTest tool	
13	Sikuli UI automation framework	
14	Squish GU testing tool	
15	SWTBot testing tool	
16	Selenium testing tool	
17	Telerik TestStudio tool	
18	TestComplete tool	
19	Test Anywhere tool	

STT	GUI Testing	Logo
20	TestPartner testing tool	
21	Jubula GUI testing tool	
22	IcuTest tool	
23	QF-Test tool	
24	QAliber testing tool	
25	RCP Testing Tool	
26	Sahi testing tool	
27	Soatest tool	
28	SWAT testing tool	

STT	GUI Testing	Logo
29	Telerik Testing Framework	Telerik
30	Telerik Test Studio GUI testing tool	Telerik Test Studio
31	Tellurium Automated Testing Framework	tellurium
32	TestStack. White Framework	
33	UI Automation Powershell Extensions	UiAutomation
34	Watir testing tool	Watir
35	Coded UI Test tool	Microsoft
36	Micro Focus Unified Functional Testing (UFT)	MICRO FOCUS Quality Center
37	Cucumber testing tool	cucumber
38	LoadUI testing tool	LoadUI
39	CrossBrowser Testing	
40	HP Quick Test Professional viết tắt là QTP	hp QuickTest Professional

2.2.1. Coded UI Test là gì?

Coded UI Test (CUIT) là một loại kiểm thử tự động phần mềm điển hình thúc đẩy ứng dụng thông qua giao diện người dùng (UI). Kiểm thử này liên quan đến kiểm thử chức năng của các điều khiển UI. CUIT cho phép kiểm thử các chức năng của toàn bộ ứng dụng bao gồm cả giao diện người dùng. CUIT sử dụng Visual Studio IDE để viết các kịch bản, vì việc ghi chép có thể được thực hiện bằng Visual Studio. (Hình 1)

2.2.2. Tính năng của Coded UI Test

Các tính năng của Coded UI Test bao gồm:

- Kiểm thử chức năng.
- Tạo mã trong VB/C#.
- Tích hợp với Application lifecycle management (ALM) story.
- Xây dựng, triển khai và kiểm thử trong phòng thí nghiệm hoặc như một phần của bản Build.
- Local, chạy từ xa, thu thập dữ liệu.
- Khả năng mở rộng phong phú.
- Ghi nhận ý định và phát lại linh hoạt.

2.2.3. Công nghệ hỗ trợ Coded UI Test

Với Coded UI, một ứng dụng có giao diện người dùng (UI) có thể dễ dàng được kiểm thử. Ứng dụng có thể dựa trên Window hoặc Web.

Coded UI hỗ trợ các công nghệ như:- Ứng dụng máy tính để bàn dựa trên Window.

