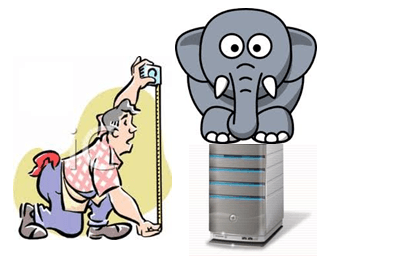
1. **SÁCH**
2. *Thế nào là performent testing?*

* Performance testing là việc thực hiện test để xác định một hệ thống có thể đáp ứng và ổn định với yêu cầu độ tải cao.
* Nó có thể phục vụ để điều tra, đo đạc, xác nhận hoặc xác minh chất lượng các thuộc tính của hệ thống như: khả năng thay đổi, tính tin cậy, và tài nguyên sử dụng.



1. *Thế nào là “stress test”?*

* **Kiểm thử áp lực (stress test)** :là kiểm thử được tiến hành bằng cách kiểm thử hệ thống trong điều kiện **tải bất hợp lý** để xác định điểm dừng (breakpoint) của hệ thống.

1. *Server- side testing là gì?*

* Kiểm thử bên phía Server (server-side) là kiểm thử yêu cầu các sử dụng sáng tạo (creative use) bên phía người dùng sử dụng giao diện, việc tạo ra một số giao diện khách hàng tùy chỉnh để đạt được các phần mềm phía máy chủ, sử dụng các công cụ và tiện ích, và thực hiện các

kỹ thuật mới.

* Hầu hết các ứng dụng được xây dựng sử dụng một số thành phần server (server components) và những phần mềm tùy chỉnh tạo nên các ứng dụng. Bạn sẽ cần phải xem xét thử nghiệm khả năng tương tác của các lớp giao diện cũng như cách nó tương tác với các thành phần này.

1. *User Interface testing là gì và những kỹ thuật áp dụng trong đó*
   1. *Khái niệm*
   * *Kiểm tra giao diện người dùng là kiểm tra sự tương tác giữa các đơn vị, kiểm tra khả năng sử dụng và các chức năng nhiệm vụ khác.*
   1. *Những kỹ thuật áp dụng*
   * *Kiểm thử thiết kế giao diện*
     + Thực hiện kiểm thử giao diện người dùng là thực hiện kiểm tra các ứng dụng với một mắt hướng hoạt động của nó. Đánh giá liệu các tính năng giao diện người dùng làm việc đúng cách.s
   * *Kiểm thử thực thi giao diện*
     + Kiểm tra khả năng sử dụng bao gồm một loạt các phương pháp để thiết lập các sản phẩm, giao nhiệm vụ cho người sử dụng để thực hiện, có những người sử dụng thực hiện các nhiệm vụ, và quan sát tương tác người dùng và thu thập thông tin để đánh giá khả năng sử dụng.
   * *Kiểm thử khả năng sử dụng và tiếp cận*
     + *Khả năng sử dụng (Usability Testing)*
       - Khả năng sử dụng là một thước đo giúp các nhà thiết kế của một sản phẩm hoặc dịch vụ (bao gồm cả các ứng dụng web, ứng dụng phần mềm và ứng dụng di động) xác định sự hài lòng của người sử dụng khi họ tương tác với một sản phẩm hoặc dịch vụ thông qua giao diện, bao gồm giao diện người dùng.
       - Kiểm tra khả năng sử dụng bao gồm một loạt các phương pháp để thiết lập các sản phẩm, giao nhiệm vụ cho người sử dụng để thực hiện, có những người sử dụng thực hiện các nhiệm vụ, và quan sát tương tác người dùng và thu thập thông tin để đánh giá khả năng sử dụng.
     + Khả năng tiếp cận - truy cập (Accessibility Testing)
       - Khả năng tiếp cận là về việc cung cấp sự hài lòng lớn nhất cho người dùng.
       - Các sự khác biệt chính là khả năng tiếp cận hoàn thành mục tiêu của mình thông qua làm việc sản phẩm có thể sử dụng cho một dân số lớn hơn, kể cả những người khuyết tật. Một số lượng lớn các thông tin về phương pháp và công cụ dùng để thiết kế các sản phẩm tiếp cận được rộng rãi hiện nay.
2. **TOOL**
3. *Junit*
   1. *Mục đích*
      * Framework đơn giản dùng cho việc tạo các unit testing tự động. Cung cấp cho chúng ta các gói lớp có sẵn cho phép chúng ta viết các phương thức test một cách dễ dàng. Chạy các test có thể lặp đi lặp lại
   2. *Tính năng*
      * Xác nhận việc kiểm tra kết quả được mong đợi
      * Các Test Suite cho phép chúng ta dễ dàng tổ chức và chạy các test Hỗ trợ giao diện đồ họa và giao diện dòng lệnh
4. *Marven*
   1. *Mục đích*
      * **Maven** là công cụ quản lý và thiết lập tự động 1 dự án phần mềm. Chủ yếu dùng cho các lập trình viên java, nhưng nó cũng có thể được dùng để xây dựng và quản lý các dự án dùng C#, Ruby, Scala hay ngôn ngữ khác.
      * Apache Maven là một công cụ bao quát và quản lý dự án phần mềm. Dựa trên khái niệm về project object model – POM (mô hình đối tượng dự án), Maven có thể quản lý kiến trúc của dự án, tạo tài liệu và báo cáo dựa trên thông tin lưu trữ tập trung (2).
   2. *Tính năng*
      * Cốt lõi của Maven là một plugin execution framework (nền tảng thực thi plugin); tất cả công việc đều do các plugin thực hiện.
      * Có hai loại plugin:
        + Build plugin: <build/>
        + Reporting plugin: <reporting/>
      * Plugin
        + Maven Surefire Plugin
          - Được dùng trong giai đoạn *test* của build lifecycle để thực hiện unit test.
          - Tạo ra hai tập tin báo cáo:

