**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

****

**ĐỖ TRUNG HIẾU - 12520135**

**TRẦN BÌNH MINH - 12520263**

**ĐỒ ÁN CHUYÊN NGÀNH**

**SỬ DỤNG GIỌNG NÓI ĐỂ ĐIỀU KHIỂN TRÌNH DUYỆT CHROME TRÊN ANDROID**

**GIẢNG VIÊN HƯỚNG DẪN**

**ThS Nguyễn Công Hoan**

**TP. HỒ CHÍ MINH, 06 - 2016**

**LỜI MỞ ĐẦU**

Ngày nay, cùng với sự phát triển của kinh tế và xã hội, công nghệ thông tin phát triển càng ngày càng mạnh mẽ, nhu cầu của người tiêu dùng càng ngày càng cao không chỉ sử dụng thao tác các ứng dụng bằng tay, các thao tác bình thường mà người dùng còn muốn hướng tới việc giảm thiểu công việc cho đôi tay bằng việc thao tác với các bộ phận khác: ví dụ như thao tác ứng dụng bằng giọng nói, bằng mắt với các ứng dụng hoặc các trang báo điện tử đọc tin tức thời sự, vv.

Vì vậy công nghệ nhận diện giọng nói là một công nghệ khá mới mẻ. Chúng ta có thể áp dụng vào trong đời sống văn hóa xã hội.

**LỜI CẢM ƠN**

Trong suốt thời gian thực hiện đề tài, nhóm chúng em đã nhận được rất nhiều sự giúp đỡ từ các thầy cô của trường Đại học Công nghệ thông tin, đặc biệt là thầy Nguyễn Công Hoan– người đã trực tiếp giảng dạy và hướng dẫn chúng em thực hiện đề tài này.

Nhóm chúng em cũng không quên gửi lời cảm ơn sâu sắc đến bạn bè và gia đình – những người luôn kề vai sát cánh ủng hộ chúng em trong suốt quá trình thực hiện đề tài này.

Mặc dù nhóm chúng em đã cố gắng hết mình nhưng do khả năng có hạn nên chắc chắn đề tài này sẽ không thể tránh khỏi những sai sót. Chúng em rất mong sẽ nhận được những góp ý quý báo từ các thầy cô để đề tài này được hoàn chỉnh hơn.

**Nhóm sinh viên thực hiện:**

**Trần Bình Minh**

**Đỗ Trung Hiếu**

**Tháng 6, năm 2016**

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**MỤC LỤC**

[**Chương I. MỞ ĐẦU** 5](#_Toc456152111)

[1.1. Lý do chọn đề tài 6](#_Toc456152112)

[1.2. Mục đích 6](#_Toc456152113)

[1.3. Đối tượng 6](#_Toc456152114)

[1.4. Phạm vi đề tài 6](#_Toc456152115)

[**Chương II. TỔNG QUAN** 7](#_Toc456152116)

[2.1. Phần mềm đã có 7](#_Toc456152117)

[2.2. Vấn đề cần tập trung 7](#_Toc456152118)

[**Chương III. CƠ SỞ LÝ THUYẾT** 8](#_Toc456152119)

[3.1 Công nghệ Android 8](#_Toc456152120)

[**3.1.1 Hoàn cảnh ra đời** 8](#_Toc456152121)

[**3.1.2 Các phiên bản Android** 9](#_Toc456152122)

[3.2 Ngôn ngữ lập trình Java 12](#_Toc456152123)

[**Chương IV. PHƯƠNG PHÁP VÀ TRIỂN KHAI THỰC HIỆN** 16](#_Toc456152124)

[4.1 Phương pháp 16](#_Toc456152125)

[4.2 Kiến trúc phần mềm 18](#_Toc456152126)

[4.3 Triển khai 18](#_Toc456152127)

[**Chương V. KẾT LUẬN** 18](#_Toc456152128)

[5.1 Kết quả đạt được 18](#_Toc456152129)

[5.2 Khó khăn khi thực hiện 19](#_Toc456152130)

[5.3 Bài học kinh nghiệm 19](#_Toc456152131)

[**Chương VI. HƯỚNG PHÁT TRIỂN** 19](#_Toc456152132)

[6.1 Hướng phát triển 19](#_Toc456152133)

[6.2 Tài liệu tham khảo 20](#_Toc456152134)

