

MỤC LỤC

A.GIỚI THIỆU ĐỀ TÀI	2
B.CƠ SỞ LÝ THUYẾT	3
I. LÝ THUYẾT VỀ KIỂM THỬ PHẦN MỀM	3
1.1. Kiểm thử phần mềm là gì ?	3
1.2. Phân loại kỹ thuật kiểm thử	4
1.3. Các cấp độ kiểm thử phần mềm	4
1.4. Quy trình kiểm thử phần mềm	4
II. LÝ THUYẾT VỀ KIỂM THỬ TỰ ĐỘNG	6
2.1. Khái quát về kiểm thử phần mềm tự động	6
2.2. Kiểm thử tự động là gì	6
2.3. Tại sao phải kiểm thử tự động	6
2.4. Nguyên tắc kiểm thử tự động	7
2.5. Quy trình kiểm thử tự động	10
2.6. So sánh kiểm thử tự động và kiểm thử thủ công	10
C.CƠ SỞ THỰC TIỄN	12
I. GIỚI THIỆU CHUNG VỀ PHẦN MỀM TEST COMPLETE	12
1.1. Giới thiệu về Test complete	12
1.2. Lịch sử hình thành	12
1.3. Đặc điểm của Test complete	13
1.4. Cài đặt	14
1.5. Giao diện phần mềm	17
II. HƯỚNG DẪN SỬ DỤNG PHẦN MỀM	20
2.1. Khởi tạo một Dự án test (Create Project)	20
2.2. Ghi lại một bài test (Create a test)	25
2.3. Chạy bài test đã được ghi trước đó (Running the Recorded test)	26
2.4. Sửa chữa các kịch bản test đã ghi.	29
D.KẾT LUẬN	32

A. GIỚI THIỆU ĐỀ TÀI

Hiện nay, sự phát triển mạnh mẽ cũng như bước chuyển mình nhanh chóng của các xu thế công nghệ thông tin trên thế giới đã mang lại cho Việt Nam đồng thời thuận lợi và khó khăn. Do đó, những dự án, chương trình quốc gia nhằm thúc đẩy hiệu quả ứng dụng CNTT trong mọi mặt đời sống kinh tế - chính trị - xã hội đang ngày càng được chú trọng và gấp rút triển khai. Kéo theo đó là nhu cầu về lĩnh vực kiểm thử phần mềm, đặc biệt là kiểm thử phần mềm tự động.

Tại Việt Nam, khái niệm này tuy không mới mẻ song cũng chưa hoàn toàn quen thuộc. Thực tế cho thấy, số lượng đơn vị đào tạo chuyên sâu, các tester chuyên nghiệp về kiểm thử phần mềm không nhiều, chưa thể đáp ứng đủ cho các dự án doanh nghiệp. Nếu xét theo tiêu chuẩn quốc tế, tỷ lệ giữa lập trình viên và tester là 1:3 (cứ 3 lập trình viên thì có 1 tester), đôi khi tỉ lệ này là 1:1 với những dự án đặc thù; thì tại Việt Nam, tỉ lệ đáp ứng được công việc tester chỉ rơi vào khoảng 1.5. Dù biết công tác kiểm thử, đảm bảo chất lượng giữ vai trò quan trọng trong việc mang lại thành công của các dự án phần mềm song không phải công ty nào cũng có đủ chuyên môn và điều kiện cho phép để thực hiện quy trình này.

Tuy nhiên, với những lợi thế cạnh tranh như: nguồn nhân lực rẻ có sẵn trình độ kỹ thuật; đầu tư phát triển cơ sở hạ tầng nhanh; môi trường đầu tư an toàn; chất lượng dịch vụ nổi trội và tỉ lệ thay đổi nhân sự thấp... Việt Nam có thể hi vọng và tin tưởng vào khả năng trở thành đối tác kinh doanh đầy tiềm năng và hấp dẫn trong ngành kiểm thử phần mềm.

Sau quá trình tìm hiểu nhóm quyết định lựa chọn đề tài : “Tìm hiểu công cụ Test Complete” để làm báo cáo kết thúc môn học. Rất mong nhận được ý kiến nhận xét, đóng góp của thầy và các bạn để báo cáo của nhóm được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn !

B. CƠ SỞ LÝ THUYẾT

1. LÝ THUYẾT VỀ KIỂM THỬ PHẦN MỀM

1.1. Kiểm thử phần mềm là gì ?

Kiểm thử phần mềm là quy trình được sử dụng để đánh giá, kiểm tra chất lượng phần mềm ở nhiều khía cạnh khác nhau dựa trên các yêu cầu của người sử dụng đối với sản phẩm phần mềm, nhằm đảm bảo phần mềm hoạt động tốt trong các môi trường, trường hợp khác nhau.

Kiểm thử phần mềm là một cuộc kiểm tra được tiến hành để cung cấp cho các bên liên quan thông tin về chất lượng của sản phẩm hoặc dịch vụ được kiểm thử. Kiểm thử có thể cung cấp cho doanh nghiệp một quan điểm, một cách nhìn độc lập về phần mềm để từ đó cho phép đánh giá và thấu hiểu được những rủi ro trong quá trình triển khai phần mềm.

Trong kỹ thuật kiểm thử không chỉ giới hạn ở việc thực hiện một chương trình hoặc ứng dụng với mục đích đi tìm các lỗi phần mềm (bao gồm các lỗi và các thiếu sót) mà còn là một quá trình phê chuẩn và xác minh một chương trình máy tính / ứng dụng / sản phẩm nhằm:

- Đáp ứng được mọi yêu cầu hướng dẫn khi thiết kế và phát triển phần mềm.
- Thực hiện công việc đúng như kỳ vọng.
- Có thể triển khai được với những đặc tính tương tự.
- Và đáp ứng được mọi nhu cầu của các bên liên quan.

Tùy thuộc vào từng phương pháp, việc kiểm thử có thể được thực hiện bất cứ lúc nào trong quá trình phát triển phần mềm. Theo truyền thống thì các nỗ lực kiểm thử được tiến hành sau khi các yêu cầu được xác định và việc lập trình được hoàn tất nhưng trong Agile (là một tập hợp các phương pháp phát triển phần mềm linh hoạt dựa trên việc lặp đi lặp lại và gia tăng giá trị) thì việc kiểm thử được tiến hành liên tục trong suốt quá trình xây dựng phần mềm. Như vậy, mỗi một phương pháp kiểm thử bị chi phối theo một quy trình phát triển phần mềm nhất định.

1.2. Phân loại kỹ thuật kiểm thử

Ta phân loại kiểm thử dựa vào yếu tố: Chiến lược kiểm thử, phương pháp kiểm thử và kỹ thuật kiểm thử.

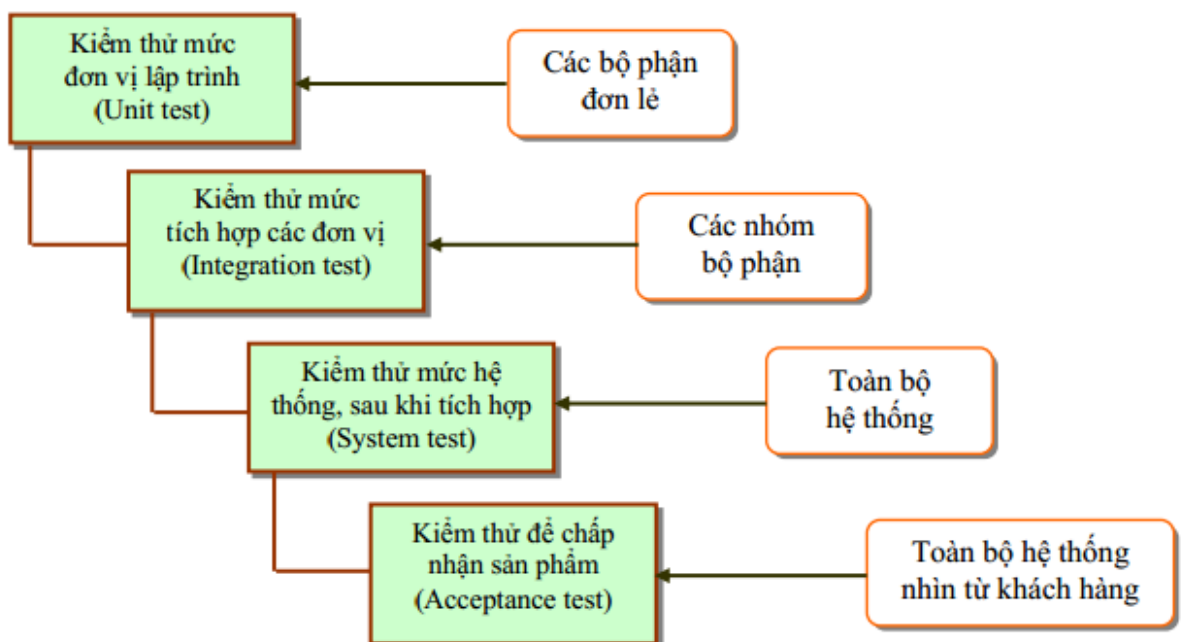
- Dựa vào chiến lược kiểm thử ta có thể phân chia kiểm thử thành 2 loại: kiểm thử thủ công và kiểm thử tự động

- Theo phương pháp tiến hành kiểm thử ta chia kiểm thử thành 2 loại: Kiểm thử tĩnh và kiểm thử động.

- Dựa vào kỹ thuật kiểm thử ta có thể phân chia kiểm thử thành 3 loại: Kiểm thử hộp đen, kiểm thử hộp trắng, kiểm thử hộp xám.

1.3. Các cấp độ kiểm thử phần mềm

Thực tế, KTPM không đơn giản như nhiều người thường nghĩ, công việc này có nhiều mức độ khác nhau và có mối tương quan với các chặng phát triển trong dự án PTPM. Trong một dự án kiểm thử phần mềm bao gồm 4 mức độ cơ bản: Kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống và kiểm thử chấp nhận.



Hình 1.1- Bốn cấp độ cơ bản của kiểm thử phần mềm

1.4. Quy trình kiểm thử phần mềm

Dẫu cho các biến thể tồn tại giữa các tổ chức lập trình thì vẫn có một quy trình điển hình để kiểm thử. Mẫu dưới đây là phổ biến trong các tổ chức sử dụng

mô hình phát triển Waterfall (thác nước). Các hoạt động tương tự thường được tìm thấy trong các mô hình phát triển khác, nhưng có thể có hoặc không rõ ràng.

- Phân tích yêu cầu: Kiểm thử thường sẽ bắt đầu lấy các yêu cầu trong các giai đoạn của vòng đời phát triển phần mềm. Trong giai đoạn thiết kế, các Tester làm việc với các nhà phát triển để xác định những khía cạnh của một thiết kế được kiểm chứng và những thông số được kiểm tra.

- Lập kế hoạch kiểm thử: Chiến lược kiểm thử, kế hoạch kiểm thử, kiểm thử sáng tạo... Và có một kế hoạch là cần thiết vì nhiều hoạt động sẽ được thực hiện trong thời gian kiểm thử.

- Kiểm thử phát triển: Các quy trình kiểm thử, các kịch bản, Test Case, các dữ liệu được sử dụng trong kiểm thử phần mềm.

- Kiểm thử thực hiện: Dựa trên các kế hoạch, các văn bản kiểm thử và các báo cáo bất kỳ lỗi nào tìm thấy cho nhóm phát triển.

- Kiểm thử báo cáo: Sau khi hoàn tất kiểm thử, các Tester tạo ra các số liệu và báo cáo cuối cùng về nỗ lực kiểm thử của họ và có sẵn sàng phát hành phần mềm hay không.