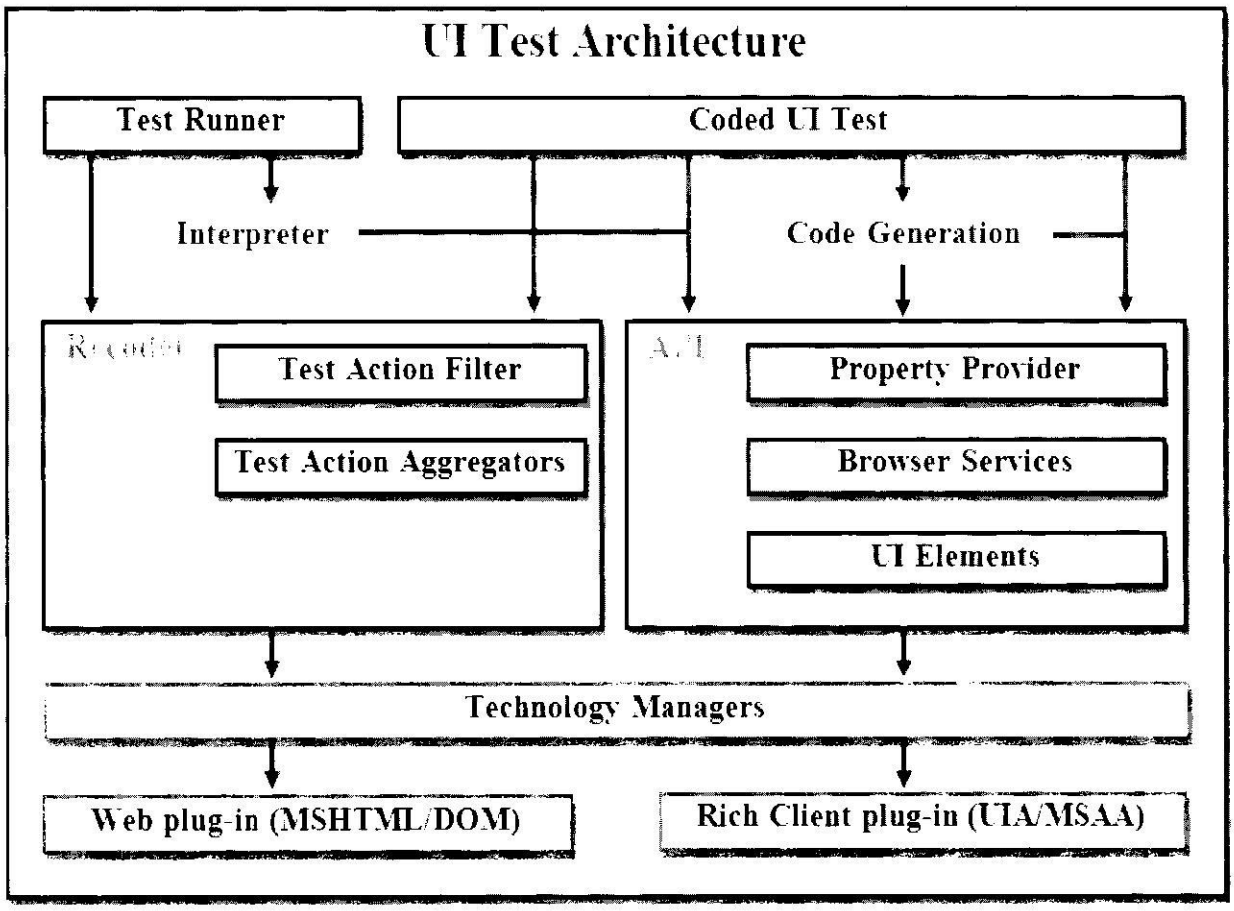
- Dịch vụ Web (SOAP, ASPX, v.v ...).
- Ứng dụng Window phone.
- WPF (Windows Presentation Foundation).
- Các ứng dụng web (HTML, Silverlight, HTML5).

2.2.4. Tại sao sử dụng Coded UI cho kiểm thử tự động

Lý do tại sao giao diện được mã hóa cho kiểm thử tự động được ưa thích như:

- Lập trình viên và người kiểm thử phần mềm có thể hợp tác hiệu quả bằng cách sử dụng cùng một

Hình 1: Kiến trúc UI Test



công cụ / ngôn ngữ.

- Hỗ trợ các dự án trên nền tảng Windows cũng như Webs.

- Cơ chế xác định các phần tử là một tính năng tuyệt vời của Coded UI. Ngoài ra, nó hỗ trợ đồng bộ hóa.

- Công cụ Playback hỗ trợ các tính năng như 'WaitForControlExist', 'WaitForReadyLevel', v.v ...

- Với sự giúp đỡ của "Test Agents" các kiểm thử tự động có thể được chạy trên các máy từ xa.

- Các nhóm tự động hóa có thể phát triển các kiểm thử phức tạp, sử dụng Coded UI với các lớp Framework.

- Người kiểm thử phần mềm có thể nắm bắt các ngoại lệ và ghi lại kết quả một cách hiệu quả bằng việc sử dụng log4net.dll.

- Công cụ Coded UI hỗ trợ lập trình mô tả. Nó cho phép người kiểm thử phần mềm tự động hóa các kịch bản dựa trên các thuộc tính đối tượng.

- Hỗ trợ các điều khiển AJAX.