**Chương I. MỞ ĐẦU**

* 1. **Lý do chọn đề tài**
* Nhìn thấy được sự phát triển mạnh mẽ của mảng mobile đặc biệt sự phát triển mạnh mẽ của các thiết bị ngày càng tinh vi sắc sảo đòi hỏi kéo theo sự phát triển của các ứng dụng đi kèm ngày càng phải thông minh mạnh mẽ giúp ích cho con người giảm thiểu sức lao động, tư duy nhanh nhạy hơn.
* Thấy được số lượng smartphone tăng đột biến, nhu cầu người dùng tăng mạnh, muốn phát triển và mở ra một hướng mới trong mảng mobile. Đó cũng lý do nhóm chọn đề tài “**SỬ DỤNG GIỌNG NÓI ĐỂ ĐIỀU KHIỂN TRÌNH DUYỆT CHROME TRÊN ANDROID**”.
  1. **Mục đích**
* Hướng tới một thế giới công nghệ thông minh, hiện đại.
* Có nhiều ứng dụng, game có thể điều khiển bằng giọng nói
* Mở rộng được nhiều nền tảng không chỉ android mà còn nhiều nền tảng khác: iOS, WindowPhone….
  1. **Đối tượng**
* Hướng tới tất cả các đối tượng, mọi tầng lớp chỉ cần có smartphone đều có thể sử dụng.
* Đặc biệt đề tài hướng tới giới trẻ, tầng lớp tri thức của xã hội.
  1. **Phạm vi đề tài**
* Hiện tại đề tài chỉ dừng lại ở mức đồ án môn học với các demo nhỏ.
* Trong thời gian sắp tới đề tài có thể phát triển thành các app dịch vụ, hoặc phát triển trực tiếp trong các sản phẩm của nhóm.

**Chương II. TỔNG QUAN**

**2.1. Phần mềm đã có**

* Qua sự hướng dẫn của thầy nhóm đã có website tham khảo về tài: selendroid.io
* Mã nguồn mở của website có các module:
* Android server: Có thể view nhiều loại thiết bị khác nhau
* Selendroid Client: bộ thư viện java phía client.
* Selendroid Common:
* Selendroid Grid Plugin
* Selendroid Server Common
* Selendroid Server: chạy bên cạnh app của bạn trên thiết bị Android
* Selendroid Standalone: quản lý thiết bị Android bởi viêc cài đặt module Selendroid Server dưới app test.
* Selendroid test app

2.2. **Vấn đề cần tập trung**

* Cần tập trung vào một module trong project selendroid-testapp quan tâm đến việc có can thiệp vào Google Chrome trong Android, có thể can thiệp vào kiến trúc của hệ điều hành Android.
* Giải quyết vấn đề: có thể cắt đường dẫn gọi HTTP và đường dẫn gói dịch vụ và App under Test hay không.
* Có gọi được danh sách các website hay không.
* Có thực hiện được các thao tác cơ bản trên một site nào đó viết các đoạn code thực hiện thao tác này.
* Thực hiện việc chuyển các hành động này từ thao tác sang giọng nói.

**Chương III. CƠ SỞ LÝ THUYẾT**

**3.1 Công nghệ Android**

Ngày nay hệ điều hành Android đã trở nên quá quen thuộc với số lượng thiết bị sử dụng đứng đầ, chiếm gần 50% số lượng người dùng và không ngừng tăng trưởng. Vậy điều gì đã khiến cho sản phẩm của gã khổng lồ google thành công đến vậy? Hay trước đó, nó đã được ra đời ra sao và trưởng thành thế nào?

Tổng quan về **hệ điều hành Android**.

**3.1.1 Hoàn cảnh ra đời**

Khi nghành công nghiệp **điện thoại di động** sang một trang mới, một hệ điều hành mở có thể dùng chung cho nhiều hãng điện thoại với những tùy chọn riêng biệt là một miếng mồi béo bở mà Google nhìn thấy đầu tiên. Dưới đây là những cột mốc đáng nhớ trong quá trình hình thành và phát triển của hệ điều hành Android.