- Phân tích kết quả kiểm thử hoặc phân tích thiếu sót được thực hiện bởi đội ngũ phát triển kết hợp với khách hàng để đưa ra quyết định xem những thiếu sót gì cần phải được chuyển giao, cố định và từ bỏ (tức là tìm ra được phần mềm hoạt động chính xác) hoặc giải quyết sau.

- Test lại khiếm khuyết: Khi một khiếm khuyết đã được xử lý bởi đội ngũ phát triển, nó phải được kiểm tra lại bởi nhóm kiểm thử.

- Kiểm thử hồi quy: Người ta thường xây dựng một chương trình kiểm thử nhỏ là tập hợp của các bài kiểm tra cho mỗi tích hợp mới, sửa chữa hoặc cố định phần mềm, để đảm bảo rằng những cung cấp mới nhất đã không phá hủy bất cứ điều gì và toàn bộ phần mềm vẫn còn hoạt động một cách chính xác.

- Kiểm thử đóng gói: Mỗi phép thử thỏa mãn các chỉ tiêu truy xuất và thu được những kết quả quan trọng như: bài học kinh nghiệm, kết quả, các bản ghi, tài liệu liên quan được lưu trữ và sử dụng như một tài liệu tham khảo cho các dự án trong tương lai.

II. LÝ THUYẾT VỀ KIỂM THỬ TỰ ĐỘNG

2.1. Khái quát về kiểm thử phần mềm tự động

Kiểm thử phần mềm tốn nhiều chi phí nhân công, thời gian. Trong một số dự án, chi phí kiểm thử phần mềm chiếm 40% tổng giá trị của dự án. Nếu cần ứng dụng an toàn hơn, chi phí kiểm thử còn cao hơn nữa. Do đó một trong các mục tiêu của kiểm thử là tự động hóa nhiều, nhờ đó mà giảm thiểu chi phí, giảm lỗi, đặc biệt giúp việc kiểm thử hồi qui dễ dàng và nhanh chóng hơn. Tự động hóa việc kiểm thử là dùng phần mềm điều khiển việc thi hành kiểm thử, so sánh kết quả có được với kết quả mong muốn, thiết lập các điều kiện đầu vào, các kiểm soát kiểm thử và các chức năng báo cáo kết quả...

2.2. Kiểm thử tự động là gì

Kiểm thử tự động là quá trình thực hiện một cách tự động các bước trong một kịch bản kiểm thử. Kiểm thử tự động bằng một công cụ nhằm rút ngắn thời gian kiểm thử.

2.3. Tại sao phải kiểm thử tự động

Kiểm thử phần mềm tự động với mục đích:

- Giảm bớt công sức và thời gian thực hiện quá trình kiểm thử
- Tăng độ tin cậy.
- Giảm sự nhầm chán cho con người
- Rèn luyện kỹ năng lập trình cho kiểm thử viên
- Giảm chi phí cho tổng quá trình kiểm thử.

Khi nào cần kiểm thử tự động:

- Không đủ tài nguyên: Khi số lượng TestCase quá nhiều mà kiểm thử viên không thể hoàn tất trong thời gian cụ thể
- Kiểm tra hồi quy: Nâng cấp phần mềm, kiểm tra lại các tính năng đã chạy tốt và những tính năng đã sửa. Tuy nhiên, việc này khó đảm bảo về mặt thời gian
- Kiểm tra khả năng vận hành phần mềm trong môi trường đặc biệt:
 - Đo tốc độ trung bình xử lý một yêu cầu của Web server.
 - Xác định số yêu cầu tối đa được xử lý bởi Web Server .

- Xác định cấu hình máy thấp nhất mà PM vẫn có thể hoạt động tốt.

2.4. Nguyên tắc kiểm thử tự động

Thực sự là sai lầm khi nghĩ tự động là đơn giản chụp lại, ghi lại 1 tiến trình kiểm thử thủ công. Thực tế, kiểm thử tự động có những điểm khác với kiểm thử thủ công. Nó có những lỗi và khả năng dự đoán.

Vì thế, những cơ hội thành công với kiểm thử tự động sẽ được cải thiện đáng kể trong trường hợp bạn thực sự hiểu nó.

Test	Automation
Application expertise	Development expertise
What to test	How to automate
Test Cases	Test scripts

Kiểm thử tự động tuân theo đầy đủ những nguyên tắc kiểm thử nói chung, đó là các nguyên tắc sau:

❖ Nguyên tắc 1 – Kiểm thử đưa ra lỗi

Kiểm thử có thể cho thấy rằng phần mềm đang có lỗi, nhưng không thể chứng minh rằng phần mềm không có lỗi. Kiểm thử làm giảm xác suất lỗi chưa tìm thấy vẫn còn trong phần mềm, thậm chí là không còn lỗi nào, nó không phải là bằng chứng của sự chính xác.

❖ Nguyên tắc 2 – Kiểm thử mọi thứ là không thể

Kiểm thử mọi thứ (tất cả các tổ hợp của điều kiện input đầu vào) là không thể thực hiện được, trừ phi nó chỉ bao gồm một số trường hợp bình thường (ít trường hợp tổ hợp thì có thể test toàn bộ được). Thay vì kiểm thử toàn bộ, việc phân tích rủi ro và dựa trên sự mức độ ưu tiên chúng ta có thể tập trung việc kiểm thử vào một số điểm cần thiết.

❖ Nguyên tắc 3 – Kiểm thử sớm

Để tìm được bug sớm, các hoạt động kiểm thử nên được bắt đầu càng sớm càng tốt trong qui trình phát triển (vòng đời phát triển) phần mềm hoặc hệ thống, và nên tập trung vào các hoạt động đã định trước.

❖ Nguyên tắc 4 – Sự tập trung của lỗi

Nỗ lực kiểm thử nên tập trung một cách cân đối vào mật độ lỗi dự kiến và lỗi phát hiện ra sau đó trong các mô-đun. Một số ít các mô-đun thường chứa nhiều lỗi không phát hiện ra trong lúc kiểm thử trước khi phát hành (release), hoặc chịu trách nhiệm cho hầu hết các lỗi hoạt động của phần mềm.

❖ Nguyên tắc 5 – Nghịch lý thuốc trừ sâu

Nếu việc kiểm thử tương tự nhau được lặp đi lặp lại nhiều lần, thì cuối cùng sẽ có một số trường hợp kiểm thử (ca kiểm thử - test case) sẽ không còn tìm thấy bất kỳ lỗi nào mới. Để khắc phục "nghịch lý thuốc trừ sâu" này, các trường hợp kiểm thử cần phải được xem xét và sửa đổi thường xuyên, và cần phải viết các test case mới và khác nhau để thực hiện nhiều phần khác nhau của phần mềm hoặc hệ thống để tìm ra lỗi tiềm ẩn nhiều hơn nữa.

Nguyên tắc này giống như việc trừ sâu trong nông nghiệp, nếu chúng ta cứ phun một loại thuốc với nồng độ giống nhau trong một khoảng thời gian dài thì có một số con sâu sẽ quen dần và cuối cùng việc phun thuốc giống như là tắm chúng vậy (bị lờn thuốc) => lúc đó chúng ta không thể diệt sạch chúng được. Do vậy, để diệt sạch sâu một cách hiệu quả, người ta thường thay đổi loại thuốc trừ sâu, mỗi loại chỉ dùng trong khoảng thời gian ngắn.

❖ Nguyên tắc 6 – Kiểm thử theo các ngữ cảnh độc lập

Nguyên tắc này là việc testing phụ thuộc vào ngữ cảnh, test trong nhiều ngữ cảnh khác nhau.

Để hiểu rõ hơn chúng ta xem ví dụ sau:

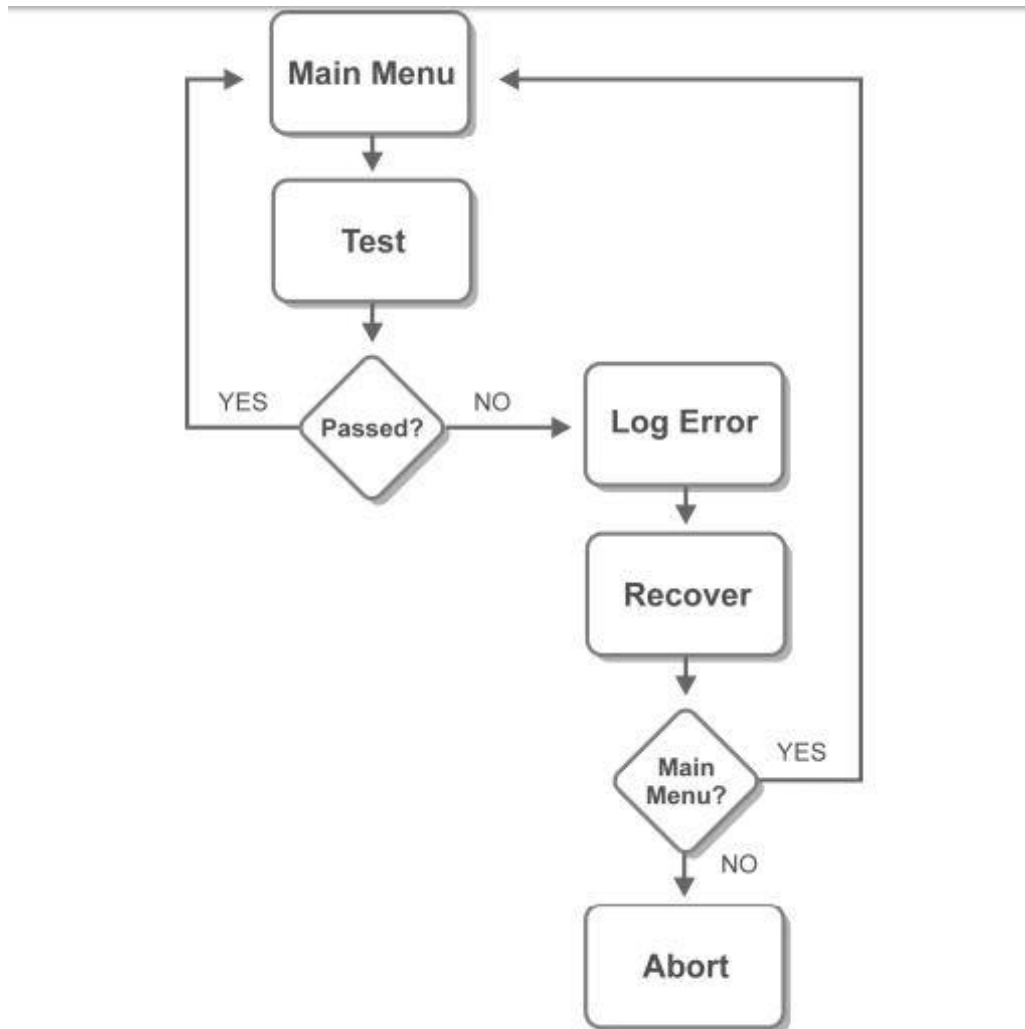
Ví dụ cũng với một chương trình calculator có rất nhiều chức năng, nhưng:

- Nếu test chương trình này cho mẫu giáo thì chỉ cần test cộng trừ là OK.
- Nếu test chương trình này cho cấp 2 thì cộng trừ nhân chia.
- Nếu test chương trình này cho đại học thì tích phân, đạo hàm, v.v....

