

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM**  
**KHOA CÔNG NGHỆ THÔNG TIN**

-----Ω+Ω-----

**MÔN HỌC: PHÁT TRIỂN PHẦN MỀM NGUỒN MỞ**

**BÁO CÁO YÊU CẦU GIAI ĐOẠN 2**  
**TÌM HIỂU DỰ ÁN FOSS**  
**PHÁT TRIỂN PLUGIN CHO WORDPRESS**

*Giảng viên:*

**Thầy Lê Xuân Định**

**Thầy Đỗ Nguyên Kha**

**Thầy Nguyễn Lê Hoàng Dũng**

**Thầy Trương Phước Lộc**

*Nhóm thực hiện:*      **5T&N**

**Trần Thanh Triều**    1012477

**Nguyễn Huỳnh Tiến** 1012450

**Tống Thành Triệu**    1012478

**Phạm Phúc Thịnh**    1012429

**Trương Minh Thân**    1012416

**Nguyễn Thị Nga**      1012576

## Table of Contents

Table of Contents .....	2
<b>I.Mô tả yêu cầu .....</b>	<b>3</b>
<i>1.Yêu cầu tìm hiểu project .....</i>	<i>3</i>
<i>2. Yêu cầu phát triển plugin cho WordPress .....</i>	<i>3</i>
<b>II.Thực hiện .....</b>	<b>4</b>
<i>1.Pháp lý .....</i>	<i>4</i>
<i>2.Kinh tế.....</i>	<i>5</i>
3. Mã nguồn mở và tính pháp lý của phần mềm.....	10
4.Xã hội.....	12
<b>III.Tìm hiểu cài đặt và phát triển plugin cho Wordpress .....</b>	<b>20</b>
<i>1.Plugin là gì .....</i>	<i>20</i>
<i>2.Tại sao lại phải dùng plugin .....</i>	<i>21</i>
<i>3.Ta cần phải có những kiến thức gì trước khi bắt đầu? .....</i>	<i>21</i>
<i>4.Công cụ cần thiết.....</i>	<i>21</i>
5.Cấu trúc thư mục và tập tin .....	21
6.Tên của plugin .....	22
7.Tên thư mục và tên của các tập tin .....	22
8.Bắt đầu viết code cho plugin .....	22
9.Viết code cho plugin .....	23
Hook, Action và Filter.....	23
Các tập tin ngoại .....	23
Sản phẩm:.....	24

## **I. Mô tả yêu cầu**

### **1. Yêu cầu tìm hiểu project**

**Pháp lý:** Các giấy phép / hợp đồng mà các bên cần cung cấp / đồng ý. Khảo sát cả 3 bên: Users, Contributors, Project. Trong đó, nếu có nhiều loại Users và nhiều loại Contributors thì những giấy phép / hợp đồng tương ứng cho từng loại là gì.

#### **- Kinh tế:**

- o Users được kinh doanh phần mềm lấy từ project này như thế nào, theo những hình thức nào?
- o Contributors được lợi ích kinh tế gì khi tham gia project?
- o Project thu nguồn lợi / hỗ trợ kinh tế từ đâu?

#### **- Xã hội:**

- o Tổ chức cộng đồng như thế nào, gồm những loại thành viên nào? Mỗi loại thành viên có quyền hạn & trách nhiệm gì, tham gia những hoạt động gì?
- o Những định hướng vĩ mô, kế hoạch, quyết định được thực hiện như thế nào?
- o Cộng đồng gồm những thành phần xã hội nào (tình nguyện viên tự do, nhân viên công ty nào, sinh viên, nghiên cứu viên, v.v.). Tỷ lệ nhân lực và tỷ lệ đóng góp của từng thành phần (nếu không có số liệu thống kê thì SV tự ước lượng).

#### **- Kỹ thuật:**

- o Cơ sở hạ tầng được tổ chức như thế nào: tài liệu, diễn đàn, chat, quản lý lỗi, quản lý mã nguồn.
- o Hệ thống phần mềm được thiết kế như thế nào: bởi ai, theo quy trình gì, theo mô hình gì?
- o Hệ thống module trong mã nguồn: phân chia module thế nào, tương tác ra sao?