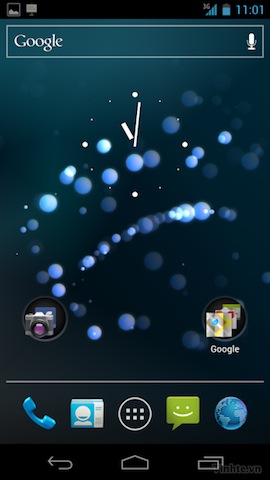


* 10/2013, Android (inc) ra đời như một hãng phần mềm, với mục tiêu tạo ra những thiết bị thông minh, đáp ứng nhu cầu của người dùng
* 8/2005, gã khổng lồ google mua lại Android với toàn bộ nhân viên.
* 11/2007, Open Handset Alliance – Liên minh thiết bị cầm tay mở rộng ra đời với các thành viên Texas Instruments, Tập đoàn Broadcom, Google, HTC, Intel, LG, Tập đoàn Marvell Technology, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel và T-Mobile
* 10/2008, hệ điều hành Android đã chính thức trở thành phần mềm mã nguồn mở.
* 11/2008, Liên minh OHA ra mắt gói phát triển phần mềm Android SDK cho nhà lập trình. Theo đó, các công ty thứ ba được phép thêm những ứng dụng của riêng của họ vào Android và bán chúng mà không cần phải hỏi ý kiến Google.
* 12/2008, có thêm 14 thành viên mới gia nhập dự án Android được công bố, gồm có ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, và Vodafone Group Plc.

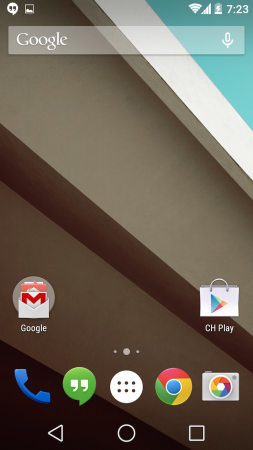
**3.1.2 Các phiên bản Android**

Phiên bản đầu tiên của hệ điều hành Android ra đời vào tháng 9/2008 và tháng 2/2009 không có tên gọi chính thức, sau này, những phiên bản tiếp theo ra đời với tên gọi là những món tráng miệng với vần đầu của bảng Anphabet C-D-E-F-G-H-I…

* Android Cupcake 1.5, 4/2009: Phiên bản này có một số tính năng đáng chú ý như: khả năng ghi lại và xem video thông qua chế độ máy ghi hình, tải video lên YouTube và ảnh lên Picasa trực tiếp từ điện thoại, tích hợp bàn phím ảo với khả năng đoán trước văn bản, tự động kết nối với một thiết bị Bluetooth trong một khoảng cách nhất định, các widget và thư mục mới có thể cài đặt linh động trên màn hình chủ.
* Android Donut 1.6, 9/2009: Phiên bản này giúp Nâng cao trải nghiệm trên kho ứng dụng Android Market, tích hợp giao diện tùy biến cho phép người dùng xóa nhiều ảnh cùng lúc, nâng cấp Voice Search, nâng cấp khả năng tìm kiếm bookmarks, history, contacts và web trên màn hình chủ, bước đầu hỗ trợ màn hình độ phân giải WVGA.
* Android Eclair 2.0 + 2.1, 10/2009. Phiên bản này có sự cải thiện rõ rệt trong giao diện người dùng, tối ưu hóa tốc độ phần cứng, hỗ trợ nhiều kích cỡ và độ phân giải màn hình hơn, thay đổi giao diện duyệt web và hỗ trợ chuẩn HTML5, Exchange ActiveSync 2.5, nâng cấp Google Maps 3.1.2, camera zoom kĩ thuật số tích hợp đèn flash, nâng cấp bàn phím ảo và kết nối Bluetooth 2.1.