*\*.txt*

*\*xml*

* + - * Failsafe Plugin
        + Được thiết kế để chạy integration test.
        + Bốn phase (giai đoạn) chạy integration test của Maven:

*pre-integration-test*

*integration-test*

*post-integration-test*

*verify*

1. *Gradle*
   1. *Mục đích*
      * **Gradle**là một công cụ tự động build project dựa trên các khái niệm của Apache Ant, Apache Maven.
      * Với sự kết hợp các ưu điểm của cả Ant và Maven đồng thời mang lại các cải thiện mới thông qua việc sử dụng ngôn ngữ Groovy (một ngôn ngữ chạy trên JVM) trong việc giải quyết bài toán về đóng gói.
      * Gradle được thiết kế cho việc build multi-project có thể phát triển được mở rộng, và hỗ trợ xây dựng gia tăng bởi việc xác định thông minh các bộ phận của cây xây dựng được up-to-date, do đó bất kỳ công việc phụ thuộc vào những bộ phận sẽ không cần phải thực hiện lại.
      * Gradle được thiết kế cho việc build multi-project có thể phát triển được mở rộng, và hỗ trợ xây dựng gia tăng bởi việc xác định thông minh các bộ phận của cây xây dựng được up-to-date, do đó bất kỳ công việc phụ thuộc vào những bộ phận sẽ không cần phải thực hiện lại.
   2. *Tính năng*
   * **Groovy**

* Tất cả API của Gradle được xây dựng từ ngôn ngữ Groovy.
* Apache Groovy là 1 ngôn ngữ hướng đối tượng cho nền tảng java. Nó là 1 ngôn ngữ động có các tính năng tương tự Python, Ruby,…
* Nó có thể được sử dụng như là 1 ngôn ngữ kịch bản cho nền tảng java
* Cấu trúc Groovy trong Gradle đơn giản giúp giảm bớt sự khó khăn khi tự động hóa biên dịch
  + **Tasks**
    - Mỗi project gradle thực hiện tự động hóa build được tạo thành nhiều task khác nhau. 1 Task là 1 phần công việc mà build project thực hiện.
    - 1 task có thể biên dịch thành nhiều lớp, files, tạo thư viện jar, tạo java doc,…
  + **Quản lý dependencies**
    - * Gradle xây dựng script cho 1 quá trình build project, mỗi project có 1 vài dependencies and 1 vài publications. Dependencies là 1 thứ hỗ trợ cho việc build project (ví dụ như cần 1 file JAR từ project khác hoặc từ file jar từ ổ cứng). Publications là kết quả của project ví dụ như là class files build files
      * Hầu hết các project đều không khép kín. Nó cần 1 số file build từ project khác để biên dịch. Gradle có thể sử dụng các script đặc biệt dùng để download thư viện jar từ internet