### **2. Yêu cầu phát triển plugin cho WordPress**

Mã nguồn WordPress sử dụng từ zip (để phát triển plugin) hoặc git/svn (để kiểm tra lỗi)

Yêu cầu chức năng của plugin:

- SV tự đề xuất chức năng cụ thể và yêu cầu giáo viên phụ trách duyệt lại
- Phải bao gồm cả hai thành phần giao diện: front-end (giao diện chính, kết hợp với chức năng blog của WordPress) và back-end (kết hợp giao diện quản lý của admin để quản lý tham của số plugin)
- Sử dụng API của WordPress để tích hợp plugin:
  - o Filter (ít nhất 1): [http://codex.WordPress.org/Plugin\\_API/Filter\\_Reference](http://codex.WordPress.org/Plugin_API/Filter_Reference)
  - o Action (ít nhất 2): [http://codex.WordPress.org/Plugin\\_API/Action\\_Reference](http://codex.WordPress.org/Plugin_API/Action_Reference)
- Sử dụng database thông qua API của WordPress:

[http://codex.WordPress.org/Database\\_API](http://codex.WordPress.org/Database_API) và/hoặc

[https://codex.WordPress.org/Class\\_Reference/wpdb](https://codex.WordPress.org/Class_Reference/wpdb)

- Có giao diện quản lý trong trang admin:

[http://codex.WordPress.org/Creating\\_Options\\_Pages](http://codex.WordPress.org/Creating_Options_Pages)

- Giao diện đơn giản nhưng cần có sử dụng CSS và Javascript

- Plugin cần chứa trong thư mục bao gồm plugin và các tập tin CSS, Javascript và hình ảnh.

## II. Thực hiện

### 1. Pháp lý

- ❖ **Tổng quan:** Wordpress là một hệ thống xuất bản blog viết bằng ngôn ngữ PHP và sử dụng CSDL MySQL. Tiền thân của Wordpress là b2/cafeblog, được phát triển bởi Michel Valdrighi. Tên Wordpress được đề xuất bởi Christine Selleck, một người bạn của nhà phát triển chính Matt Mullenweg.
- ❖ **Giấy phép:** giống như hầu hết các phần mềm, Wordpress được phân phối theo một giấy phép, đó là GNU GPLv2 hoặc phiên bản sau đó (General Public License) từ quỹ phần mềm tự do (Free Software Foundation). Một bản sao chép của giấy phép được bao gồm trong một bản sao chép của Wordpress. Một phần của giấy phép này vạch ra yêu cầu đối với sản phẩm phái sinh, như plugin hoặc themes. Những mã Wordpress của bản phái sinh kế thừa các giấy phép GPL.
- ❖ **GNU GPL:** là giấy phép mã nguồn mở, mã nguồn mở không có nghĩa là bạn View mã nguồn.

Tại sao wordpress chọn được phân phối theo giấy phép GPL: Wordpress phát hành theo giấy phép GPL bởi cả 2 lý do thực tế và lý tưởng. Wordpress được sinh ra từ một đề cập rất tự do trước đó.

#### ▪ Giấy phép và pháp lý cho các đối tượng trong wordpress:

Mọi người được quyền sao chép và phân phối bản sao đúng nguyên văn của giấy phép này, nhưng không được thay đổi.

➤ **Đối với user:**

➤ **Đối với contributor:** một contributor chỉ có thể tạo và chỉnh sửa những bài đăng mà họ sở hữu, nhưng không thể công bố chúng. Khi một trong những bài

đăng của họ đã sẵn sàng để được công bố và đã được sửa đổi, người quản trị cần được thông báo cá nhân bởi người đóng góp để xem xét nó. Hơn nữa, một khi bài đăng của người đóng góp được phê duyệt và công bố bởi người quản trị, thì bài đăng đó có thể không còn được chỉnh sửa bởi chính người đóng góp đó. Người đóng góp không có khả năng tải lên các tập tin hoặc hình ảnh, nhưng họ có thể xem số liệu thống kê của trang web.