* 2.2 Android Froyo 5/2010: Phiên bản này chú trọng nâng cấp tốc độ xử lí, giới thiệu engine Chrome V8 JavaScript, hỗ trợ Adobe Flash10.1, thêm tính năng tạo điểm truy cập Wi-Fi. Một tính năng đáng chú ý khác hỗ trợ chuyển đổi nhanh chóng giữa các ngôn ngữ và từ điển trên bàn phím đồng thời cho phép cài đặt và cập nhật ứng dụng ở các thiết bị mở rộng bộ nhớ. Một trong những smartphone đầu tiên chạy phiên bản Android 2.2 Froyo là LG Optimus One.
* Android Gingerbread 2.3, 12/2010: Phiên bản này đã nâng cấp đáng kể giao diện người dùng, cải thiện bàn phím ảo, thêm tính năng copy/paste, hỗ trợ công nghệ giao tiếp tầm sóng ngắn NFC, hỗ trợ chuẩn video WebM và nâng cao tính năng copy–paste. Cùng với phiên bản Gingerbread, Google cũng ra mắt điện thoại đầu tiên của hãng sử dụng nền tảng này là Google Nexus S.
* Android Honeycomb 3.0, 2/2011: Đây là phiên bản hệ điều hành dành riêng cho máy tính bảng tablet với giao diện mới tối ưu hóa cho tablet, từ các thao tác đều phụ thuộc màn hình cảm ứng (như lướt web, duyệt mail..). Honeycomb hỗ trợ bộ xử lí đa nhân và xử lý đồ họa đồng thời hỗ trợ nhiều màn hình home khác nhau, cho phép người dùng dễ dàng tùy biến giao diện nếu muốn.
* Android 4.0 Ice Cream Sandwich, cuối 2011. Đây cũng là lần đầu tiên Google hợp nhất hệ điều hành dành cho smartphone và cho máy tính bảng vào làm một. Android 4.0 cũng nhắm đến việc duyệt web nhanh hơn, tối ưu hóa hiệu suất hoạt động của thiết bị, kéo dài thời gian dùng pin…
* Android Jelly Bean 4.1 +4.2, 2012: quan trọng hơn hết của Jelly Bean không phải là về giao diện hay ứng dụng mới mà về Project Butter giúp mang lại độ mượt chưa từng có cho Android. Theo như giải thích của Google, độ mượt này có được là nhờ vào dự án Butter
* Android 4.3 Jelly Bean 4.3, 24/4/2013: đi kèm những tính năng mới như hỗ trợ kết nối Bluetooth Smart, bộ API OpenGL ES 3.0, bổ sung tính năng sử dụng Wi-Fi để định vị ngay cả khi người dùng tắt kết nối này đi cùng nhiều thay đổi lớn nhỏ khác.



* Android 4.4 KitKat, 15/10/2013 với các cải tiến mới Chế độ toàn màn hình – Immersive Mode, Hiệu ứng chuyển cảnh màn hình -Transition Manager, Storage Access Framework, Chromium WebView, NFC,Cổng hồng ngoại – Infrared Blasters …
* Android L, 26/06/2014 với những cải tiến về chất lượng pin, giao diện thanh thoát nhẹ nhàng theo thiết kế Material Design, Android Runtime (ART) đã được sử dụng mặc định
* Android 5.0 Lollipop: Thiết kế Material Design mới

Điều đầu tiên bạn sẽ nhận thấy sau khi cập nhật lên Android 5.0 Lollipop là giao diện được thiết kế lại. Thiết kế mang phong cách phẳng Material Design của Google khiến hệ điều hành trông nhẹ nhàng và thoáng đãng hơn. Giờ đây thông báo hiện thị trực tiếp trên màn hình khóa, ẩn các nội dung nhạy cảm trên thông báo, tuổi thọ pin dài hơn với chế độ tiết kiệm pin, chúng ta có thể chỉnh chế độ để không bi quấy rầy.Thêm các thiết bị tin cậy, có thể tìm kiếm settings, sử dụng chế độ khách với bạn bè, ghim ứng dụng, truy cập quick setting mới, nhanh chóng kiểm tra việc sử dụng dữ liệu, hiệu suất cải thiện, độ sáng tương ứng phù hợp với chế độ sáng tự động, văn bản có độ tương phản cao, bạn có thể biến đổi màu, ứng dụng vẫn ở trong overview sau khi khởi động, sử dụng speak với Google ở bất cứ đâu.

* + Android 6.0 [Marshmallow](https://developer.android.com/about/versions/marshmallow/index.html): quản lý app trược tiếp, hỗ trợ các developer với các lớp trong View hoặc Activity Object.

**3.2 Ngôn ngữ lập trình Java**

Giới thiệu

* Là một trong những ngôn ngữ lập trình mạnh và được sử dụng đông đảo trong phát triển phần mềm, các trang web, game hay ứng dụng trên các thiết bị di động, ngôn ngữ lập trình Java đã và đang trở nên lớn mạnh hơn bao giờ hết. Bài viết này xin được trình bày một vài đặc điểm tiêu biểu của ngôn ngữ lập trình Java để các bạn có được một cái nhìn tổng quát về ngôn ngữ này.
* Trong quá trình tìm hiểu một vài ngôn ngữ mới, tôi bắt gặp được một thứ gọi là Java Virtual Machine. Đây là một phần của công nghệ Java, nó có thể thông dịch file Byte code ra Machine code để giúp chương trình có thể chạy trên nhiều nền tảng khác nhau.

Lịch sử phát triển ngôn ngữ lập trình Java

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

Java được tạo ra với tiêu chí "Viết (code) một lần, thực thi khắp nơi" ("Write Once, Run Anywhere"  - WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó .

wikipedia.org

Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.