❖ Nguyên tắc 7 – Sự sai lầm về việc không có lỗi

Việc tìm và sửa chữa lỗi sẽ không giúp được gì nếu hệ thống được xây dựng xong nhưng không thể dùng được và không đáp ứng được nhu cầu và sự mong đợi của người dùng. (Nghĩa là nếu sau khi code, test rồi fix bug, làm đủ tất cả các trường hợp và cuối cùng cho ra một sản phẩm không như mong đợi hoặc

không đáp ứng được nhu cầu của khách hàng thì dự án phần mềm đó coi như thất bại mặc dù đã được test xong).



Hình 1: Tối ưu hóa trong kiểm thử tự động

2.5. Quy trình kiểm thử tự động

Trong 1 môi trường lí tưởng thì kiểm thử sẽ song song với chu trình phát triển của 1 ứng dụng:

Software

Planning	Requirements	Design	Code	Test	Maintain
----------	--------------	--------	------	------	----------

Testware

Test Plan	Test Cases	Test Scripts	Test Execution/Maintenance
-----------	------------	--------------	----------------------------

Hình 2: quy trình kiểm thử tự động

STT	Bước thực hiện	Mô tả
1	Tạo kịch bản kiểm thử	Giai đoạn này dùng công cụ kiểm thử để ghi lại các thao tác lên phần mềm cần kiểm tra và tự động sinh ra kịch bản kiểm thử
2	Chỉnh sửa kịch bản	Chỉnh sửa để kịch bản kiểm thử thực hiện kiểm tra theo đúng yêu cầu đặt ra. Cụ thể, làm theo trường hợp kiểm thử cần thực hiện
3	Chạy kịch bản kiểm thử	Chạy kịch bản kiểm thử để kiểm tra phần mềm có đưa ra đúng như kết quả mong muốn không
4	Đánh giá kết quả	Đánh giá kết quả sau khi chạy kịch bản kiểm thử.

2.6. So sánh kiểm thử tự động và kiểm thử thủ công

Tiêu chí	Kiểm thử thủ công	Kiểm thử tự động
Thời gian	Mất nhiều thời gian thực thi nhưng không phải test lặp đi lặp lại.	Mất ít thời gian thực thi nhưng quá trình test lặp tăng hơn nhiều so với kiểm thử thủ công
Độ linh động	Linh động do kiểm thử thủ công nên có thể phát hiện và xử lý những tình huống phát sinh trong quá trình test. Và có thể tìm ra lỗi mới	Không linh động vì kiểm thử theo script, Kiểm thử hiệu năng và tải trọng nên quá trình test không phát hiện ra lỗi mới. Chỉ thích hợp với kiểm thử hồi quy.

Phụ thuộc	Phụ thuộc vào trạng thái của con người nên kết quả test có thể kém chính xác đối với dự án lớn có nhiều testcase	Nhất quán, nên kết quả test là chính xác và không phụ thuộc vào yếu tố ngoại cảnh
Bảo trì	Không cần bảo trì	Cần bảo trì
Kết quả	Có kết quả ngay lập tức	Cần 1 thời gian mới có kết quả
Ưu điểm	Kiểm thử linh hoạt và trong quá trình test sẽ tìm đc ra lỗi mới	Kiểm thử tự động thích hợp cho việc kiểm thử lặp đi lặp lại, có thể tái sử dụng testCriKiểm thử hiệu năng và tải trọng . Thích hợp giả lập test hiệu năng, chịu tải cũng như giả lập hệ thống kiểm thử
Hạn chế	Nếu sử dụng kiểm thử thủ công mà kiểm thử 1 chức năng lặp đi lặp lại thì sẽ tốn nhiều thời gian và sẽ khó chính xác. Nên thay thế bằng kiểm thử tự động để đỡ mất thời gian giám sát, tối ưu hóa việc sử dụng tài nguyên máy tính để kiểm thử.	Nếu sử dụng kiểm thử tự động mà kiểm thử ít sẽ rất lãng phí thời gian và nhân lực và công việc viết testScript Kiểm thử hiệu năng và tải trọng , trong trường hợp này thì nên thực hiện kiểm thử thủ công

C. CƠ SỞ THỰC TIỄN

1. GIỚI THIỆU CHUNG VỀ PHẦN MỀM TEST COMPLETE

1.1. Giới thiệu về Test complete

TestComplete là một môi trường kiểm thử tự động cho một loạt các loại ứng dụng và công nghệ, bao gồm (nhưng không giới hạn) Windows, .NET, WPF, Visual C++, Visual Basic, Delphi, C++ Builder, Java và các ứng dụng Web và dịch vụ.

TestComplete được định hướng như nhau đối với chức năng kiểm thử, đơn vị. Nó cung cấp hỗ trợ cho các thử nghiệm hồi quy hàng ngày và hỗ trợ nhiều loại thử nghiệm: thử nghiệm dữ liệu điều khiển, kiểm thử đối tượng điều khiển, và những người khác.

Bạn tạo ra các bài kiểm thử bằng cách ghi lại chúng hoặc lệnh kiểm thử chỉnh sửa trong bảng và biên tập viên của TestComplete. Kiểm thử có thể được chạy từ bên trong TestComplete hoặc họ có thể được xuất khẩu sang một ứng dụng bên ngoài và chạy đó.

TestComplete nhận đối tượng và điều khiển trong các ứng dụng thử nghiệm và cung cấp các lệnh đặc biệt để mô phỏng hành động sử dụng với họ. Nó cũng cung cấp các trạm kiểm soát cụ thể, cho phép bạn dễ dàng kiểm thử trạng thái ứng dụng trong thời gian chạy thử nghiệm.

TestComplete hiện nay được sử dụng bởi hơn 5000 công ty.

1.2. Lịch sử hình thành

TestComplete được phát triển đầu tiên vào năm 1999 bởi công ty AutomatedQA với tên Aqtest. Từ đó cho đến năm 2012, TestComplete trải qua nhiều phiên bản khác nhau. Phiên bản hiện tại là TestComplete 9.31.

Các phiên bản trải qua:

- Aqtest 1.x (1.01; 1.5)
- TestComplete 2.x (2.0; 2.02; 2.03; 2.04)
- TestComplete 3.x (3.0; 3.01; 3.02; 3.03; 3.04; 3.05; 3.06; 3.07; 3.08; 3.09; 3.10)

- TestComplete 4.x (4.0; 4.10; 4.20; 4.21; 4.22;4.23; 4.24; 4.25; 4.26; 4.27; 4.28; 4.29; 4.30)
- TestComplete 5.x (5.0; 5.1; 5.11; 5.12; 5.13; 5.14)
- TestComplete 6.x (6.0; 6.10; 6.11; 6.12; 6.20; 6.30; 6.40; 6.50; 6.51; 6.52)
- TestComplete 7.x (7.0; 7.10; 7.20; 7.50; 7.51; 7.52)
- TestComplete 8.x (8.0; 8.10; 8.20; 8.50; 8.60; 8.70)
- TestComplete 9.x (9.0; 9.10; 9.20; 9.30; 9.31)

1.3.Đặc điểm của Test complete

❖ Các tính năng chính

- Keyword Testing: Kiểm tra từ khóa
- Full-Featured Script Editor: Chỉnh sửa đầy đủ các kịch bản
- Test Record and Playback: Cho phép ghi và chạy lại quá trình test
- Script Debugging Features: Gỡ lỗi
- Access to Methods and Properties of Internal Objects : Truy cập đến các phương thức và thuộc tính của bên trong đối tượng
- Unicode Support: Hỗ trợ bộ gõ Unicode
- Issue-Tracking Support

❖ Các dạng testing được hỗ trợ

- Functional (or GUI) Testing: Kiểm tra hàm
- Regression testing: Kiểm tra hồi quy
- Unit testing: Kiểm tra đơn vị
- Distributed Testing: Kiểm tra phân tán
- Load Testing: Kiểm tra truyền tải
- Web Testing: Kiểm tra trên nền Web
- Functional and load testing of web services: Kiểm tra các hàm và truyền tải của dịch vụ Web
- Coverage Testing
- Data-Driven Testing

- Manual Testing: Kiểm tra bằng tay
- Keyword testing: Kiểm tra từ khóa

❖ Các ngôn ngữ viết mã hỗ trợ

- VBScript
- JScript
- DelphiScript
- C++Script
- C#Script

1.4.Cài đặt

1.4.1. Yêu cầu hệ thống

❖ Yêu cầu hệ điều hành:

- Microsoft Windows 8
- Microsoft Windows Server 2012.
- Microsoft Windows 7
- Microsoft Windows Vista
- Microsoft Windows Server 2008
- Microsoft Windows Server 2008 R2.
- Microsoft Windows Server 2003
- Microsoft Windows XP 32-bit SP3 and XP 64-bit SP2.
- Microsoft Windows Embedded Standard 7
- Microsoft Windows Embedded Standard 2009 (with the Add Hardware Control Panel component installed).

Yêu cầu cấu hình:

Cấu hình tối thiểu	Cấu hình đề nghị
<ul style="list-style-type: none">• Intel Pentium II 400 MHz or higher.• 256 MB of RAM.• 700 MB of hard disk space.• 1024×768 or higher resolution monitor.• Mouse or other pointing device.• Microsoft Internet Explorer 7.0 or later.	<ul style="list-style-type: none">• Intel Pentium 4 3 GHz, Intel Core 2 Duo 2 GHz or higher.• 1 GB of RAM on Windows XP and Windows Server 2003. 2 GB of RAM on Windows Vista, Windows 7 and later operating systems.• 700 MB hard disk space.• 1280×1024 or higher resolution monitor.• Mouse or other pointing device.• Microsoft Internet Explorer 7.0 or later.

❖ Yêu cầu bổ sung:

- .NET Application Testing

Microsoft .NET Framework v. 2.0, 3.0, 3.5, 4.0 or 4.5 is required.

- Web Testing

Internet Explorer ver. 7–11

Mozilla Firefox ver. 18–25

Google Chrome 27-32

Opera Software Opera ver. 12.10 - 12.15

Apple Safari for Windows ver. 5.

1.4.2. Cài đặt

Link download: <http://smartbear.com/products/qa-tools/automated-testing-tools/free-testcomplete-trial/> đăng ký rồi tải về phiên bản dùng thử 30 ngày.

Hoặc mua bản quyền với giá 4999\$.

The screenshot shows a web browser window with the URL `smartbear.com/products/qa-tools/automated-testing-tools/free-testcomplete-trial/`. The page features the SmartBear logo and a navigation menu with links to Products, Resources, Support, Community, About Us, News & Events, and Blog. The main heading is "Download Your Free TestComplete Trial".

The registration form includes the following fields:

- Work Email *
- First Name *
- Last Name *
- Company *
- Country * (Please Choose...)
- Phone Number * (Example: 231-231-2312)

A green "Sign Up Now" button is located below the form. Below the button, a small red text line reads: "We respect your Privacy. And of course you can opt-out at any time you wish."