- **Contributor agreements:** là thỏa thuận giữa dự án nguồn mở và người đóng góp cho dự án mà đặt ra những gì dự án có thể làm với bản quyền của sự đóng góp: mã, dịch thuật, tài liệu, tác phẩm nghệ thuật, vv. Bản thỏa thuận đóng góp cho sự tự tin rằng người giám hộ đầu ra của dự án có các quyền cần thiết hơn tất cả những người đóng góp để cho phép phân phối sản phẩm theo bất kỳ giấy phép hoặc không theo bất kỳ giấy phép mà phù hợp với những nguyên tắc cụ thể. Các thỏa thuận đóng góp tránh càng xa càng tốt các vấn đề pháp lý tương lai liên quan đến việc đóng góp cá nhân, chẳng hạn như tranh chấp về nguồn gốc hoặc quyền sở hữu các quyền tương ứng.
- **Có 2 loại thỏa thuận người đóng góp:** thỏa thuận chuyển nhượng bản quyền-Copyright assignments agreements(CAA) và thỏa thuận cấp phép bản quyền-Copyright license agreements(CLA). CAA yêu cầu chuyển nhượng do đó chuyển giao bản quyền cho người chủ sở hữu dự án. CLA yêu cầu một giấy phép không thể thu hồi cho phép chủ sở hữu dự án sử dụng sự đóng góp đó.
- **Đối với tác giả:** một tác giả có thể tạo, chỉnh sửa, công bố hoặc xóa chỉ những bài đăng mà họ sở hữu, cũng như tải lên các tập tin và hình ảnh. Tác giả không có quyền truy cập để tạo, chỉnh sửa, hoặc xóa các trang, cũng như không thể chỉnh sửa các bài đăng của những người dùng khác.

## 2.Kinh tế

### ▪ Users :

Kinh doanh dịch vụ nhờ vào ứng dụng của phần mềm lấy từ project.

Sử dụng các phần mềm từ project để phục vụ cho mảng kinh doanh của mình ( ví dụ : phần mềm quản lý hàng hóa được sử dụng để thuận tiện hơn khi quản lý nhờ đó tăng thu nhập ).

Bán các vật phẩm có hình biểu tượng của project.

Những người làm phần mềm mã nguồn mở thu lợi từ đâu: Tất cả những người làm phần mềm mã nguồn mở, tất nhiên là không thể bán phần mềm mã nguồn mở, vì nó là miễn phí, và ai cũng đọc source code được. Nhưng không phải là người làm phần mềm mã nguồn mở viết phần mềm nhằm mục đích từ thiện, cứu vớt 50 triệu trẻ em ở châu Phi hay giúp đỡ một quốc gia nghèo đói như Việt nam.

▪ **Lợi ích của mã nguồn mở:**

- Lợi dụng được chất xám của cộng đồng lập trình viên: Khi một lập trình viên muốn có một chức năng gì đó, ví dụ muốn có một web server, nhưng không muốn bỏ tiền mua, anh ta hoặc một nhóm bạn có thể viết một web server đơn giản, open source nó. Thế là cộng đồng những lập trình viên muốn viết code cho web server có thể download source code về, chỉnh sửa, viết thêm vào, và lâu dần sẽ tạo ra một sản phẩm tốt, vì có sự tham gia và thử nghiệm của một cộng đồng lớn. Apache (so far the best web server in the world) đã ra đời trong hoàn cảnh như thế. IBM, sau khi chết một cách ngu xuẩn với Visual Age, đã quyết định open source Eclipse để lợi dụng chất xám của cộng đồng. Ngày nay, chúng ta thấy Eclipse is one of the best IDE. Và IBM cũng thu lợi bằng cách lấy lại những đóng góp của cộng đồng cho Eclipse để cho vào Web Sphere App Dev Studio, đem bán lấy hàng tỉ đô.
- **Lợi ích về công nghệ:** Promote best practice và tránh Bureaucracy Cộng đồng mã nguồn mở là một cộng đồng lớn với high spirit, kiểm soát chủ yếu bởi những kiến trúc sư phần mềm và lập trình viên có trình độ cao, giàu kinh nghiệm và đam mê lập trình. Nếu không giỏi, tất nhiên chẳng ai tự viết ra được phần mềm hay framework, mà chỉ có đi copy, ăn cắp rồi chỉnh sửa, Việt hóa (như chúng ta thường thấy trong những phần mềm được giải cao của Trí tuệ Việt nam hay Nhân tài đất Việt). Nếu không đam mê, thì chẳng ai sau mười mấy tiếng một ngày làm thân trâu ngựa cho loài chó dê, làm tay sai để quốc xong về làm tay sai cho vợ con, mà sau khi ăn sushi trừ bữa, uống rượu ngoại cảm hơi, vẫn còn tiếp tục dành thời gian ngồi ôm laptop viết phần mềm thay vì ôm gái. Với trình độ, kinh nghiệm, đam mê và sự thông minh, cộng đồng mã nguồn mở luôn luôn cố gắng phát minh công nghệ mới, ứng dụng công nghệ tốt, promote best practice và làm cho sản phẩm ngày càng tốt hơn. Free-spirit của cộng đồng làm cho những phần mềm mã nguồn mở (usual) có

