

AUTOMATION TESTING FOR MOBILE

Nhóm F2 – Oh Bug!

Table of Contents

01

Define

Khái niệm automation testing
và mobile automation testing

02

Process

Quy trình của mobile
automation testing

03

Advantages and Disadvantages

Lợi ích và nhược điểm của
mobile automatio testing

04

Cases

Những trường hợp nào nên
dùng Automation Testing cho
ứng dụng Mobile

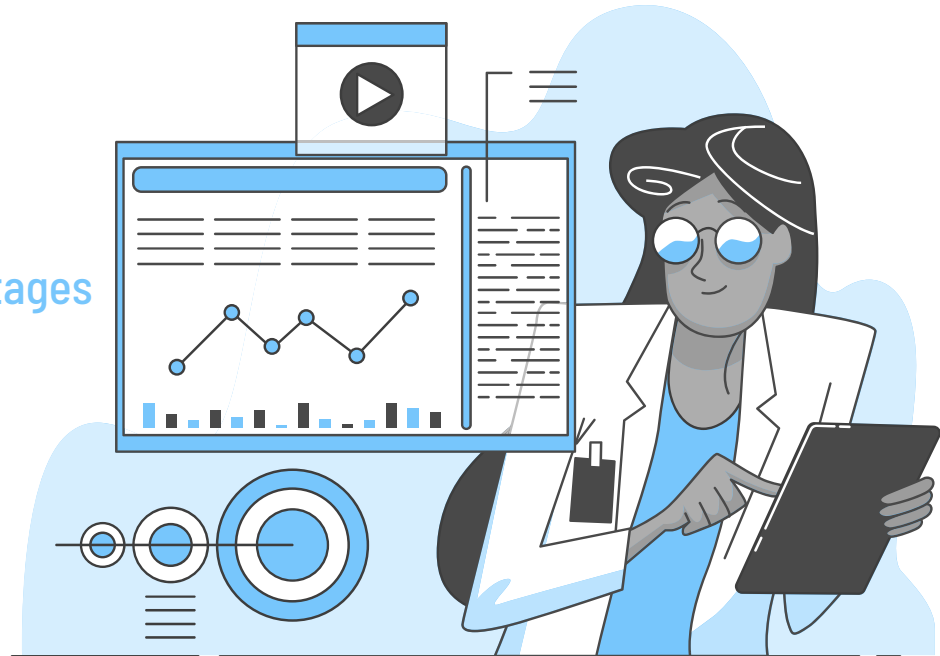


Table of Contents

05

Tools

Một số công cụ kiểm thử tự động
...

06

Compare

Bảng so sánh giữa các công cụ
...

07

Demo

Demo các kịch bản kiểm thử với các công cụ.
...



01

Define

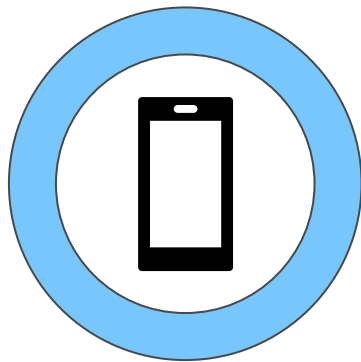
Khái niệm về automation testing và
mobile automation testing



Automation Testing

Kiểm thử tự động là việc thực hiện kiểm thử bằng máy mà người kiểm thử sẽ khởi động hệ thống, nhập dữ liệu đầu vào, kiểm tra và so sánh với dữ liệu đầu ra và ghi lại kết quả.

...



Mobile Automation Testing

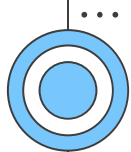
Là quá trình kiểm tra chức năng, hiệu suất, khả năng sử dụng và tính nhất quán trên các thiết bị di động như điện thoại di động, máy tính bảng, đảm bảo rằng ứng dụng hoạt động đúng và tương thích trên các thiết bị khác, nền tảng, hệ điều hành khác nhau và trong điều kiện sử dụng khác nhau.

...