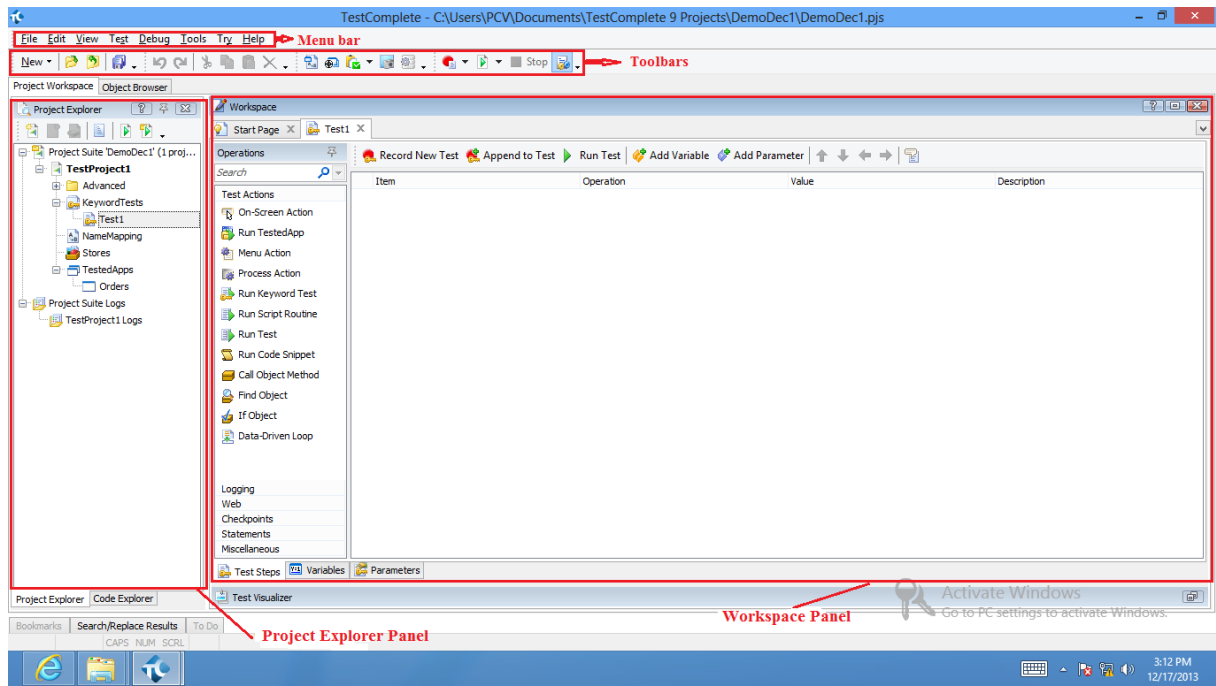
To the right of the form, there is an "FAQs" section with two questions:

- Do trials require a valid email address?**
Yes they do. We email you a license key and/or login information so a valid email address is needed.
- If I buy the software will I have to re-install it?**
In most cases the software does not need to be reinstalled or reconfigured once purchased.

Below the FAQs, there is a testimonial from Paul Wensley - Terragrica, Inc., which is partially obscured by a large quotation mark graphic.

1.5. Giao diện phần mềm

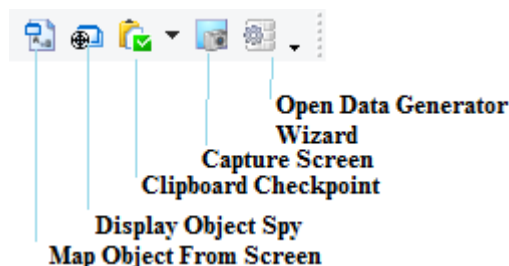
❖ Giao diện chính



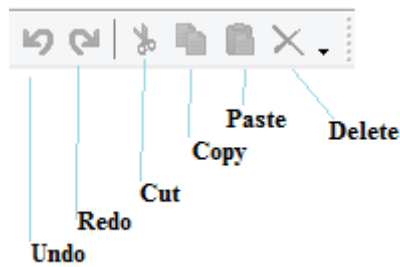
Menubar	Hiển thị những menu của công cụ
Toolbar	Chứa những buttons để giúp bạn quản lý test của bạn
Project Explorer Panel	Sử dụng để hiển thị và thay đổi cấu trúc của các dự án TestComplete và dãy dự án. Nó cũng hiển thị cấu trúc của các bản ghi của dự án và chạy thử nghiệm.
Workspace Panel	Là khu vực làm việc chính của bạn trong TestComplete và giữ chỗ cho các biên tập viên và các tấm cho phép bạn xem và chỉnh sửa các nội dung của dự án, hạng mục công trình và các bản ghi đăng nhập

❖ Giới thiệu 1 số thanh công cụ trên giao diện chính

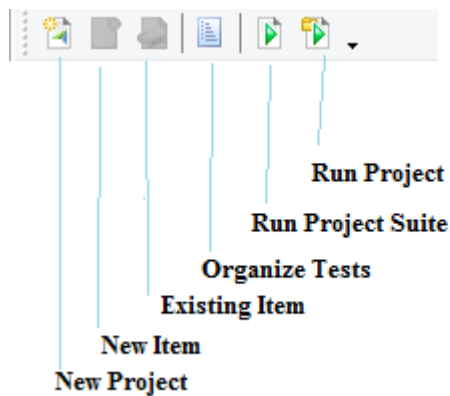
Tools: cung cấp các lệnh có ảnh hưởng đến quá trình ghi âm



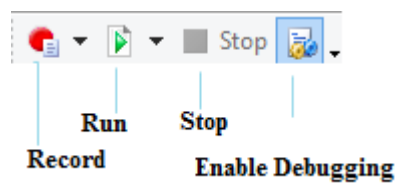
Edit: cung cấp các lệnh chỉnh sửa tiêu chuẩn



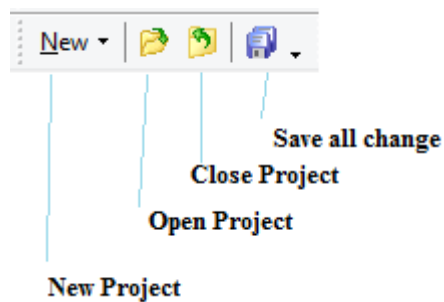
Project Explorer : Cung cấp các lệnh ảnh hưởng tới Project Explorer Panel



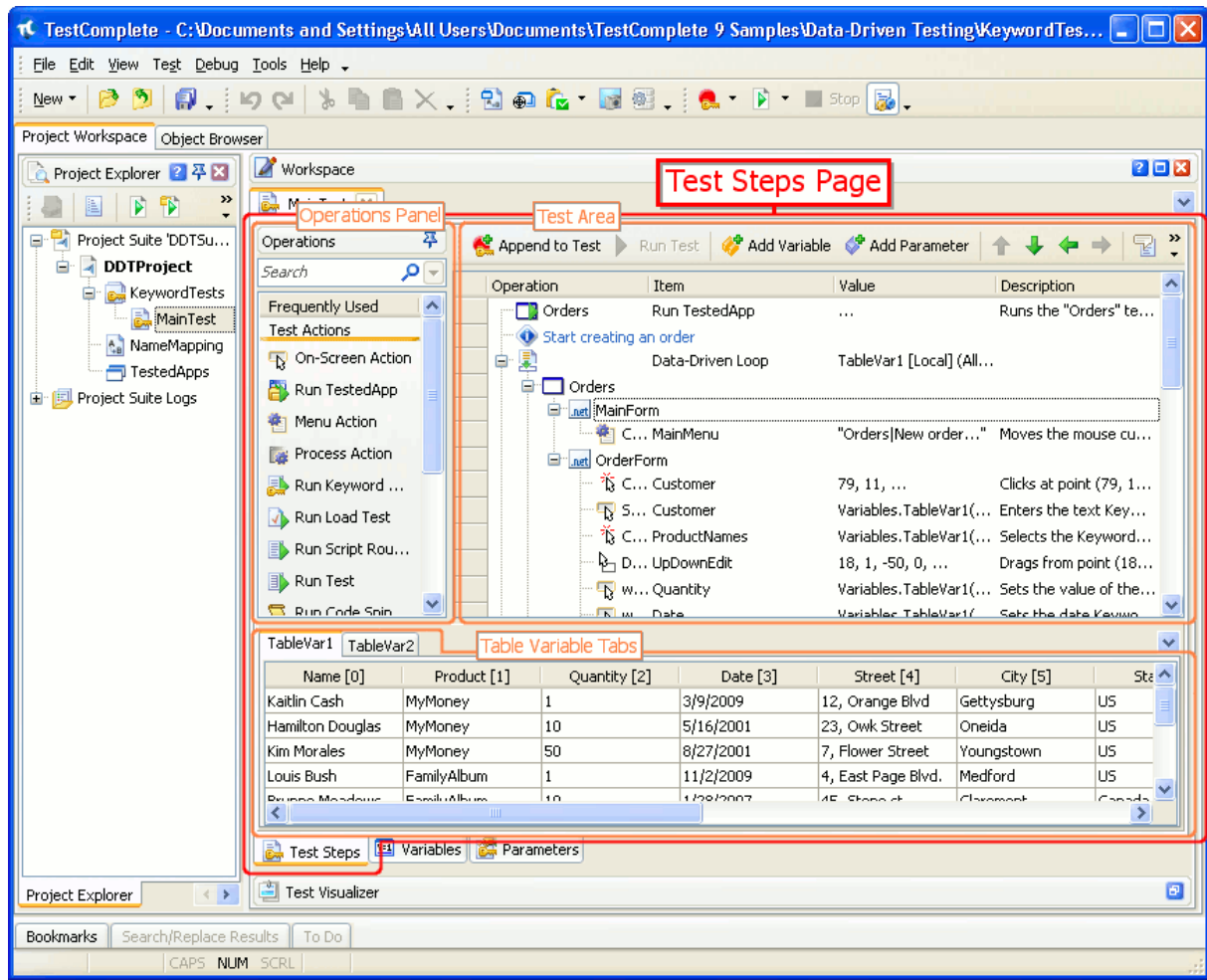
Test Engine: Cung cấp các lệnh có ảnh hưởng ghi lại và gỡ lỗi kiểm thử



Standard: cung cấp các lệnh có ảnh hưởng đến cấu trúc của các bộ dự án hiện tại .



❖ Giao diện Test Steps Page



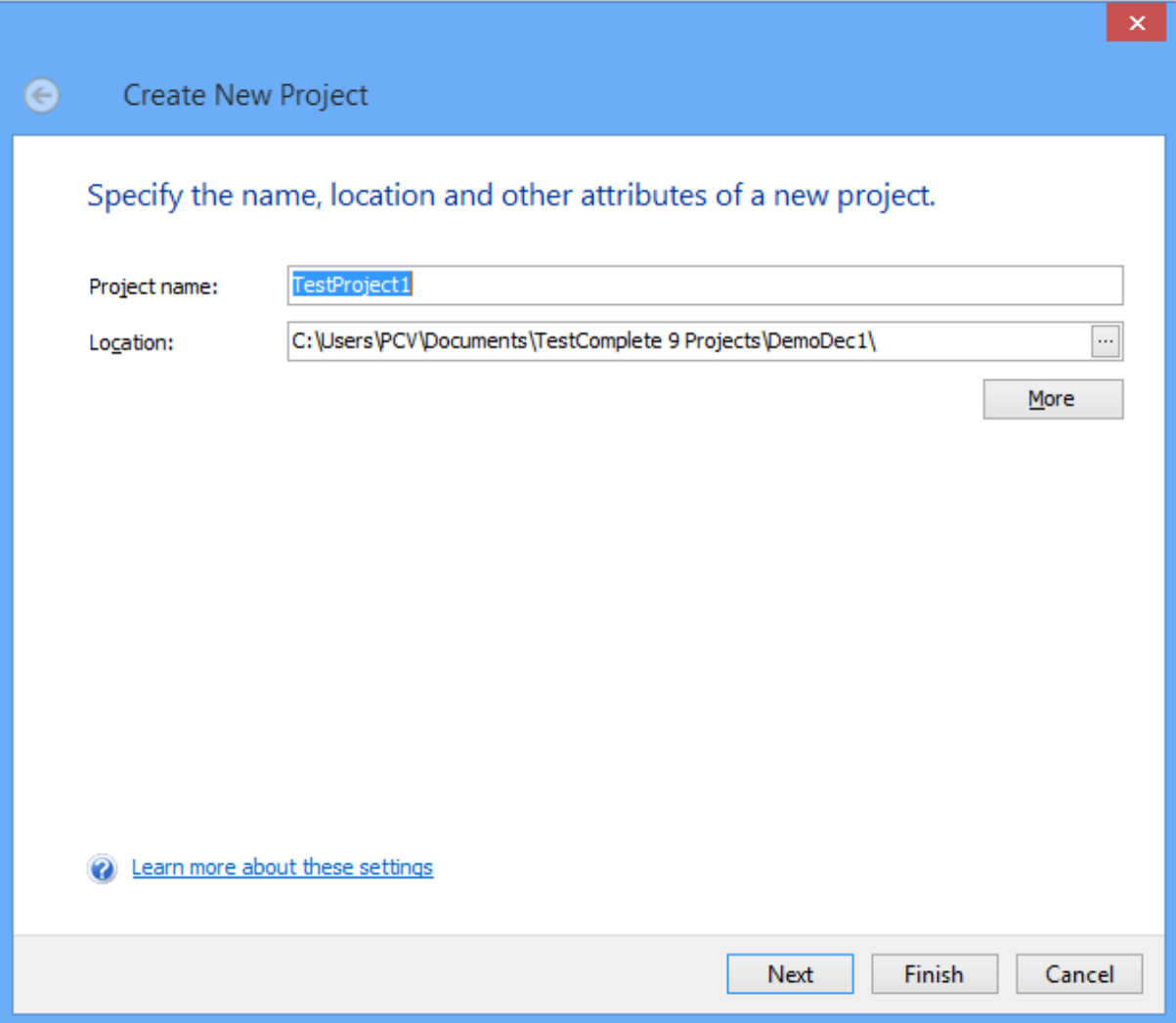
Operations Panel	Liệt kê tất cả các hoạt động bạn có thể thêm vào các bài kiểm tra của bạn
Test Area	Chứa các hoạt động tạo nên kiểm thử từ khóa hiện tại .
Table Variable Tabs	Hiển thị nội dung của bảng biến kiểm thử từ khóa của