thiết kế tốt hơn và chất lượng tốt hơn phần mềm được làm bởi những tập đoàn tư bản quan liêu thối nát, quản lý không phải bởi những người giỏi làm việc, mà chỉ bởi những con lợn tư bản only good at vibrating bullshit, những thằng không biết về kỹ thuật whatsoever, mà chỉ biết đến revenue, và công thức toán học duy nhất mà chúng biết là  $MBA = BMW$ . Một ví dụ điển hình là JBoss Enterprise có chất lượng, features và performance far more better than IBM Web Sphere App Server.

- **Lợi ích về nghề nghiệp:** Những lập trình viên như Garvin King, Rod Johnson ... viết những open source framework để nhằm mục đích ứng dụng vào việc làm phần mềm cho khách hàng (viết phần mềm bảo hiểm, finance ...) nhanh hơn, có chất lượng cao hơn. Do đó họ có danh tiếng tốt, có nhiều khách hàng hơn. Phần mềm của họ khi được cộng đồng đóng góp, sẽ có nhiều chức năng hơn, chất lượng cao hơn, sử dụng lại vào việc làm phần mềm tốt hơn. Những người đóng góp vào việc viết phần mềm của họ cũng học hỏi được nhiều hơn, sử dụng công cụ thuần thục hơn, có nền tảng để phát minh ra các công nghệ khác. Khách hàng khi thấy việc ứng dụng những open source framework đem lại thời gian phát triển nhanh hơn, chất lượng công việc cao hơn cũng sẵn sàng trả nhiều tiền hơn.
- **Lợi ích về tiền :-** Bán value added features: Tất nhiên là không ai đi bán open source software, nhưng người ta có thể viết thêm các chức năng cho nó, cung cấp thêm giá trị gia tăng. Ví dụ như Linux là free. Setup Linux nói chung là khó. Nhưng công ty Redhat viết thêm anaconda để setup linux dễ dàng hơn, đóng gói vào CD, DVD kèm với tài liệu hướng dẫn và bán. Nên nhớ ở đây, Redhat không bán Linux, mà bán CD kèm với tài liệu và thêm anaconda. Hoặc Redhat thêm enterprise feature vào Linux và bán Redhat Enterprise Linux. Nhưng do GPL, người dùng anaconda và added features của Enterprise phải được cung cấp source code, vì thế, chúng ta mới được download free Redhat Linux và Fedora (free Enterprise Linux) từ redhat.com. Hoặc IBM bán WADS, nhưng license của Eclipse không bắt buộc những gì viết thêm vào phải open source, nên IBM (người Đức gọi IBM là Ich bin müde - I am tired) cứ ung dung bán WADS. d2 - Bán consultant service Không phải phần mềm mã nguồn mở nào cũng dễ dùng. Triển khai Linux Enterprise cho doanh nghiệp, hoặc phát triển J2EE cho những corporation lớn như Boeing, Airbus, Lockheed Martin, Nike, Liz Claiborne... dùng Spring, Hibernate, Acegi,