02

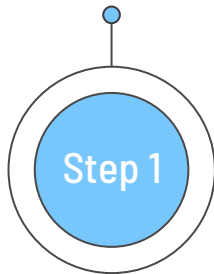
Process

Quy trình thực hiện Automation
Testing cho ứng dụng di động

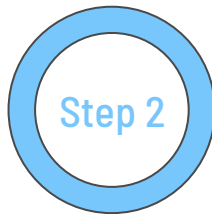


Process

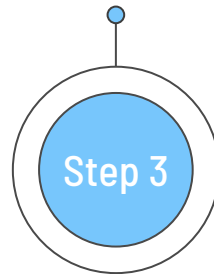
Xác định mục tiêu



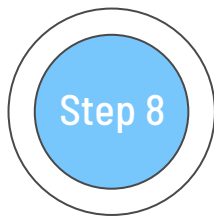
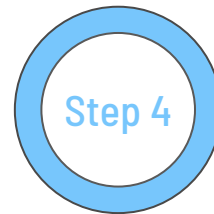
Lựa chọn công cụ



Xây dựng kịch bản



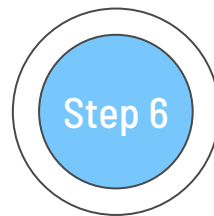
Tự động hóa kiểm thử



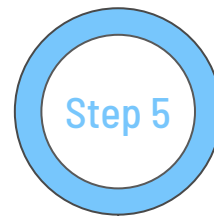
Tối ưu hóa



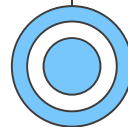
Báo cáo



Phân tích kết quả



Chạy kiểm thử



03

Advantages and Disadvantages

Ưu điểm và nhược điểm mobile
automation testing

Advantages



Tốc độ

Nhất là khi có nhiều trường hợp kiểm thử.



Tính lặp lại

Kiểm thử tự động có thể chạy nhiều test-case với cùng một quy trình



Độ tin cậy

Kiểm thử tự động ít xảy ra lỗi của con người hơn



Tiết kiệm thời gian

Không yêu cầu nỗ lực thủ công cho các tác vụ lặp đi lặp lại



Tích hợp liên tục

Tích hợp vào quy trình tích hợp liên tục, cho phép phát hiện sớm các lỗi

Disadvantages



Chi phí thiết lập ban đầu

Duy trì, bao gồm đầu tư vào công cụ, tập lệnh và cơ sở hạ tầng.



Độ phức tạp

Có thể phức tạp và khó hiểu đối với các bên liên quan phi kỹ thuật.



Chi phí bảo trì

Khi công cụ có sự thay đổi, dẫn đến chi phí bảo trì bổ sung



Kết quả "pass" giả

Bỏ sót một số lỗi, đặc biệt là những lỗi liên quan đến UX.

04

Cases

Những trường hợp nên sử dụng
Automation Testing cho ứng dụng
di động



Cases



Kịch bản lặp

Các kịch bản kiểm thử mà cần được thực hiện nhiều lần, automation testing có thể giúp tiết kiệm thời gian và tăng tính chính xác



Tích hợp

Kiểm tra sự tương tác giữa ứng dụng di động và các thành phần hệ thống bên ngoài



Đa nền tảng, thiết bị

Ứng dụng cần hoạt động trên nhiều loại thiết bị và nền tảng khác nhau





Cases



Hiệu suất, độ ổn định

Automation testing có thể tạo ra tải tự động và đo lường hiệu suất theo cách thức được thiết lập trước đó



Đa ngôn ngữ

Đảm bảo rằng tất cả các phiên bản ngôn ngữ của ứng dụng hoạt động đúng như mong đợi trên mọi thiết bị và hệ điều hành



Tương thích

Kiểm tra tính tương thích của ứng dụng trên nhiều loại thiết bị di động và phiên bản hệ điều hành khác nhau



05 Tools

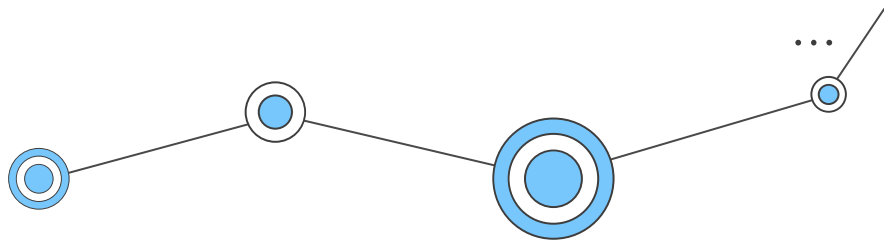
Một số công cụ kiểm thử tự động



Katalon Studio



Katalon

- <https://katalon.com/>
 - Katalon Studio giúp kiểm tra tự động các ứng dụng và dịch vụ , cung cấp một môi trường tích hợp để tạo, quản lý và chạy các kịch bản kiểm tra tự động một cách dễ dàng mà không cần biết nhiều về lập trình.
- 