II. HƯỚNG DẪN SỬ DỤNG PHẦN MỀM

2.1. Khởi tạo một Dự án test (Create Project)

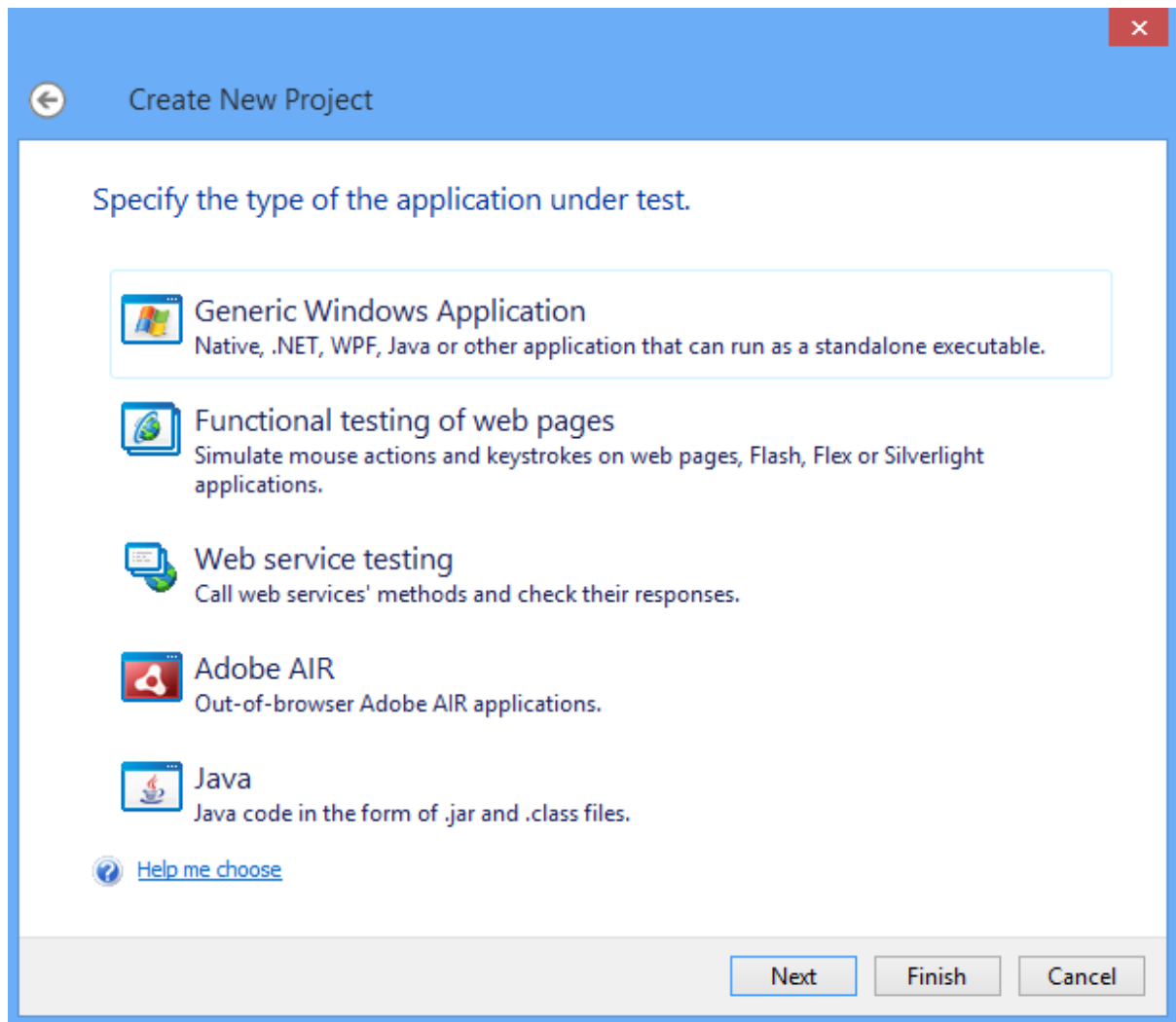
Để khởi tạo một dự án Kiểm thử ta thực hiện các bước như sau:

- Nếu bạn đang mở một Project bạn phải thoát khỏi Project đó. Để làm như thế ta thực hiện: Chọn File | Close từ giao diện chính của TestComplete
- Chọn File | New | New Project từ giao diện chính. Hộp thoại mới sẽ xuất hiện:



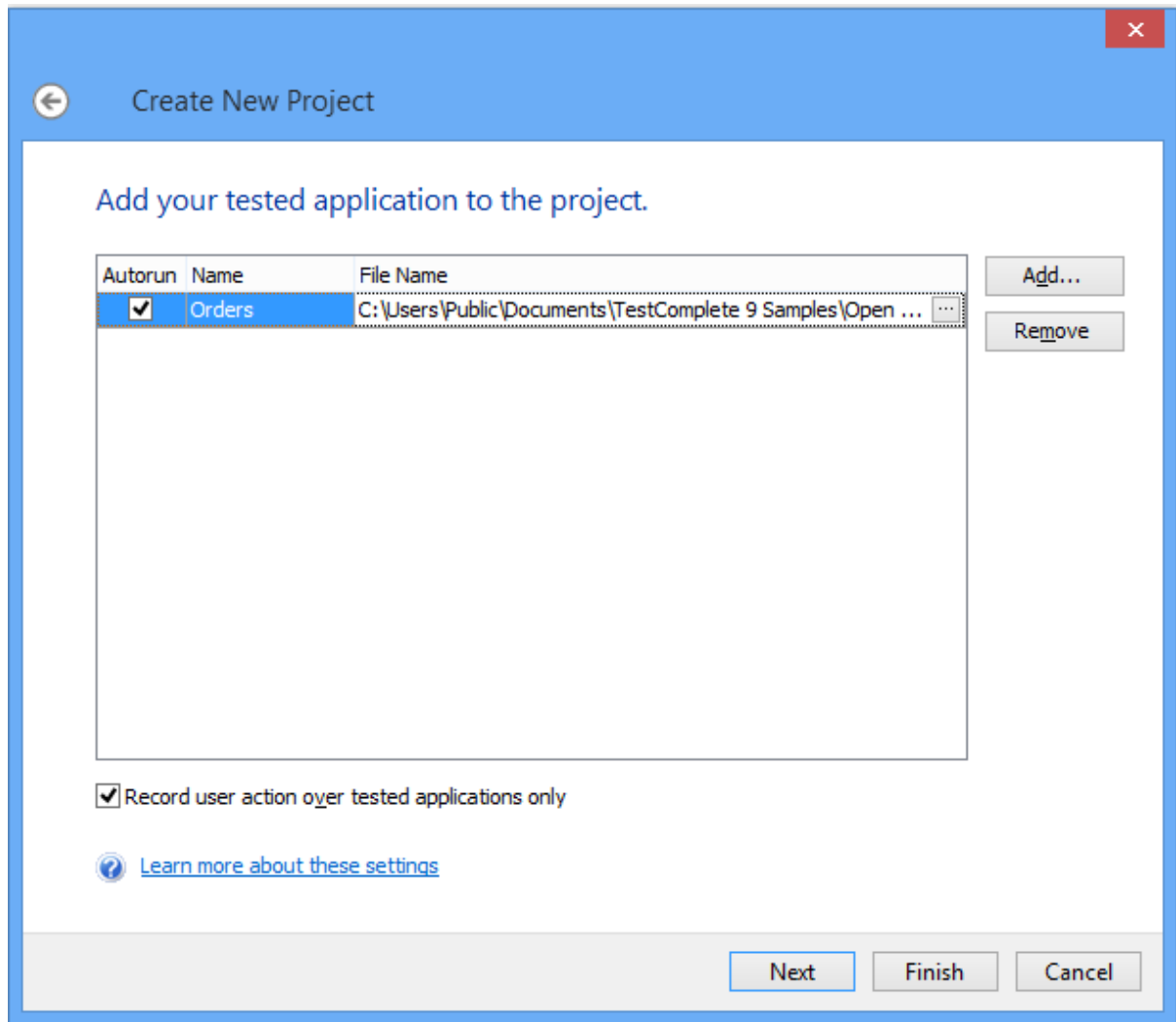
- Cho phép Lưu tên dự án và Nơi lưu trữ dự án của bạn.
- Chọn Next để đến bước tiếp theo.