2.2.5. Làm thế nào để tạo ra các Coded UI Test

Để tạo các kiểm thử Coded UI, chúng ta có thể thực hiện theo 3 cách sau:

1. Tạo từ bản ghi hành động hiện tại (chuyển đổi từ ghi chép kiểm thử thủ công).

Toàn bộ quá trình tạo ra kịch bản CUIT được chia làm 3 phần:

+ Phần 1: Recording - CUIT hỗ trợ người dùng ghi lại các tương tác của người dùng với UI. Các hoạt động được ghi lại đó tạo nên kịch bản lệnh CUIT.

+ Phần 2: Playing back - Trong giai đoạn này, chúng ta sẽ thực thi các kịch bản đã được ghi lại để xác minh và kiểm soát độ ổn định và tỷ lệ thành công của kịch bản.

+ Phần 3: Saving - Khi ghi được một kịch bản ổn định, chúng ta có thể lưu lại để chạy hoặc test hồi quy trong tương lai.

2. Tạo một bài Coded UI Test mới từ đầu.

3. Viết mã từ đầu.

2.2.6. Nội dung của một bài Coded UI Test

Khi tạo một bài kiểm thử Coded UI, trình tạo Coded UI Test sẽ tạo một Map. Điều này bao gồm UI được kiểm thử, các phương thức kiểm thử (test methods), các tham số (parameters), các xác nhận (assertions), vv... Đối với mỗi bài kiểm thử, nó cũng tạo ra một tập tin cho lớp. (Bảng 2)

1. **UIMap.Designer.cs:** Trình thiết kế chứa chế

Bảng 2. Danh sách các tập tin khi tạo Coded UI Test

File	Content	Editable
UIMap.Designer.cs	Phần khai báo thuộc tính phương thức lớp UIMap.	No
UIMap.cs	Lớp UIMap (một phần).	Yes
CodedUITest1.cs	Thuộc tính của phương thức lớp CodeUITest1.	Yes
UIMap.uitest	Ánh xạ XML của UI cho kiểm thử. Nó chỉ được chỉnh sửa thông qua trình soạn thảo UIMap.	No

độ xem mã của UIMap. Nó được tạo khi người kiểm thử ghi lại một số tương tác UI hoặc khi một số đối tượng được thêm thủ công vào UIMap.

2. **UIMap.cs:** Bất kỳ sửa đổi hoặc tùy chỉnh nào được thực hiện cho UIMap được lưu trữ trong tập tin này. Ban đầu, tập tin này sẽ trống và có thể được đưa thêm code vào sau. Nếu sửa đổi được thực hiện trực tiếp với tập tin UIMap.designer.cs, hãy đảm bảo nó không được ghi lại nếu không tất cả các thay đổi sẽ bị mất.

3. **CodedUITest1.cs:** Tập tin này chứa lớp của Coded UI Test, các phương thức kiểm thử (test methods), yêu cầu xác nhận (assertion invocation) và gọi phương thức (method invocation). Tất cả các xác nhận và phương thức mặc định được gọi ra từ tập tin này.

4. của lớp UIMap. Nó bao gồm các cửa sổ (windows), điều khiển (controls), thuộc tính (properties), phương thức (methods), hành động (actions) và xác nhận (assertions). Nó chỉ được chỉnh sửa thông qua trình chỉnh sửa UIMap.

2.2.7. Làm thế nào để thực hiện Coded UI Test

Chúng ta phải làm theo các bước sau để thực hiện Coded UI Test.

1. Tạo một dự án Coded UI.
2. Thêm tập tin Coded UI Test.
3. Ghi lại một loạt các hành động.
4. Xác minh các giá trị trong các trường UI như các hộp văn bản (Textbox).
5. Xem mã kiểm thử được tạo ra.
6. Thêm nhiều hành động và xác nhận.
7. Chỉnh sửa các chi tiết của các hoạt động kiểm thử và các xác nhận.
8. Chạy các test case để kiểm thử.