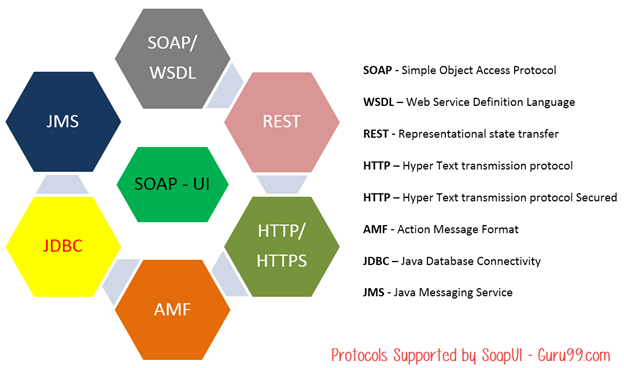
1. *Mockito*
   1. *Khái niệm/mục đích*
      * + Mockito là một testing framework mã nguồn mở cho Java phát hành theo Giấy phép MIT. Framework cho phép tạo ra các đối tượng đôi kiểm tra (đối tượng giả) trong unit tests cho mục đích phát triển Test-driven (TDD) hay Behavior Driven Development (BDD).
        + Mocking là một cách để kiểm tra các chức năng của một lớp độc lập.Mocking không đòi hỏi một kết nối cơ sở dữ liệu hay tập tin thuộc tính đọc hoặc máy chủ tập tin đọc để thử nghiệm một chức năng. các đối tượng giả làm mocking của dịch vụ thực sự. Một đối tượng giả trả về một dữ liệu giả tương ứng với một số đầu vào giả truyền cho nó.
        + Mockito là một mock framework phổ biến mà có thể được sử dụng kết hợp với JUnit. Mockito cho phép bạn tạo và cấu hình các đối tượng giả. Sử dụng Mockito đơn giản hoá các bài test cho các lớp học với phụ thuộc bên ngoài một cách đáng kể.
   2. *Hỗ trợ*
      * + Không có chữ viết tay - Không cần phải viết các đối tượng giả trên của riêng bạn.
        + Refactoring An toàn - Đổi tên tên phương pháp giao diện hoặc sắp xếp lại các thông số sẽ không phá vỡ các mã kiểm tra như Mocks được tạo ra tại thời gian chạy.
        + Hỗ trợ quay trở lại giá trị
        + Hỗ trợ ngoại lệ - Hỗ trợ trường hợp ngoại lệ.
        + Hỗ trợ kiểm tra - Hỗ trợ kiểm tra theo lệnh của phương thức gọi.
        + Hỗ trợ chú thích - Hỗ trợ tạo mocks sử dụng chú thích
2. *Soap UI*

* *Khái niệm / mục đích:*
  + *SOAP-UI là công cụ kiểm thử mã nguồn mở API hàng đầu.​*
  + *SOAP-UI cho phép kiểm thử tự động chức năng,kiểm thử hồi quy,kiểm thử tải trên các web API.​*
  + *SOAP-UI hỗ trợ các giao thức chuẩn và công nghệ để test các loại API.​*
  + *Ngoài ra SOAP-UI còn cho phép kiểm thử phi chức năng như hiệu xuất và bảo mật.*
* *Hỗ trợ/ feature*

1. *Kiểm tra chức năng​*
   * *Một công cụ mạnh mẽ cho phép tester viết Functional API Tests trong SoapUI.​*
   * *Hỗ trợ tính năng kéo-thả làm tăng tốc độ phát triển script.​*
   * *Hỗ trợ debug và cho phép tester phát triển data driven tests.​*
   * *Hỗ trợ nhiều môi trường – Dễ dàng chuyển đổi giữa QA, Dev và Prod Environments.​*
   * *Cho phép viết script cao cấp (tester có thể tùy chỉnh code theo kịch bản).​*
2. *Kiểm tra bảo mật​*
   * *Ngăn chặn SQL Injection để bảo đảm cơ sở dữ liệu.​*
   * *Thực hiện Fuzzing scan và Boundary scan để tránh những hành vi thất thường của các dịch vụ.*
3. *Kiểm thử tải​*
   * *Kiểm thử khả năng chịu tải của một ứng dụng web sử dụng loadUI. Sau khi thực hiện kiểm tra tải, LoadUI sẽ tạo ra một bản báo cáo, giúp xác định liệu các ứng dụng có thể chịu tải nặng hay không.​*
   * *Kiểm thử khả năng chịu tải của một ứng dụng web sử dụng loadUI.​*
   * *Mô phỏng mức độ cao và kiểm thử tải thực tế một cách dễ dàng.​*

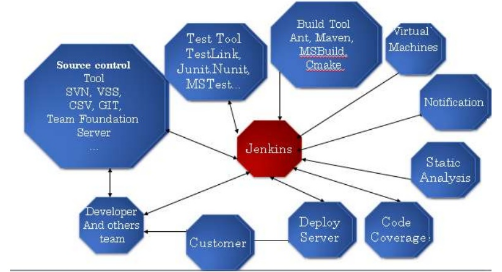
*Cho phép tùy chỉnh báo cáo chi tiết để nắm bắt các thông số hiệu suất.​*