Katalon Studio



Katalon

- Giao diện dựa trên GUI: giao diện đồ họa dễ sử dụng để tạo và quản lý các kịch bản kiểm tra
- Hỗ trợ nhiều nền tảng: iOS và Android, và dịch vụ web.
- Báo cáo và lưu trữ kết quả
- Hỗ trợ Kiểm thử đám mây
- Ghi video và phát lại: tạo kịch bản kiểm thử một cách thuận tiện
- Khả năng tích hợp: Git, Jenkins, qTest và Jira

Katalon Studio



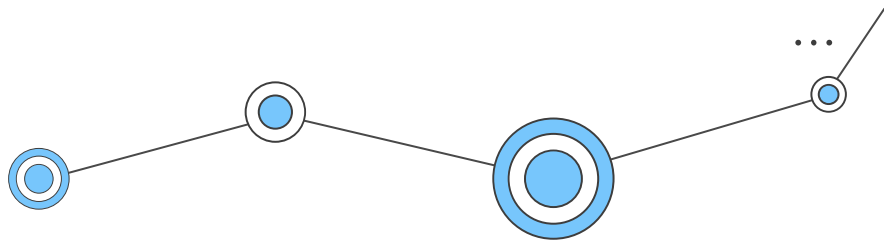
- Cài đặt và sử dụng đơn giản
- Dễ tiếp cận đối với những bạn mới bắt đầu học automation
- Hỗ trợ cho người có chuyên môn về lập trình
- Tạo bài kiểm thử hiệu quả thông qua các template sẵn có
- Kiểm thử đa trình duyệt dễ dàng
- Báo cáo kiểm thử tự động , trực quan



Katalon Studio



Katalon

- Cộng đồng support vẫn còn hạn chế
 - Chỉ hỗ Groovy và Java
 - Performance chậm khi project lớn
 - Giới hạn chức năng đối với free version
- 



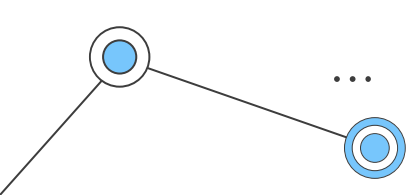
APPIUM

- <https://appium.io/docs/en/2.1/>
- Appium là một công cụ mã nguồn mở được sử dụng để kiểm thử tự động trên các ứng dụng di động trên các nền tảng iOS và Android.
- Appium được xây dựng trên cơ sở của WebDriver. Với kiến trúc đa nền tảng, Appium cho phép viết các kịch bản kiểm thử một lần và chạy chúng trên cả iOS và Android mà không cần sửa đổi mã nguồn.

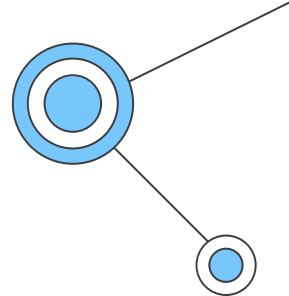


APPIUM

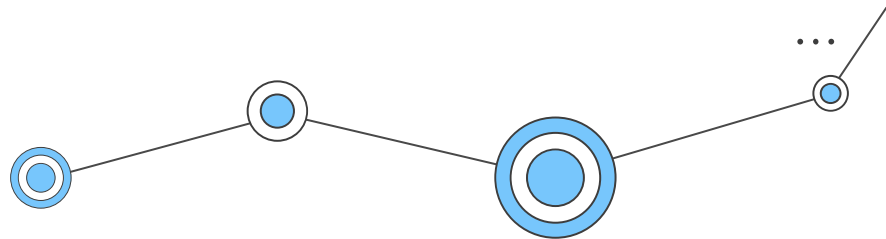
- Mã nguồn mở và miễn phí
- Hỗ trợ đa nền tảng: iOS và Android.
- Appium hỗ trợ nhiều ngôn ngữ lập trình như Java, Python, Ruby, JavaScript và C#.
- Hỗ trợ các ứng dụng native, hybrid và web.
- Tìm kiếm các thành phần giao diện đa nền tảng.