Đặc điểm của ngôn ngữ lập trình Java

**1. Tựa C++, hướng đối tượng hoàn toàn**

Trong quá trình tạo ra một ngôn ngữ mới phục vụ cho mục đích chạy được trên nhiều nền tảng, các kỹ sư của Sun MicroSystem muốn tạo ra một ngôn ngữ dễ học và quen thuộc với đa số người lập trình. Vì vậy họ đã sử dụng lại các cú pháp của C và C++.

Tuy nhiên, trong Java **thao tác với con trỏ bị lược bỏ** nhằm đảo bảo tính an toàn và dễ sử dụng hơn. Các thao tác overload, goto hay các cấu trúc như struct và union cũng được loại bỏ khỏi Java. Các vấn đề này xin được làm rõ ở những bài viết sau.

**2. Độc lập phần cứng và hệ điều hành**

Một chương trình viết bằng ngôn ngữ Java có thể chạy tốt ở nhiều môi trường khác nhau. Gọi là khả năng "**cross-platform**”. Khả năng độc lập phần cứng và hệ điều hành được thể hiện ở 2 cấp độ là cấp độ **mã nguồn** và cấp độ **nhị phân**.

**Ở cấp độ mã nguồn:** Kiểu dữ liệu trong Java nhất quán cho tất cả các hệ điều hành và phần cứng khác nhau. Java có riêng một bộ thư viện để hỗ trợ vấn đề này. Chương trình viết bằng ngôn ngữ Java có thể biên dịch trên nhiều loại máy khác nhau mà không gặp lỗi.

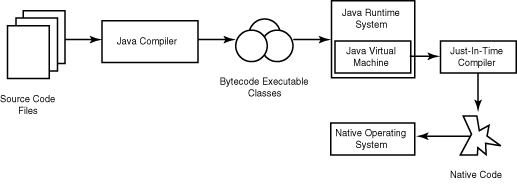
**Ở cấp độ nhị phân:** Một mã biên dịch có thể chạy trên nhiều nền tảng khác nhau mà không cần dịch lại mã nguồn. Tuy nhiên cần có Java Virtual Machine để thông dịch đoạn mã này.

**3. Ngôn ngữ thông dịch**

Ngôn ngữ lập trình thường được chia ra làm 2 loại (tùy theo các hiện thực hóa ngôn ngữ đó) là ngôn ngữ thông dịch và ngôn ngữ biên dịch. Bạn có thể xem thêm tại đây - [Phân Biệt Compile Và Interpret](http://www.stdio.vn/articles/read/23/phan-biet-compile-va-interpret).

Ngôn ngữ lập trình Java thuộc loại ngôn ngữ thông dịch. Chính xác hơn, Java là loại ngôn ngữ **vừa biên dịch vừa thông dịch**. Cụ thể như sau

Khi viết mã, hệ thống tạo ra một tệp .java. Khi biên dịch mã nguồn của chương trình sẽ được biên dịch ra mã **byte code**. Máy ảo Java (Java Virtual Machine) sẽ thông dịch mã byte code này thành **machine code**  (hay native code) khi nhận được yêu cầu chạy chương trình.



**Ưu điểm** : Phương pháp này giúp các đoạn mã viết bằng Java có thể chạy được trên nhiều nền tảng khác nhau. Với điều kiện là JVM có hỗ trợ chạy trên nền tảng này.

**Nhược điểm** : Cũng như các ngôn ngữ thông dịch khác, quá trình chạy các đoạn mã Java là chậm hơn các ngôn ngữ biên dịch khác (tuy nhiên vẫn ở trong một mức chấp nhận được).

**4. Cơ chế thu gom rác tự động**

Khi tạo ra các đối tượng trong Java, JRE sẽ tự động cấp phát không gian bộ nhớ cho các đối tượng ở trên heap.

Với ngôn ngữ như C \ C++, bạn sẽ phải yêu cầu hủy vùng nhớ mà bạn đã  cấp phát, để tránh việc thất thoát vùng nhớ. Tuy nhiên vì một lý do nào đó, bạn không hủy một vài vùng nhớ, dẫn đến việc thất thoát và làm giảm hiệu năng chương trình.

Ngôn ngữ lập trình Java hỗ trợ cho bạn điều đó, nghĩa là bạn không phải  tự gọi hủy các vùng nhớ. Bộ thu dọn rác của Java sẽ theo vết các tài nguyên đã được cấp. Khi **không có tham chiếu** nào đến vùng nhớ, bộ thu dọn rác sẽ tiến hành thu hồi vùng nhớ đã được cấp phát.