1. *Hỗ trợ các giao thức và công nghệ*



*​*

1. *Selenium*
   1. *Khái niệm / Mục đích*
      * *Selenium là một công cụ kiểm thử phần mềm tự động, được phát triển bởi ThoughtWorks từ năm 2004 với tên ban đầu là JavaScriptTestRunner.*
      * *Selenium là một công cụ kiểm tra tự động các ứng dụng Web, nó có thể chạy trên hầu hết các trình duyệt và hệ điều hành.*
      * *Selenium là một công cụ mã nguồn mở. Điểm này là mạnh nhất của Selenium khi so sánh với các Test Tool khác. Vì là mã nguồn mở nên chúng ta không cần lo lắng về phí bản quyền hay thời hạn sử dụng.*
      * *Có thể triển khai trên đa nền tảng: Windowns, Linux, Mac.*
      * *Hỗ trợ hầu hết các ngôn ngữ lập trình Web như HTML, Java, PHP, Python, Ruby, Perl…*
      * *Selenium bao gồm 4 phiên bản chính: Selenium IDE, Selenium Remote Control (Selenium RC), Selenium Gird và Selenium WebDriver.*
   2. *Hỗ trợ*
      * *Selenium IDE*
        + *Là Framework đơn giản nhất trong bộ Selenium.*
        + *Cho phép Record / Playback “Thu và chạy lại ” một test script.*
        + *Là một add – on hỗ trợ Firefox phiên bản 2.0 trở lên. Cung cấp một giao diện dễ sử dụng để phát triển và chạy trường hợp kiểm thử cá nhân, bộ kiểm tra toàn bộ.*
        + *Cho phép lưu test case đã Record dưới nhiều ngôn ngữ lập trình khác nhau như Java, PHP, C#, Ruby…*
      * *Selenium Remote Control*
        + *Là công cụ kiểm tra cho phép bạn sử dụng một ngôn ngữ lập trình để chỉnh sửa, cải tiến linh động và mở rộng trong việc phát triển logic thử nghiệm.*
        + *Có thể nhận các test script được thu bởi Selenium IDE. Cho phép cải chỉnh sửa cải tiến linh động bằng ngôn ngữ lập trình khác nhau. Sau đó khởi động một trình duyệt Web để kiểm tra trực tiếp trên đó.*
        + *Cung cấp một API và thư viện cho mỗi ngôn ngữ lập trình được hỗ trợ: HTML, Java, C#, PHP,...*
      * *Selenium WebDriver*
        + *Là một bản nâng cấp của Selenium RC cho phép gửi các câu lệnh trực tiếp đến trình duyệt và truy xuất kết quả.*
        + *Được thiết kế để cung cấp một giao diện lập trình ngắn gọn đơn giản hơn ngoài việc giải quyết một số hạn chế trong các API Selenium RC.*
        + *Được phát triển tốt hơn để hỗ trợ cho các trang Web động. Mục đích của WebDriver là hỗ trợ cho các vấn để kiểm thử Web – App hiện nay.*
      * *Selenium Gird*
        + *Hỗ trợ cho Selenium RC và Selenium WebDriver.*
        + *Selenium Grid cho phép người dùng thực thi kiểm thử song song trên nhiều máy khác nhau với nhiều trình duyệt khác nhau.*
        + *Có 2 lý do để sử dụng Selenium Gird là:*
          - *Để kiểm thử trên nhiều trình duyệt, nhiều phiên bản của trình duyệt, và các trình duyệt chạy trên hệ điều hành khác nhau.*
          - *Để giảm thời gian hoàn thành của việc kiểm thử.*
2. *Jenkin*
   1. *Mục đích*
      * *Là một ứng dụng web application mã nguồn mở đóng vai trò máy chủ build và test của hệ thống tích hợp liên tục*
      * *Vai trò của Jenkin trong CI*

**

* + - * *Nhờ vào việc hỗ trợ đa nền tảng, công nghệJenkins giúp khép kín quy trình phát triển phầnmềm một cách tự động theo mô hình Agile nóichung và hệ thống tích hợp liên tục nói riêng.*
      * *Với hơn 1200 plugin và được phát triển thêm khôngngừng Jenkins đã trở thành trợ thủ đắc lực cho hệthống CI.*
  1. *Hỗ trợ feature*
     + *Check out Source code*
       - *Git*
       - *Svn*
       - *Cvs*
       - *Perforce*
       - *…*
     + *Hỗ trợ nhiều công nghệ phát triển phần mềm*
       - *Java*
       - *.NET*
       - *Php*
       - *Shell script*
     + *Test đa dạng*
       - *Junit*
       - *Testing*
       - *Checkstyles*
       - *findBugs*
     + *Notify Developers*
       - *Twitter*
       - *Email*
       - *Rss*
       - *Android/ iphone*

# **So sánh sự khác biệt giữa các tool, tool hỗ trợ những gì (Tóm tắt)**

Automation Test là việc sử dụng một phần mềm để quản lý một quá trình kiểm thử.