2.2.8. Một số kinh nghiệm cho kiểm thử Coded UI Test

Dưới đây là một số kinh nghiệm cho một bài kiểm thử Coded UI:

- Sử dụng trình tạo kiểm thử Coded UI bất cứ khi nào có thể.
- Cố gắng không sửa đổi tập tin UIMap.designer.cs trực tiếp. Nếu không, việc thay đổi đó sẽ thực hiện đối với tập tin sẽ bị ghi đè.
- Tạo kiểm thử như một chuỗi các phương thức được ghi lại.
- Mỗi phương thức ghi lại phải hoạt động trên một trang đơn (single page), màn hình (window) hoặc hộp thoại (dialog box). Ngoài ra, hãy tạo một phương thức kiểm thử mới cho mỗi trang, màn hình, hoặc hộp thoại mới.
- Khi tạo ra một phương thức, thay vì dùng tên mặc định, chúng ta hãy đặt lại một tên khác cho phương thức có ý nghĩa hơn. Một cái tên có ý nghĩa sẽ giúp xác định mục đích của phương thức.

Nếu có thể, hãy hạn chế mỗi phương thức ghi lại dưới 10 hành động. Cách tiếp cận này khiến dễ dàng hơn để thay thế một phương thức nếu UI thay đổi.

- Để tạo ra sự xác nhận sử dụng trình tạo kiểm thử Coded UI. Nó tự động thêm một phương thức xác nhận vào tập tin UIMap.Designer.cs.
- Ghi lại các phương thức kiểm thử/ các phương thức xác nhận, nếu các giao diện người dùng thay đổi hoặc ghi lại các phần bị ảnh hưởng của một phương thức kiểm thử hiện có.
- Nếu đang code trực tiếp bằng API, sử dụng các phương thức và thuộc tính trong các lớp được tạo ra trong tập tin UIMap.Designer.cs. Các lớp này sẽ làm cho công việc trở nên đáng tin cậy và dễ dàng hơn, giúp làm việc hiệu quả hơn.

2.2.9. So sánh Coded UI Test với Selenium và QTP

Bảng 3 so sánh kiểm thử Coded UI Test với hai công cụ kiểm thử tự động hóa khá phổ biến là Selenium và QTP:

- Selenium là một testing tool mã nguồn mở được tạo bởi Jason Huggins. Sau đó Simon Stewart bắt đầu WebDriver (để khắc phục một số hạn chế của Selenium). Cả hai công cụ này được sáp nhập

để có được một công cụ thử nghiệm tuyệt vời. Selenium không hỗ trợ WPF cũng như các ứng dụng Windows.

- HP QTP (Quick Test Professional): ban đầu được viết bởi Mercury Interactive. Nó là một phần của HP Quality Centre Suite (QC). QTP hỗ trợ ứng dụng trên desktop.

- Coded UI Test (CUIT) đã được giới thiệu bởi Microsoft cùng với Visual Studio 2010. Nó tích hợp với Team Foundation Server. Visual Studio Coded UI hỗ trợ việc kiểm thử ứng dụng lưu trữ Window cũng như kiểm thử ứng dụng Window Phone. Không có công cụ nào khác hỗ trợ điều này.

Bảng 3. So sánh Coded UI Test với Selenium và QTP

Category	Selenium	QTP	Coded UI Test
Dễ dàng với IDE và các tính năng với công cụ			
Ghi và phát lại			
Tích hợp với ALM (Quản lý vòng đời ứng dụng)			
Dễ thực hiện			
Hỗ trợ cho các đối tượng			
Hỗ trợ ngôn ngữ	Perl, PHP, C#, Java, Python, Ruby	VB Script	VB.Net và C#
Hỗ trợ trình duyệt	Hỗ trợ tất cả	Firefox và IE	Vài phiên bản IE
Hỗ trợ cho các ứng dụng khác nhau	Chỉ có web	Gần như tất cả	Gần như tất cả
Note			
Hỗ trợ hoàn toàn	Hỗ trợ một phần		
Không được hỗ trợ	Các tính năng có sẵn nhưng không tương đương		

Đối với ứng dụng trên desktop thì CUIT và QTP thường được đề xuất để lựa chọn.