**5. Đa luồng**

Java hỗ trợ lập trình đa tiến trình (**multithread**) để thực thi các công việc đồng thời. Đồng thời cũng cung cấp giải pháp đồng bộ giữa các tiến trình (giải pháp sử dụng priority...).

**6. Tính an toàn và bảo mật**

**Tính an toàn**

Ngôn ngữ lập trình Java yêu cầu chặt chẽ về kiểu dữ liệu.

* Dữ liệu phải được khai báo tường minh.
* Không sử dụng con trỏ và các phép toán với con trỏ.
* Java kiểm soát chặt chẽ việc truy nhập đến mảng, chuỗi. Không cho phép sử dụng các kỹ thuật tràn. Do đó các truy nhập sẽ không vượt quá kích thước của mảng hoặc chuỗi.
* Quá trình cấp phát và giải phóng bộ nhớ được thực hiện tự động.
* Cơ chế xử lý lỗi giúp việc xử lý và phục hồi lỗi dễ dàng hơn.

**Tính bảo mật**

Java cung cấp một môi trường quản lý chương trình với nhiều mức khác nhau.

* Mức 1 : Chỉ có thể truy xuất dữ liệu cũng như phương phức thông qua giao diện mà lớp cung cấp.
* Mức 2 : Trình biên dịch kiểm soát các đoạn mã sao cho tuân thủ các quy tắc của ngôn ngữ lập trình Java trước khi thông dịch.
* Mức 3 : Trình thông dịch sẽ kiểm tra mã byte code xem các đoạn mã này có đảm bảo được các quy định, quy tắc trước khi thực thi.
* Mức 4: Java kiểm soát việc nạp các lớp vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống.

Java Virtual Machine

Nói về Java mà không nói đến JVM (Java Virtual Machine) thì quả là một thiếu sót. Tuy nhiên trong phạm vi bài viết này, xin được tóm gọn lại một số đặc điểm cơ bản của chiếc máy ảo Java này. Các vấn đề cụ thể sẽ được đề cập ở những bài viết sau.

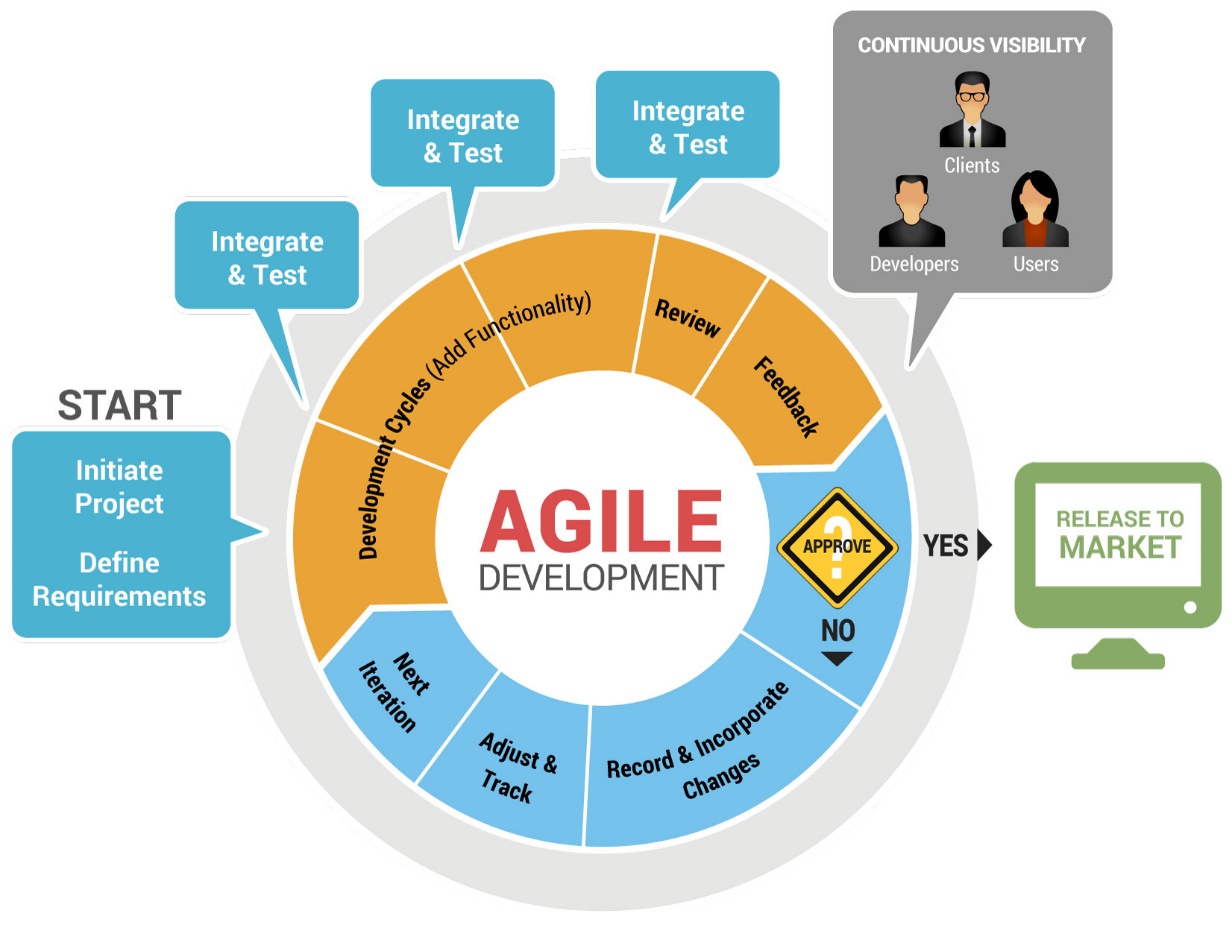
* Máy ảo Java là phần mềm giả lập máy tính, nó tập hợp các lệnh logic để xác định hoạt đông của máy.
* Có thể xem nó như là một hệ điều hành thu nhỏ.
* JVM chuyển mã byte code thành machine code tùy theo môi trường tương ứng (gọi là khả năng khả chuyển)
* JVM cung cấp môi trường thực thi cho chương trình Java (gọi đó là khả năng độc lập với nền).
* Sun MicroSystem chịu trách nhiệm thiết kế, phát triển các máy ảo Java chạy trên các hệ điều hành cũng như kiến trúc phần cứng khác nhau. Điều này cho thấy có khá nhiều loại máy ảo Java.