APPIUM



- Hỗ trợ kiểm thử trên thiết bị thật và giả lập (Emulators và Simulators).
- Tích hợp nhiều Framework kiểm thử: Appium tích hợp các Framework phổ biến như TestNG, JUnit và XCTest.
- Hỗ trợ Kiểm thử trên đám mây (Cloud testing).



Nightwatch.js



Nightwatch.js

- <https://nightwatchjs.org/>
- Nightwatch.js là một framework linh hoạt cho kiểm thử tự động mà mở rộng khả năng của mình để bao gồm kiểm thử cho ứng dụng di động, cả trên nền tảng Android và iOS, thông qua việc tích hợp với Appium.

Nightwatch.js



Nightwatch.js

- Tích hợp với Appium: cho phép kiểm thử tự động trên cả Android và iOS.
- Khung kiểm thử thống nhất: Đảm bảo sự nhất quán giữa các kịch bản kiểm thử trên web và di động.
- Kiểm thử chức năng Từ Đầu Đến Cuối
- Kiểm thử chéo nền tảng
- Ngôn ngữ JavaScript: dễ tiếp cận và sự quen thuộc cho nhà phát triển
- Thực hiện kiểm thử đồng thời

Nightwatch.js



Nightwatch.js

- Hiệu suất chéo nền tảng
- Khung kiểm thử thống nhất
- Tích hợp lệnh toàn diện: Bộ lệnh tích hợp sẵn giúp đơn giản hóa việc tạo kịch bản kiểm thử
- Dễ sử dụng
- Ưu tiên JavaScript
- Thực hiện kiểm thử đồng thời

Nightwatch.js



Nightwatch.js

- Hiệu năng: Giải quyết vấn đề về hiệu năng với các ứng dụng lớn và phức tạp.
- Quản lý trạng thái
- Hạn chế trong xử lý ảnh: liên quan đến xác nhận và so sánh hình ảnh.
- Cộng đồng và tài nguyên hạn chế
- Khả năng mở rộng và tích hợp
- Chưa có sự hỗ trợ mạnh mẽ cho kiểm thử đa ngôn ngữ.

Espresso



- <https://developer.android.com/training/testing/espresso>
- Espresso là một framework được cung cấp và phát triển bởi google nhằm phục vụ cho việc kiểm thử giao diện trên nền tảng Android.

Espresso



- Khả năng đồng bộ hóa tự động
- Quản lý hành vi của Android Intents
- Tự động đồng bộ hóa với giao diện người dùng
- Tính linh hoạt và mở rộng cao: API đơn giản, nhẹ, dễ học và dễ bảo trì, với khả năng linh hoạt và mở rộng cao.
- Hỗ trợ kiểm thử unit và black-box
- Module Riêng Biệt Cho Kiểm Thử Android WebView và Android Intents
- Hỗ trợ lấy phản hồi nhanh chóng
- Không sử dụng máy chủ

Espresso



- Đồng bộ tự động: đồng bộ tự động với giao diện người dùng để thực hiện kiểm thử một cách ổn định và nhanh chóng mà không cần kết nối đến máy chủ.
- Dễ sử dụng: Tích hợp với Android Studio, giúp việc gỡ lỗi và phân tích kết quả kiểm thử trở nên thuận tiện.
- Luồng làm việc đơn giản và phản hồi nhanh chóng
- API ngắn gọn

Espresso



- Hạn chế về ngôn ngữ lập trình: chỉ bao gồm Java và Kotlin
- Không hỗ trợ chéo nền tảng
- Tập trung chủ yếu vào kiểm thử giao diện người dung: hạn chế đối với dự án lớn hoặc phức tạp đòi hỏi sự phối hợp giữa nhiều đội và môi trường.
- Hơn nữa, việc tự động hóa với Espresso yêu cầu bạn phải có quyền truy cập vào mã nguồn của ứng dụng, có thể làm cho quá trình tự động hóa kiểm thử trở nên phức tạp.