- Hộp thoại cho phép bạn chọn kiểu Test và Môi trường của môi trường phần mềm được Test. <Trong hướng dẫn này sẽ chọn Generic Windows Application>



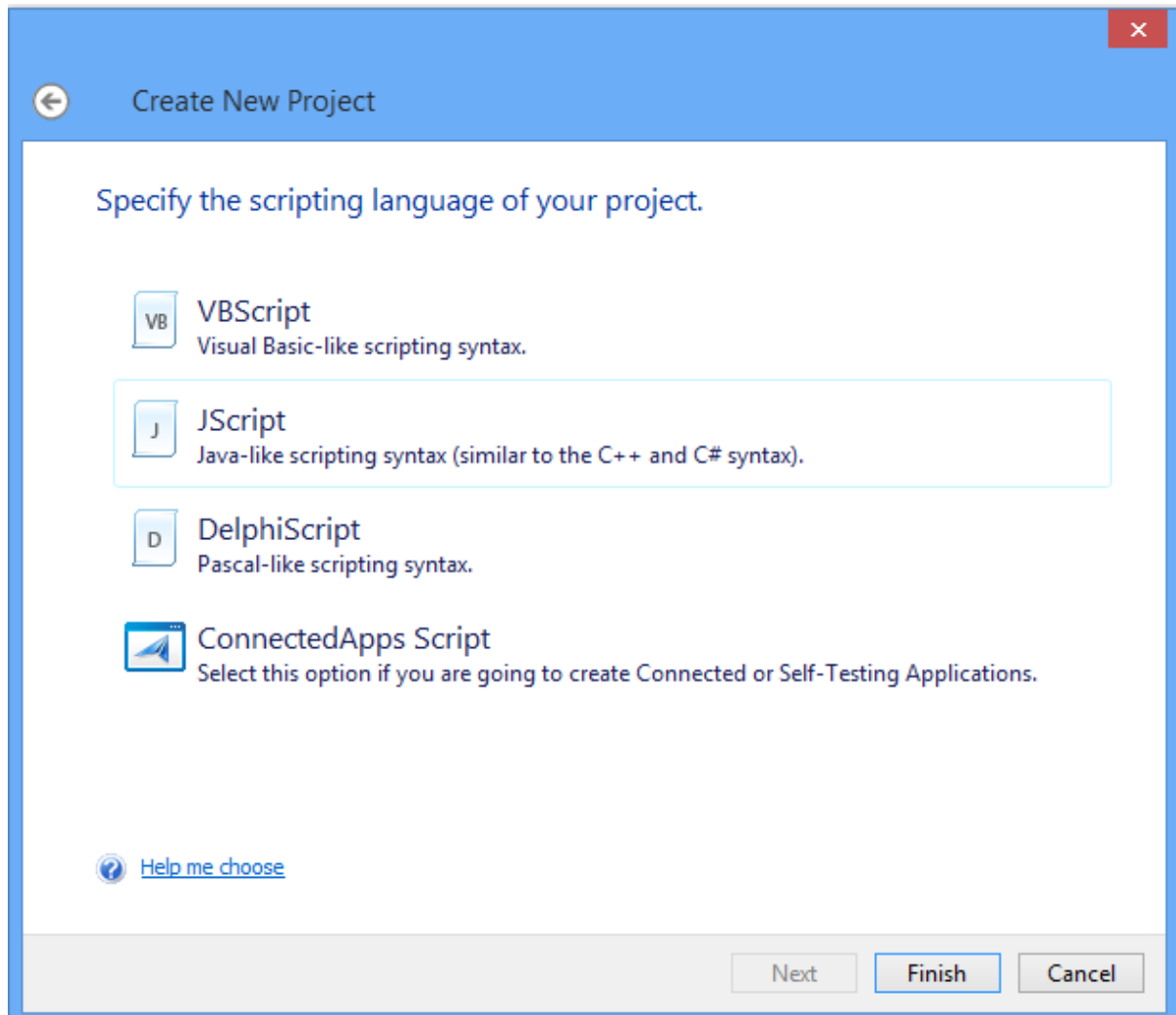
- Chọn Next để tiếp tục.

- Hộp thoại yêu cầu bạn Add đường dẫn tới ứng dụng cần test.

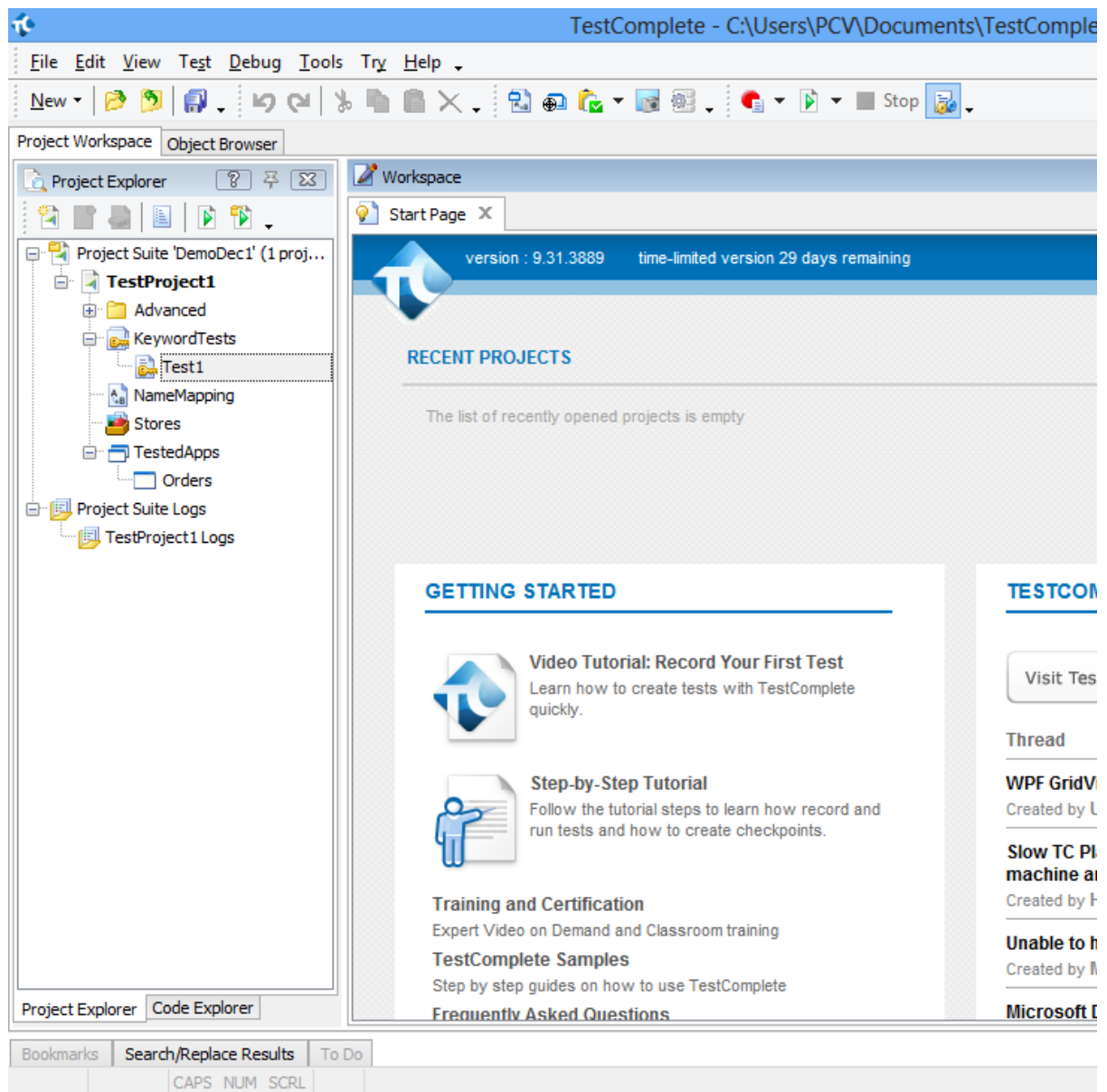


Chú ý: Khi dấu chọn ở Aautorun được chọn thì TestComplete sẽ tự động mở phần mềm theo đường dẫn lên bắt đầu thực hiện ghi lại bài test. Ngược lại nếu bỏ dấu chọn đó khi thực hiện Test bạn phải khởi động chương trình đó lên một cách thủ công.

- Chọn Next đến tiếp tục.
- Hộp thoại cho phép chọn ngôn ngữ để tạo ra các Script. <TestComplete hỗ trợ rất nhiều loại ngôn ngữ để sinh Script>



- Chọn Ngôn ngữ và Finish để kết thúc quá trình tạo.
- Project được tạo ra.



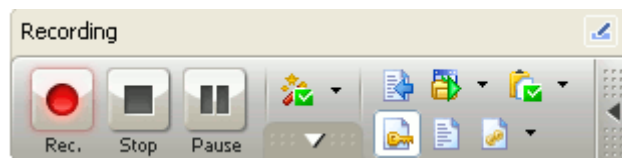
2.2. Ghi lại một bài test (Create a test)

Để ghi lại một bài test trên TestComplete bạn cần qua 3 bước.

- B1: Để bắt đầu một bài test bạn chọn Test | Record | Record Keyword Test. Hoặc Test | Record | Record Script. (Đây là 2 kiểu ghi lại bài test. Một là ghi lại và trình bày trực quan bằng hình ảnh _ Record KeyWord Test. Với Record Script sẽ ghi lại bài test dưới dạng Script với ngôn ngữ thể hiện do mình chọn khi tạo Project)

Có cách khác để là việc này là bạn chọn trên thanh công cụ Test Engine. hoặc có thể chọn vào Record New Test trên Start page.

Sau khi thực hiện Record như trên sẽ xuất hiện Recording Toolbar

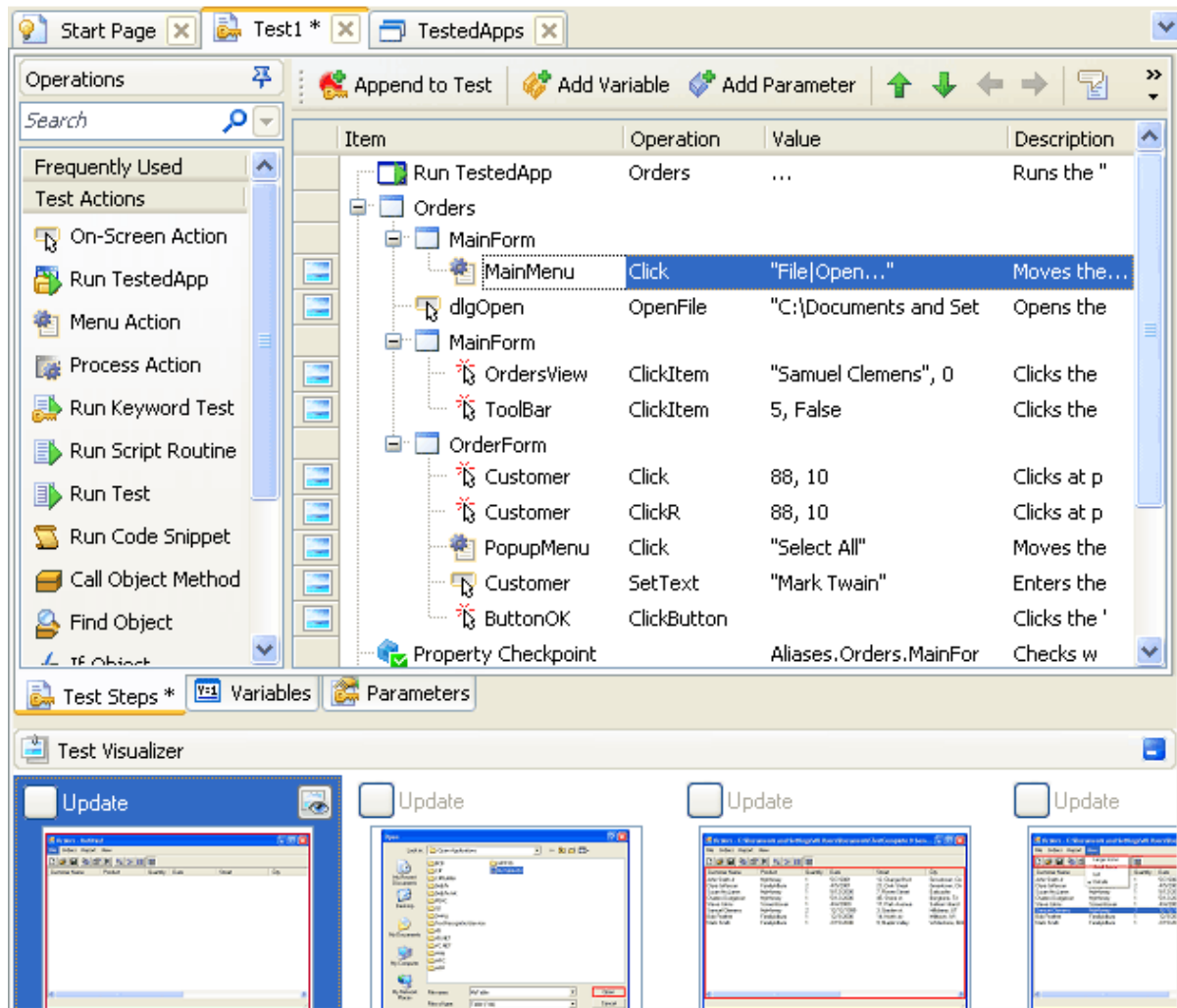


Thanh công cụ này chứa các Items cho phép ta thực hiện các chức năng khi Recording.