**Chương IV. PHƯƠNG PHÁP VÀ TRIỂN KHAI THỰC HIỆN**

**4.1 Phương pháp**

Trong dự án này, nhóm đã sử dụng quy trình phát triển phần mềm Agile.

Quy trình phát triển phần mềm Agile là một nhóm của các phương thức phát triển phần mềm mà trong đó các yêu cầu và giải pháp tiến triển thông qua sự cộng tác giữa nhóm tự tổ chức và nhóm đa chức năng. Nó thúc đẩy kế hoạch thích nghi, sự phát triển tiến hóa, phân phối sớm, cải thiện liên tục và khuyến khích các phản hồi nhanh và linh hoạt để thay đổi.



*Hình 3.3: Quy trình phát triển phần mềm Agile*

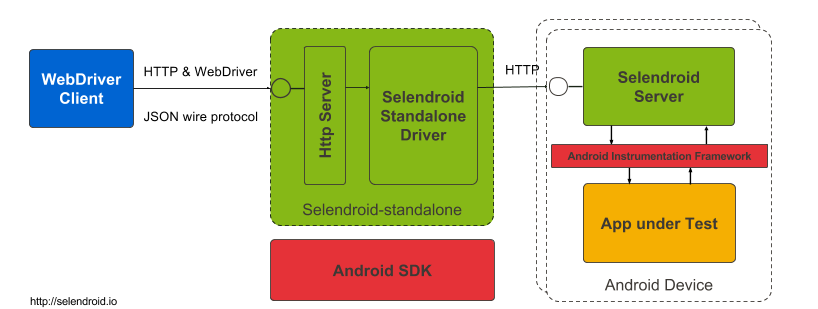
Tuyên ngôn chính của Agile:

* Cá nhân và sự tương tác giữa người với người được đặt trên quy trình và công cụ: tự tổ chức và năng động là cực kỳ quan trọng, như là sự tương tác về việc lập trình theo cặp.
* Làm việc chú trọng vào phát triển phần mềm hơn là viết tài liệu đầy đủ: làm việc với phần mềm cực kỳ hữu ích và đáng hoan nghênh hơn là chỉ trình chiếu tài liệu tới khách hàng trong cuộc họp.
* Cộng tác với khách hàng hơn là thương lượng hợp đồng: yêu cầu không thể được tập hợp đầy đủ ngay từ đầu của chu trình phát triển phần mềm, chính vì thế sự tham gia của các bên liên quan là cực kỳ quan trọng.
* Tiếp nhận phản hồi để thay đổi hơn là bám sát kế hoạch đề ra: phương pháp agile tập trung vào việc phản hồi nhanh để thay đổi và phát triển tiếp.