TestComplete



SMARTBEAR

TestComplete

- <https://smartbear.com/product/testcomplete/overview/>
- Với sự hỗ trợ tích hợp cho hơn 500 cách điều khiển và framework của bên thứ ba, TestComplete có thể xử lý và xác định các thành phần giao diện người dùng động trong hầu hết các công nghệ hiện có.



SMARTBEAR

TestComplete

TestComplete

- Ngôn ngữ lập trình được hỗ trợ: JavaScript, Python, VBScript, JScript, Delphi, C++, C#.
- Các phương pháp linh hoạt để thiết kế kiểm thử: chế độ ghi và phát lại, chế độ thủ công và tập lệnh với các từ khóa tích hợp.
- Nhận dạng đối tượng bằng nhận dạng hình ảnh dựa trên thuộc tính và AI.
- Kiểm thử song song, đa trình duyệt và đa thiết bị.
- Tích hợp với các khung thử nghiệm, công cụ CI/CD khác và hệ sinh thái SmartBear.

TestComplete



SMARTBEAR

TestComplete

Điểm mạnh

- Tính linh hoạt.
- Kiểm thử đa nền tảng.
- Nhận dạng đối tượng và viết kịch bản.
- Ghi và phát lại.
- Bộ thư viện kiểm thử và tích hợp phong phú.
- Trực quan hóa kiểm tra.

TestComplete



SMARTBEAR

TestComplete

Điểm yếu

- Chi phí bản quyền.
- Chỉ hỗ trợ Windows.
- Chi phí bảo trì.
- Ghi và phát lại.
- Tùy chọn kiểm thử đám mây hạn chế..

Ranorex Studio



- <https://www.ranorex.com/>
- Mô tả: Ranorex Studio là một giải pháp tự động hóa kiểm thử phần mềm mạnh mẽ dành cho kiểm thử ứng dụng máy tính, ứng dụng di động và trang web.
- Cung cấp một môi trường đồ họa để tạo và quản lý các kịch bản kiểm thử tự động và hỗ trợ việc kiểm tra trên nhiều nền tảng.

Ranorex Studio



Chức năng

- Ghi và phát lại kịch bản kiểm thử tự động
- Hỗ trợ kiểm thử đa nền tảng: Windows, web, Android, iOS và nhiều hệ điều hành khác.
- Quản lý kiểm thử
- Hỗ trợ cho ngôn ngữ lập trình: tùy chỉnh kịch bản kiểm thử bằng cách sử dụng C# hoặc VB.NET.
- Hỗ trợ cho kiểm thử liên tục

Ranorex Studio



Điểm mạnh

- Kiểm thử tự động đa nền tảng.
- Tạo test-case mà không cần code: Ranorex Recorder với giao diện kéo và thả cho phép thực hiện các test-case không cần code.
- Sử dụng dễ dàng: Ranorex được coi là một trong những công cụ tự động hóa thử nghiệm dễ tiếp cận nhất trên thị trường.
- Theo dõi image.



Ranorex[®]

An Idera, Inc. Company

Ranorex Studio

Điểm yếu

- Bản quyền trả phí.
- Chỉ có một số ngôn ngữ được hỗ trợ, không giống như Selenium cho phép mã hóa bằng mười ngôn ngữ lập trình phổ biến.
- MacOS không thể cài Ranorex Studio.
- Cộng đồng sử dụng không lớn, chính vì vậy nếu có bug sẽ khó trong việc tìm hiểu.
- Các phiên bản của Ranorex Studio không mang tính ổn định, thường xuyên được cập nhật.