3. Kết luận

Việc kiểm thử tự động các điều khiển trong ứng dụng thông qua giao diện người dùng (UI) được gọi là kiểm thử Coded UI Test (CUIT).

CUIT hỗ trợ các công nghệ như dịch vụ Web, ứng dụng Window phone, ứng dụng Web, v.v... Nó cho phép các nhà phát triển và người kiểm thử phần mềm cộng tác hiệu quả bằng cách sử dụng cùng các công cụ/ngôn ngữ.

Với CUIT, có thể dễ dàng sử dụng Visual Studio IDE để viết kịch bản. Với Visual Studio CUIT cung cấp một Coded UI Test Builder để ghi lại một loạt các hành động người dùng bằng cách thêm assertions cho UI control. Khi cần thay đổi assertion condition theo yêu cầu (equal to, in between, contains,...) cung cấp các giá trị mong đợi và tạo code cho nó. Sự kiện di chuột có thể được record thủ công nếu cần thiết. CUIT cho phép thực thi lại các kịch bản đã được ghi lại để xác minh và kiểm soát độ ổn định. Các script hỗ trợ với Visual Studio là một cách tuyệt vời và dễ dàng để viết và gỡ lỗi. CUIT có lẽ có ưu thế hơn các công cụ khác trong lĩnh vực này. Đối với ứng dụng trên desktop thì CUIT và QTP được đề xuất để lựa chọn.

Chúng ta cũng có thể chỉnh sửa các UIMap với sự giúp đỡ của Coded UI Editor và tìm hiểu các thuộc tính của đối tượng. Các CUIT có thể được hand code nếu cần. Coded UI Test bao gồm một thư

TÀI LIỆU THAM KHẢO:

1. Trịnh Thị Phương (2017). Kiểm thử tự động và các công cụ được sử dụng trong kiểm thử tự động. *https://viblo.asia/p/kiem-thu-tu-dong-va-cac-cong-cu-duoc-su-dung-trong-kiem-thu-tu-dong-aWj53JqbZ6m*.

2. Nguyễn Thị Phương (2017). Coded UI Test (CUIT). *https://viblo.asia/p/coded-ui-test-cuit-djeZ1BqJlWz*.

3. Lê Thị Ngân (2018). Giới thiệu 35+ công cụ kiểm tra GUI tốt nhất. *https://viblo.asia/p/gioi-thieu-35-cong-cu-kiem-tra-gui-tot-nhat-Az45bgyoKxY*.

4. Guru99. Coded UI Test Automation Framework Tutorial. *https://www.guru99.com/coded-ui-test-cuit.html*

5. SAA solutions (2018). UI Automation trong Visual Studio. *https://saa-solutions.com/student/ui-automation-trong-visual-studio/#1_Cac_the_loai_test*.

6. 30 Best GUI Testing Tools For GUI Test Automation [2021 LIST]. *https://www.softwaretestinghelp.com/best-gui-testing-tools/*

Ngày nhận bài: 19/3/2021
Ngày phản biện đánh giá và sửa chữa: 4/4/2021
Ngày chấp nhận đăng bài: 25/4/2021

Thông tin tác giả:
ThS. TRẦN VĂN THỌ
Khoa Công nghệ thông tin
Trường Đại học Công nghiệp thực phẩm Thành phố Hồ Chí Minh

UI AUTOMATION TESTING ON WINDOWS FORM WITH THE CODED UI TEST

● Master. **TRAN VAN THO**
Faculty of Information Technology
Ho Chi Minh City University of Food Industry

ABSTRACT:

User Interface (UI) Automation is the use of tools to execute test cases, allowing tested data sets to be entered into the testing system, comparing expected results with actual results and generating detailed testing reports. One of the tools for testing UI Automation on Windows Forms is Coded UI Test (CUIT). CUIT uses the Visual Studio IDE to write scripts as logging can be done by using Visual Studio. These tests involve functional testing of controls on UI controls. CUIT tests the functionality of the entire application including the user interface on Windows Forms.

Keywords: UI Automation, UI Automation testing, Windows Form, automated testing, UI automation testing.