Tuyên ngôn của Agile dựa trên 12 nguyên lý:

* Sự hài lòng của khách hàng về việc phân phối nhanh.
* Sẵn sàng thay đổi yêu cầu, thậm chí là trễ trong việc phát triển.
* Phần mềm hoạt động được phân phối thường xuyên (theo hàng tuần).
* Cộng tác hàng ngày và gần gũi giữa những người kinh doanh và nhà phát triển.
* Dự án được xây dựng từ những cá nhân năng động, những người thực sự đáng tin cậy.
* Hội thoại mặt đối mặt là cách thức tốt nhất trong giao tiếp.
* Phần mềm làm việc là thước đo cơ bản của tiến trình.
* Sự phát triển ổn định, có thể duy trì một tốc độ nhất định.
* Liên tục chú ý tới sự tuyệt vời về kỹ thuật và thiết kế tốt.
* Đơn giản – nghệ thuật tối đa hóa số lượng công việc chưa hoàn thành – là cần thiết.
* Nhóm tự tổ chức.
* Thích ứng thường xuyên với sự thay đổi hoàn cảnh.
* Mỗi tuần làm việc sẽ làm một sprint sau khi hoàn thành sprint sẽ báo cáo với thầy hướng dẫn.

**4.2 Kiến trúc phần mềm**



**4.3 Triển khai**

a. Sử dụng gói Selendroid-standalone-0.15.0-with-dependencies để can thiệp vào hệ điều hành Android và trình duyệt Google Chrome trên Android.

b. Cắt đường dẫn liên lạc mà vẫn can thiệp vào

-Không cần sử dung gói Selendroid-standalone-0.15.0-with-dependencies mà vẫn có thể can thiệp và chạy Google Chrome lên.

-Cố định danh sách các website có thể truy cập và có thể gọi nó

c.Sử dụng giọng nói để gọi các lệnh trên website ví dụ như: back, forward, home…

**Chương V. KẾT LUẬN**

**5.1 Kết quả đạt được**

* Nhóm đã thực hiện được căn bản những mục tiêu ban đầu đề ra: có thể can thiệp vào hệ điều hành Android và trình điều khiển trên Android là Google Chrome bằng gói apk.
* Không dùng đến gói apk vẫn có thể điều khiển được Google Chrome trên android
* Có thể thao tác gọi trình duyệt Google Chrome trên Android gọi một số trên website cố định và thao tác với website đó.
* Sử dụng giọng nói trong thao tác các chức năng cơ bản trong website: back, forward, home, next, read content.

**5.2 Khó khăn khi thực hiện**

* Khi tiếp cận với đề tài bị mất phương hướng(có thể) vì lúc đầu sử dụng mã nguồn của người khác.
* Sắp xếp thời gian giữa thầy và trò còn nhiều khó khăn, gián đoạn nhiều thời gian.
* Khi tiếp cận công nghệ mới phải tự tìm hiểu
* Chưa xác định được mục đích khi nhận được đề tài lúc ban đầu.

**5.3 Bài học kinh nghiệm**

* Khi nhận bất cứ đề tài vấn đề khoa học nào phải xác nhận rõ ràng mục tiêu hướng tới của đề tài, lên kế hoạch chi tiết sau khi nhận được đề tài.
* Lập ra các kế hoạch dự phòng cho trường hợp xấu nhất.
* Ghi rõ ràng các khó khăn và các khắc phục cho từng vấn đề khi thực hiện bất cứ đề tài nào.
* Gặp khó khăn về nhận diện giọng nói trong đề tài vì đây là công nghệ khá mới nhóm tiếp cận.

**Chương VI. HƯỚNG PHÁT TRIỂN**

**6.1 Hướng phát triển**

* Với đề tài phát triển đến hiện tại, chúng ta có thể viết app hoặc game sử dụng dịch vụ này.
* Chúng ta có thể viết các dịch vụ tích hợp vào trong các trang web, bán các dịch vụ cho người sử dụng.
* Người dùng có thể lưu đường dẫn và giọng nói.
* Viết các game dành cho trẻ em sử dụng công nghệ này.

**6.2 Tài liệu tham khảo**

* [1] <http://selendroid.io/setup.html>.
* [2] <https://developer.android.com/guide/components/intents-filters.html?hl=en>
* [3] <https://www.android.com/>
* [4] <http://android.vn/>
* [5] <http://vietandroid.com/>
* [6] <http://gsm.vn/forums/android.350/>