- B2: Sau khi khởi động phần Record phần mềm sẽ tự động khởi động chương trình và sẵn sàng ghi lại quá trình khi Tester thao tác. Các tính năng để thực hiện một bài test với nhiều yêu cầu khác nhau ở sẽ được Tester chọn trên thanh công cụ Recording Toolbar.

- B3: Khi kết thúc bài test chọn vào Item Stop trên thanh công cụ để kết thúc quá trình ghi lại bài test.

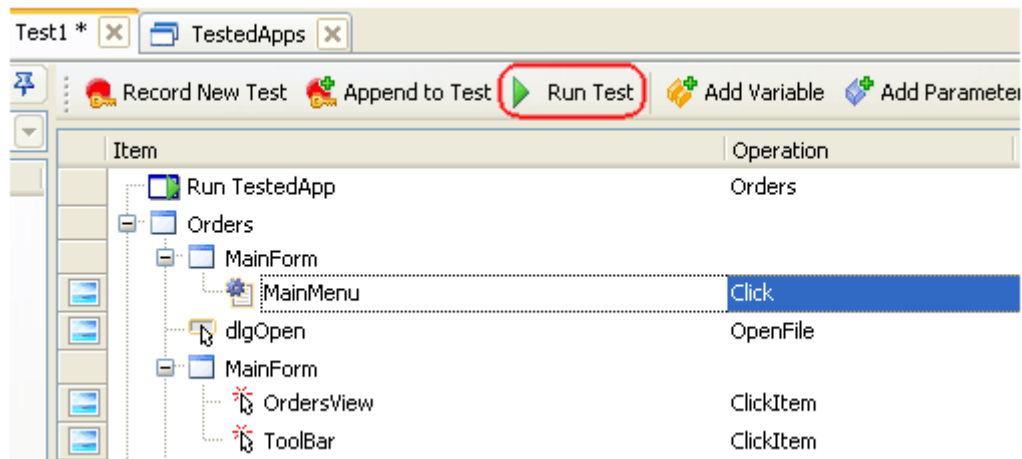
Đây là một ví dụ về kết quả hiển thị của Record Keyword Test



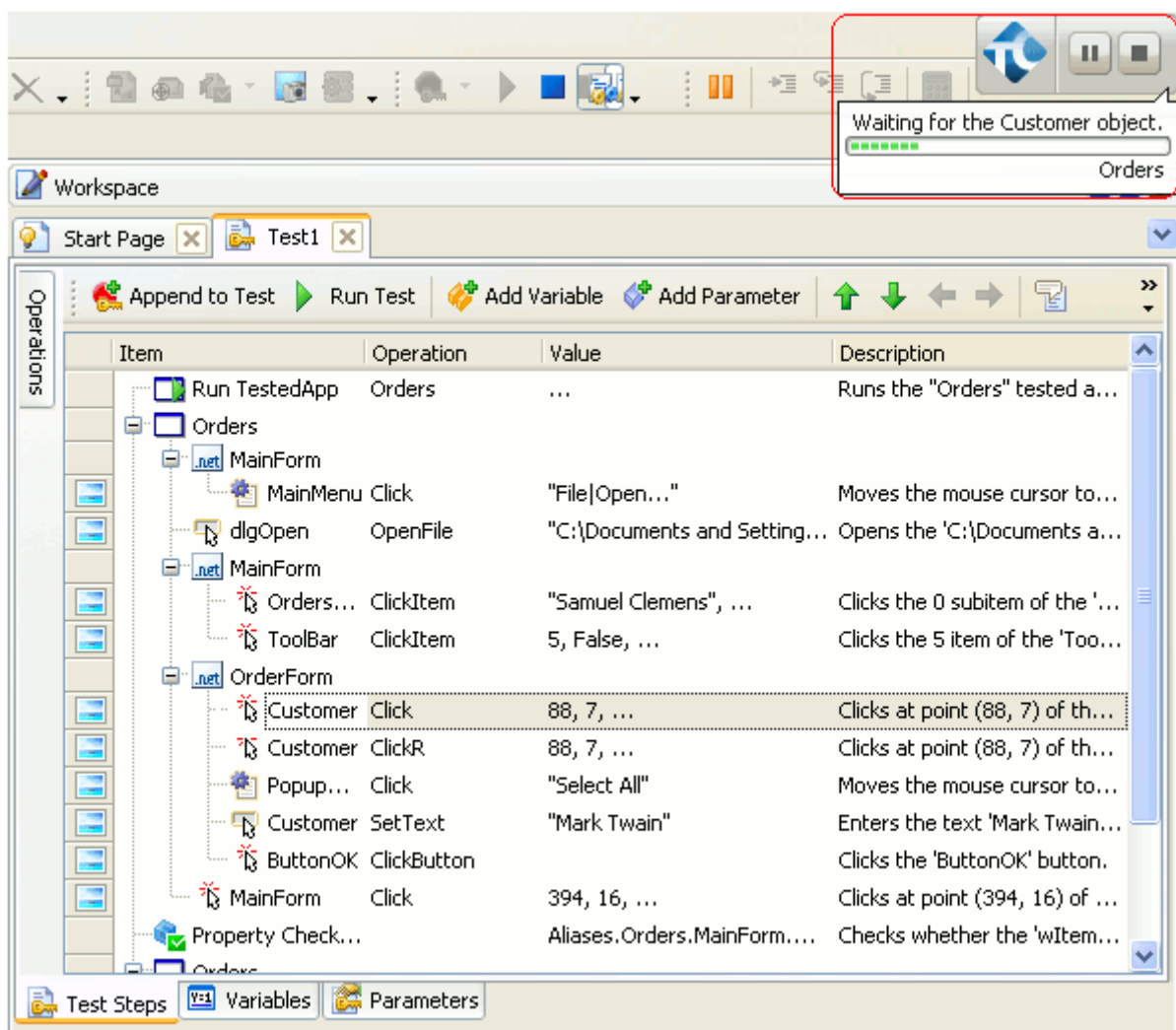
2.3.Chạy bài test đã được ghi trước đó (Running the Recorded test)

Khi có các bài test đã được tạo ra trước đó, Tester có thể chạy lại các bài test đó để kiểm tra bài test đó.

Để chạy một bài test trước đó có bạn phải chọn vào Item Run Test phía trên bài test trong khung Workplace.



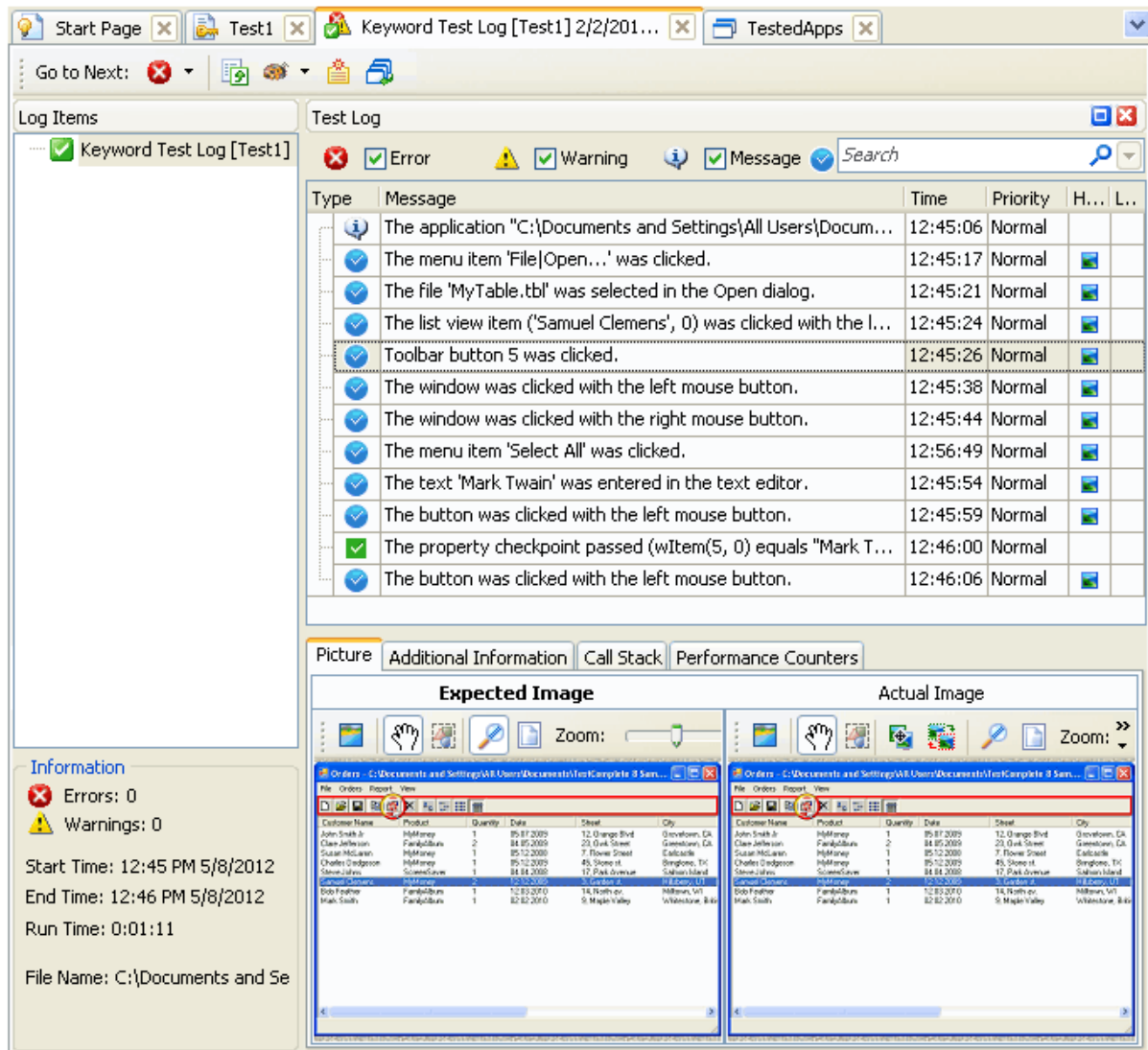
Khi thực hiện chạy bài test đó, TestComplete sẽ tự động làm lại theo các bước trong bài test đã được ghi lại.



Xuất hiện thanh công cụ cho phép tạo có thể kết thúc quá trình test lại trước khi nó tự động kết thúc (Khi chọn Item Stop) hoặc là có thể tạm dừng quá trình test lại để kiểm tra hoặc thực hiện thao tác bất kỳ. (Khi chọn Item Pause).

Cũng có cách khác để chạy bài test trước đó! Chọn bài test trong Project Explorer chuột phải chọn Run Test..

Kết thúc quá trình chạy bài test sẽ có một báo cáo về bài test. TestComplete thông kê từng bước thực hiện của bài test thông báo các Warning, Error, ...