Công việc chính của chúng ta là tạo ra các điều kiện kiểm thử, điều khiển quá trình kiểm thử và báo cáo kết quả kiểm thử

Một cách khái chung là test tự động bao gồm tự động quá trình chạy bằng tay quá trình mà chúng ta kiểm thử.

**Ưu điểm:**

+ Kỹ thuật viên ít can thiệp

+ Giảm thời gian và công sức cho kỹ thuật viên

**Khuyết điểm:**

+ Tốn chi phí tạo script

+ Tốn chi phí bảo trì các test script

+ Đòi hỏi các kỹ thuật viên có kỹ năng tạo script cao

|  |  |
| --- | --- |
| **Test Tool** | **Tính năng** |
| **Selenium** | * + Là framework đơn giản   + Cho phép record/Playback đoạn test scrip dưới nhiều ngôn ngũ khác nhau như Java, C#, Python, Ruby   + Add-on hỗ trợ Firefox phiên bản 2.0 trở lên |
| **SOAP UI** | * + Kiểm thử chức năng:   + Một công cụ mạnh mẽ cho phép tester viết Functional API Tests trong SoapUI   + Hỗ trợ tính năng kéo-thả mà làm tăng tốc độ phát triển scriptHỗ trợ gỡ lỗi và cho phép tester phát triển data driven tests.   + Kiểm thức bảo mật: Ngăn chặn SQL Injection để bảo đảm cơ sở dữ liệu   + Kiểm thức tải : thử khả năng chịu tải của một ứng dụng web sử dụng loadUI. Tích hợp với các công cụ tự động khác: Maven, HUDSON, Junit, Apache - Ant |
| **Mockito** | Kiểm tra các chức năng của một lớp độc lập  cho phép tạo ra các đối tượng đôi kiểm tra (đối tượng giả) trong unit tests cho mục đích phát triển Test-driven (TDD) hay Behavior Driven Development |
| **Gradle** | Quản lí các phụ thuộc (dependency) rất tốt (dựa trên Apache Ivy)  Sử dụng DST,Groovy là build script  Là một mô hình phong phú để miêu tả dự án |

**SO SÁNH CÁC LOẠI TOOL**

1. **Giống nhau**

Đều là những công cụ kiểm thử ứng dụng web tự động, là những công cụ mã nguồn mở, có thể triển khai trên đa nền tảng như: Windows, Ubuntu,…

1. **Khác nhau**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tool/Tiêu chí** | **SOAP UI** | **Sahi** | **Watir** | **Selenium** |
| Nền tảng | - có trình thuật sĩ (Wizard Setup) riêng để cài chương trình, sau khi cài sử dụng như các chương trình windows bình thường (sử dụng tệp .exe) | - Chạy trên nhiều trình duyệt khác nhau (IE,FF,Chrome,Opera,Safari) - Sử dụng các thuật toán khác nhau đê xác định các phần tử mà con người dễ dàng nhận thấy đc | Chạy trên các trình duyệt Firefox, Chrome, IE, Opera | -Chạy trên nhiều trình duyệt ( trừ IDE có record trên firefox) |
| Ngôn ngữ | - Groovy, JavaScript | Sahi Script, Java, Ruby Sahi Script, | ASP.Net, JSP,PHP, Rails | Java, Ruby, Perl, Python, C# |
| Tính dễ sử dụng | - Giao diện trực quan, dễ nắm bắt - cần có kiến thức chuyên sâu về API để có thể sử dụng thành thạo | - Cần cài đặt, thời gian từ 10-30 phút, phụ thuộc vào lúc cài đặt java - Hỗ trợ APIs cho các task phức tạp | Hỗ trợ test case đơn giản, linh hoạt, dễ đọc | - Thời gian cài đặt dưới 5 phút vì sử dụng plugin của firefox - Cần kiến thức ngôn ngữ lập trình |
| Tính ổn định của script và dễ dàng bảo trì | ổn định | - Không phụ thuộc XPath - Sử dụng \_near, and \_in | Dễ bảo trì | - Chạy test script trên background - Khó khăn cho tester trong việc hiểu và sửa lỗi |