ACCELQ

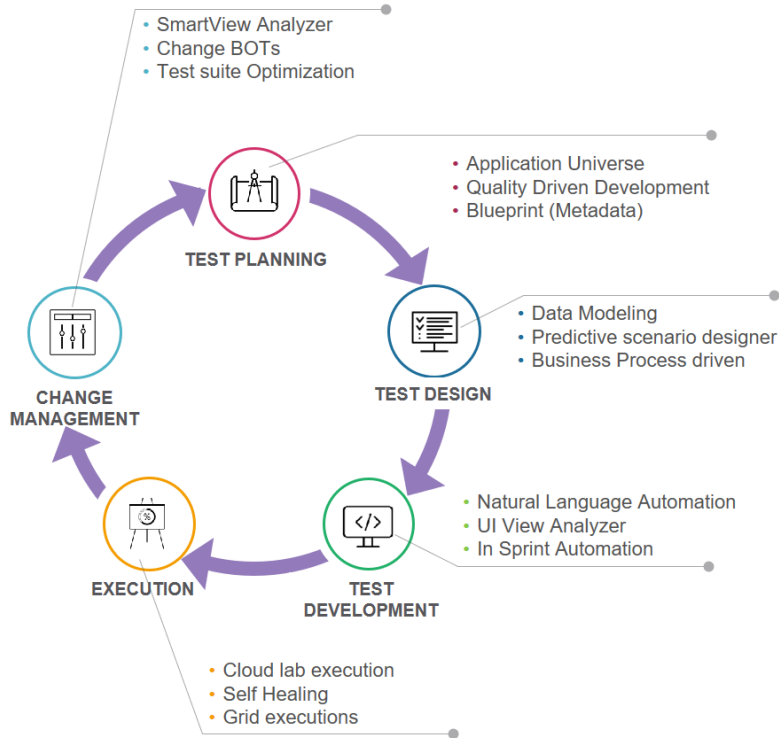


- <https://accelq.com/>
- Là một công cụ tự động hóa kiểm thử phần mềm được phát triển để tăng cường quá trình kiểm thử phần mềm. Nó cung cấp các tính năng quản lý kiểm thử, tự động hóa kiểm thử, và theo dõi quá trình kiểm thử.



ACCELQ

ACCELQ



- Tự động hóa kiểm thử
- Tạo và quản lý kịch bản kiểm thử
- Quản lý quy trình kiểm thử
- Kết hợp DevOps
- Xây dựng và quản lý dữ liệu kiểm thử
- Báo cáo và theo dõi tiến độ kiểm thử theo dõi tiến độ kiểm thử theo thời gian
- Kiểm thử liên tục và tự động
- Kiểm thử đa kênh và đa nền tảng



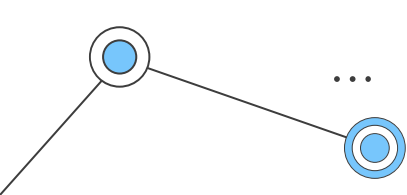
QUANTUM

- <https://www.perfecto.io/integrations/quantum>
- Quantum là một framework kiểm thử tự động mã nguồn mở được phát triển bởi Perfecto. Nó cung cấp khả năng kiểm thử tự động trên cả ứng dụng di động và trình duyệt web. Quantum hỗ trợ kiểm thử trên các nền tảng di động như iOS và Android.

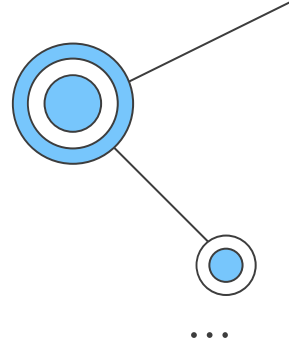


QUANTUM

- Mã nguồn mở, đơn giản và dễ sử dụng.
- Quantum hỗ trợ kiểm thử trên cả hai nền tảng di động chính là iOS và Android.
- Xử lý sự kiện trên giao diện người dùng ở thiết bị di động
- Chạy kiểm thử đa thiết bị.
- Tạo báo cáo kết quả kiểm thử
- Tích hợp với các công cụ kiểm thử liên tục (CI/CD)

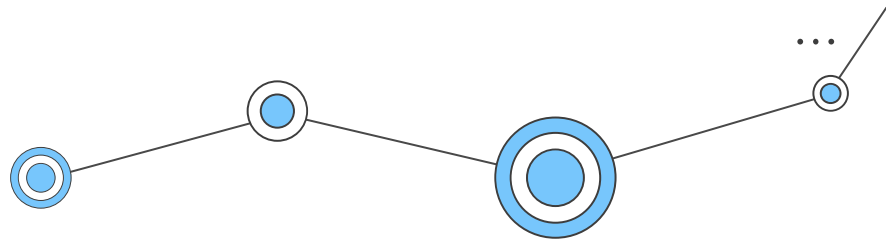


KOBITON



Kobiton

- <https://kobiton.com/>
- Kobiton là một nền tảng kiểm thử và triển khai ứng dụng di động dựa trên cloud. Nó cung cấp các công cụ và dịch vụ để kiểm thử ứng dụng di động trên nhiều thiết bị và nền tảng khác nhau.