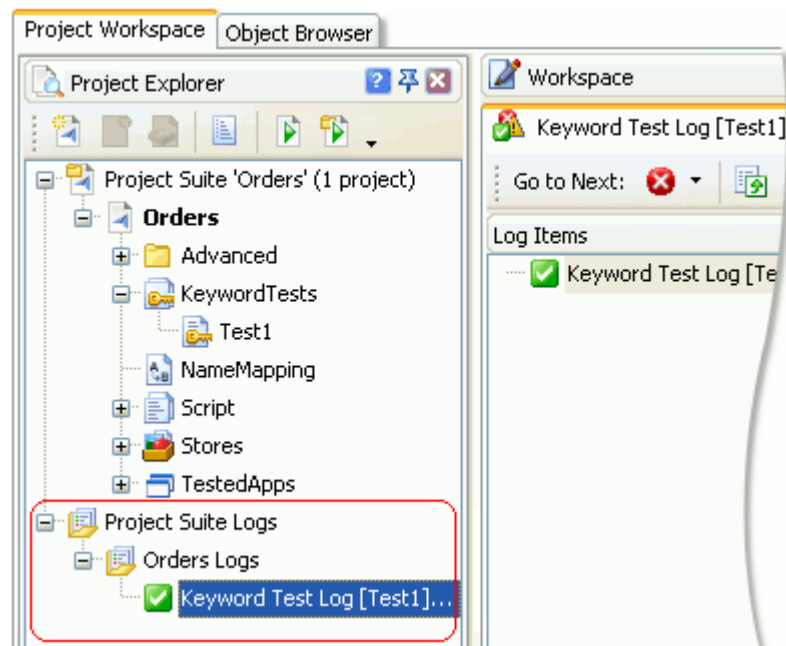


The screenshot displays the TestComplete interface with the 'Keyword Test Log [Test1] 2/2/201...' window open. The window shows a list of test steps with their corresponding messages, times, and priorities. The 'Log Items' pane on the left shows the test log for 'Keyword Test Log [Test1]'. The 'Test Log' pane on the right shows a table of test steps.

Type	Message	Time	Priority	H...	L...
Information	The application "C:\Documents and Settings\All Users\Docum..."	12:45:06	Normal		
Information	The menu item 'File\Open...' was clicked.	12:45:17	Normal		
Information	The file 'MyTable.tbl' was selected in the Open dialog.	12:45:21	Normal		
Information	The list view item ('Samuel Clemens', 0) was clicked with the l...	12:45:24	Normal		
Information	Toolbar button 5 was clicked.	12:45:26	Normal		
Information	The window was clicked with the left mouse button.	12:45:38	Normal		
Information	The window was clicked with the right mouse button.	12:45:44	Normal		
Information	The menu item 'Select All' was clicked.	12:56:49	Normal		
Information	The text 'Mark Twain' was entered in the text editor.	12:45:54	Normal		
Information	The button was clicked with the left mouse button.	12:45:59	Normal		
Success	The property checkpoint passed (wItem(5, 0) equals "Mark T...	12:46:00	Normal		
Information	The button was clicked with the left mouse button.	12:46:06	Normal		

Below the test log, there are tabs for 'Picture', 'Additional Information', 'Call Stack', and 'Performance Counters'. The 'Picture' tab is selected, showing 'Expected Image' and 'Actual Image' side-by-side. The 'Expected Image' shows a screenshot of the 'Orders' window with a red circle around the 'File' menu. The 'Actual Image' shows a similar screenshot. Below the images, there is an 'Information' pane showing test statistics: Errors: 0, Warnings: 0, Start Time: 12:45 PM 5/8/2012, End Time: 12:46 PM 5/8/2012, Run Time: 0:01:11, and File Name: C:\Documents and Se...

Báo cáo này được lưu tại phân Project Suite Logs cho phép ta tra cứu lại kết quả bài test đã được thực hiện trước đó.



2.4.Sửa chữa các kịch bản test đã ghi.

Với các kịch bản test được ghi lại ta có thể chỉnh sửa và thêm mới các công việc khác vào tại panel Test Steps Page trong WorkPlace.

Quá trình chỉnh sửa kịch bản test sẽ giúp cho kịch bản test ngày tốt hơn.

Một ví dụ là tạo một Data-Driven cho một kịch bản test sẽ làm cho các số khả năng phát hiện lỗi với một dữ liệu lớn hơn sẽ là lớn hơn.

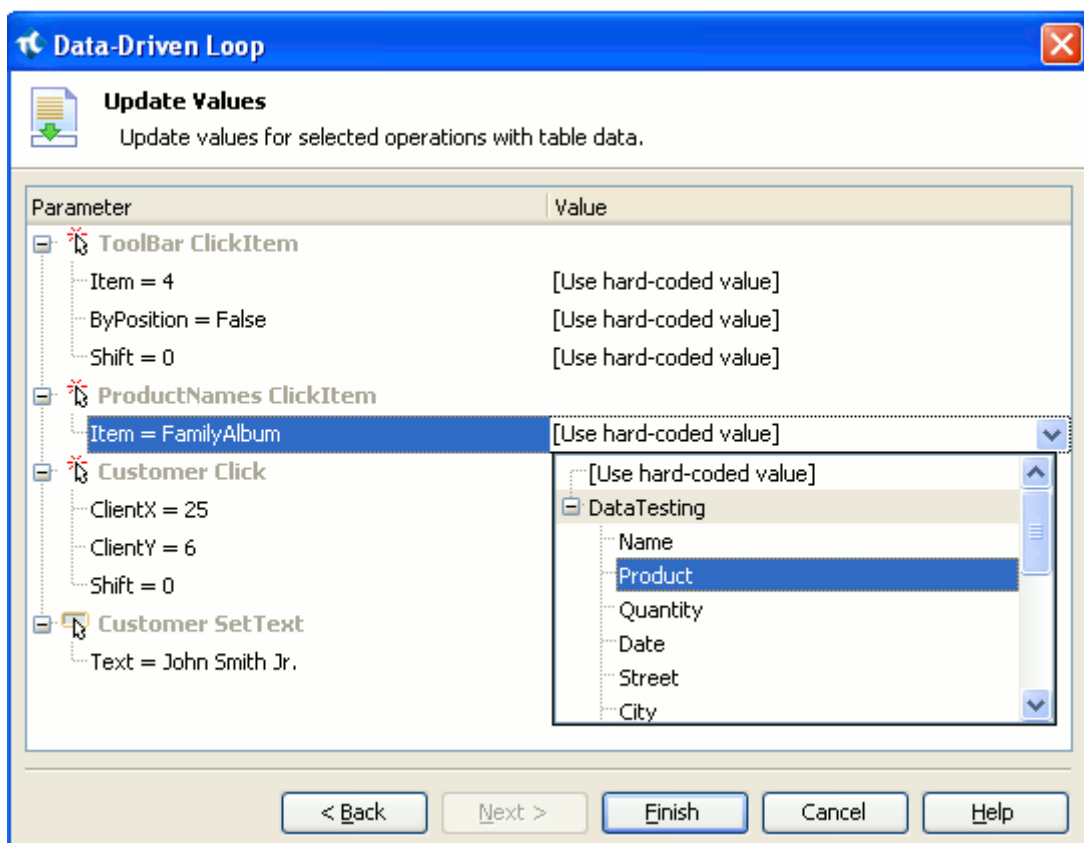
Để tạo một Data-Driven ta làm như sau:

- Tên kịch bản test sẵn có, ta chọn vùng có chứa các dữ liệu được nhập vào
- Nhấn phải chuột để chọn Data-Driven Loop...
- TestComplete sẽ hiện ra hộp toàn Data-Driven Loop wizard
- Trên trang đầu tiên của wizard này, bạn cần phải xác định tên của Bảng dữ liệu hoặc Bảng Variable mà sẽ cung cấp truy cập vào các dữ liệu được lưu trữ cần thiết. Lưu ý rằng bạn có thể chỉ định tên của một Bảng dữ liệu hoặc Bảng Variable đã có sẵn, hoặc tạo ra một Variable mới với tên cần thiết. Nếu bạn chọn để tạo ra một Variable mới, TestComplete sẽ hiển thị các trang bổ sung Variable (DB Bảng hoặc Bảng).

• Nếu bạn chọn để tạo ra một biến DB Bảng, sau đó bạn sẽ xác định loại lưu trữ dữ liệu cần thiết (một tập tin Excel, một tập tin CSV hoặc một bảng cơ sở dữ liệu) và các hồ sơ để được xử lý trong thời gian chạy thử nghiệm.

• Nếu bạn chọn để tạo ra một biến bảng, bạn sẽ nhập dữ liệu bằng tay hoặc tạo ra nó bằng cách sử dụng dữ liệu phát điện TestComplete của. Sau đó, bạn sẽ phải xác định tập hợp các hàng để được xử lý trong thời gian chạy thử nghiệm.

- Trang cuối cùng của wizard, Cập nhật các giá trị, bao gồm một danh sách các thông số được sử dụng bởi các hoạt động lựa chọn. Trên trang này, bạn có thể cập nhật giá trị của bất kỳ tham số và gán các dữ liệu này được lưu trữ trong biến được xác định trên trang đầu tiên của wizard, các thông số. Để làm điều này, nhấp vào bên trong tế bào tương ứng của cột Value. Để chọn một giá trị mới, sử dụng tiếp theo danh sách thả xuống. Danh sách này bao gồm các [Sử dụng giá trị mã hóa cứng] mục và một tập hợp các cột của bảng. Các [sử dụng giá trị mã hóa cứng] là một trong những mục mặc định, và nó có nghĩa là TestComplete sẽ không cập nhật giá trị tham số ban đầu. Để cập nhật các giá trị, sử dụng một trong các cột truy cập mà được cung cấp bởi Bảng DB hoặc biến Bảng.



Chọn Finish để kết thúc wizard.

- Cuối cùng ta sẽ có một kịch bản test đã được chỉnh sửa tốt hơn kịch bản trước.

Item	Operation	Value	Description
Run TestedApp	Orders	...	Runs the "Orders" tes...
Data-Driven Loop		DataTesting [Local] (All re...	
Orders			
MainForm			
ToolBar	ClickItem	4, False, ...	Clicks the 4 item of th...
OrderForm			
ProductNames	ClickItem	Variables.DataTesting("Pr...	Selects the KeywordT...
Customer	Click	25, 6, ...	Clicks at point (25, 6) ...
Customer	SetText	Variables.DataTesting("N...	Enters the text Keyw...
ButtonOK	ClickButton		Clicks the 'ButtonOK' ...
Orders			
MainForm	Close	...	Closes the 'MainForm' ...
dlgConfirmation			
btnNo	ClickButton		Clicks the 'btnNo' butt...

Đó là một ví dụ nhỏ cho thấy TestComplete rất hiệu quả để tạo ra các kịch bản test, hỗ trợ đắc lực cho công việc của một Tester.

D. KẾT LUẬN

Sau một thời gian đọc tài liệu và sử dụng phần mềm nhóm có một số nhận xét về công cụ Test complete cụ thể như sau:

❖ **Ưu điểm**

- Không giới hạn với các ứng dụng thử nghiệm.
- Không phụ thuộc vào loại công cụ phát triển.
- Hỗ trợ nhiều loại kiểm thử khác nhau.
- Hỗ trợ nhiều ngôn ngữ.
- Dễ sử dụng, dễ hiểu.

❖ **Nhược điểm**

- Không thể thay thế kiểm thử thủ công.
- Không có bản free. Bản trial hạn chế nhiều tính năng của Test Complete.