Kobiton

KOBITON

- Kobiton cho phép kiểm thử trên nhiều hệ điều hành, bao gồm iOS và Android.
- Tích hợp các cộng cụ Appium và Selenium.
- Hỗ trợ quản lý vòng đời ứng dụng.
- Kobiton giúp quản lý thiết bị từ xa thông qua Cloud.
- Cung cấp các báo cáo và phân tích chi tiết về quá trình kiểm thử.



Robotium

ROBOTIUM

- <https://github.com/RobotiumTech/robotium>
- Robotium là một framework kiểm thử mã nguồn mở được tạo ra để giúp việc viết các trường hợp kiểm thử giao diện người dùng (UI) tự động mạnh mẽ và chắc chắn cho ứng dụng di động Android.



Robotium

ROBOTIUM

- Là công cụ mã nguồn mở và miễn phí.
- Dễ sử dụng và giúp tiết kiệm thời gian để viết và thực thi.
- Tập trung việc hỗ trợ kiểm thử tự động trên Android.
- Hỗ trợ kiểm thử các ứng dụng native và hybrid.
- Tích hợp các công cụ: Maven, Gradle và Ant.



06

Compare

Bảng so sánh giữa các công cụ



	Katalon	Appnium	TestComplete	Ranorex Studio	Nightwatch.js
Giá cả	Gồm 2 version: Free và Premium	Miễn phí	Đắt đỏ - 3655 đô la	Có trả phí	Miễn phí
Hệ điều hành công cụ hoạt động	Windows, macOS, Linux	Windows, macOS, Linux	Chỉ Windows	Chỉ Windows	Windows , macOS , Linux
Cách tiếp cận (GUI hay code)	GUI và Code	GUI và Code	GUI và Code	GUI và Code	Code
Ngôn ngữ kiểm thử	Groovy và Java	Java, Ruby, Python, PHP, JavaScript, C#	JavaScript, Python, VBScript, JScript, Delphi, C++, C#.	C#, VB.NET	JavaScript, TypeScript
Hỗ trợ Android	Có	Có	Có	Có	Có
Hỗ trợ iOS	Có	Có	Có	Có	Có
Có ghi và phát lại kiểm thử hay không?	Có	Có	Có	Có	Không
Cộng đồng hỗ trợ	Cộng đồng hỗ trợ còn hạn chế	Cộng đồng lớn mạnh	Cộng đồng hỗ trợ không lớn	Cộng đồng hỗ trợ không lớn	Cộng đồng hỗ trợ không lớn

	Espresso	Quantum	Kobiton	AcceLQ	Robotium
Giá cả	Miễn phí	Có trả phí	Trải nghiệm, Có phí	Trải nghiệm, Có phí	Miễn phí
Hệ điều hành công cụ hoạt động	Android	Windows, macOS, Linux	Windows, macOS	Windows, macOS, Linux	Windows, macOS, Linux
Cách tiếp cận (GUI hay code)	Code	Code	GUI, Code	GUI	Code
Ngôn ngữ kiểm thử	Java , Kotlin	Java và JavaScript	C#, Java, Ruby, NodeJS, PHP, or Python	"natural language programming" and application abstraction	Java
Hỗ trợ Android	Có	Có	Có	Có	Có
Hỗ trợ iOS	Không	Có	Có	Có	Không
Có ghi và phát lại kiểm thử hay không?	Không	Không	Có	Có	Không
Cộng đồng hỗ trợ	Cộng đồng hỗ trợ không lớn	Cộng đồng hỗ trợ không lớn	Cộng đồng hỗ trợ không lớn	Cộng đồng hỗ trợ không lớn	Cộng đồng hỗ trợ không lớn

Thanks!

Do you have any questions?