Tapestry ... không phải là việc mà bọn giáo sư hay trẻ con học Ph.D. ở MIT, Berkeley, Stanford hay Carnegie Mellon ra có thể làm được. Giá consultancy cho những triển khai ứng dụng loại này hiếm khi dưới tiền triệu \$. Ví dụ project Nike trị giá khoảng 35 triệu, project Liz Claiborne khoảng hơn chục triệu. Đây là chưa kể tiền công vài trăm \$ / giờ cho senior developer, senior consultant Ăn độ loại vét đĩa, tiền ăn sushi trừ bữa, uống rượu mạnh cầm hơi, tiền xe Lincoln, máy bay, khách sạn ... d3 - Dùng phần mềm mã nguồn mở để làm phần mềm thương mại đem bán Rất nhiều phần mềm thương mại sử dụng các framework mã nguồn mở để làm component trong phần mềm của mình. Ví dụ Windchill, một phần mềm collaborative software chuyên dụng cho các tập đoàn sản xuất lớn như Boeing, Airbus, Lockheed Martin, NASA, Ferrari, Toyota ... sử dụng rất nhiều phần mềm và framework mã nguồn mở như Apache, Tomcat, Log4J, itext, JFreeChart, Jasper, Xerces, Xalan... Giá của một Windchill deployment không bao giờ dưới tiền triệu.

- **Giải quyết khâu oai:** He..he..Lập trình viên cũng như các nghề khác như hàng thịt lợn hay gái bán hoa, đều có tự ái nghề nghiệp của mình. Khi giải quyết được một vấn đề công nghệ khó, làm ra một phần mềm hay, có nhiều người biết cũng thích. Đôi khi có một số lập trình viên viết một số cái nhỏ nhỏ để giải quyết khâu oai. 4. Lợi ích của mã nguồn mở đối với người sử dụng • Được dùng phần mềm miễn phí. • Có nhiều chọn lựa, không bị phụ thuộc vào một công ty nào. • Có một cộng đồng support lớn. • Có nhiều phần mềm đa dạng. Còn nhiều lợi ích khác, em các bác lười viết.

#### ▪ **Những điều cần quan tâm khi chuyển sang mã nguồn mở**

Máy tính không phải là cái TV, để có thể đem TV đen trắng vứt ra bãi rác rồi bê về cái TV màu. Khi chuyển từ một hệ điều hành này sang một hệ điều hành khác, từ một phần mềm này sang một phần mềm khác, người ta không thể vứt toẹt cái cũ đi rồi thay cái mới vào. Sau đây là những vấn đề cần quan tâm:

- **Những dữ liệu đã có:** Một công ty như JP Morgan không thể một sớm một chiều vứt bỏ toàn bộ dữ liệu khách hàng và tài chính được lưu trong mainframe hay AS 400 để nhảy qua làm mới trong J2EE. Phải có integration planning to move data.
- **Availability of Application:** Một công ty chuyên làm đồ họa, vẽ bản vẽ trong một graphic apps trên Mac, không thể một sớm một chiều vứt bỏ Mac OS để nhảy qua Linux, nếu trên Linux không có graphic software that understand



the existing graphic file format của các bản vẽ phò mà công ty đó đã vẽ trên máy Mac phò.

- **Existing application:** Một ngân hàng dân dụng đã đầu tư khối tiền của và chất đen của sự ngu dốt để phát triển phần mềm quản lí ngân hàng trên Window (như đa số các ngân hàng ở Việt nam đã làm, đang làm và sẽ làm) thì không thể một sớm một chiều nhảy qua Linux, vứt đi vài năm và vài triệu USD đã đầu tư, rồi viết lại phần mềm quản lí ngân hàng từ đầu.
- **Education cost :** Khi chuyển từ một phần mềm này qua một phần mềm khác, người sử dụng phải được đào tạo lại. Phải tính toán đến chi phí tài liệu, đào tạo, thời gian nhân viên nghỉ để học ...
- **Productivity of User:** Một user chuyển từ phần mềm này sang phần mềm khác, bao giờ cũng cần thời gian để làm quen, học hỏi, rút kinh nghiệm. Productivity sẽ giảm trong một thời gian nhất định. Tùy theo chức năng của phần mềm, việc giảm productivity sẽ có ảnh hưởng đến tài chính. Ví dụ một nhân viên ngân hàng chuyển từ một phần mềm này sang phần mềm khác, thì giảm productivity sẽ giảm lợi nhuận của ngân hàng. Còn một nhân viên lập trình Việt nam chuyển từ Window sang Linux hoặc từ Java qua Ruby, thì không có hại gì cho công ty, vì tất cả những gì nó viết trên Window hay Linux, bằng Java hay Ruby chỉ là "Hello, World"
- **Transition and deployment cost:** Khi chuyển phần mềm trong một công ty hay tổ chức lớn, thì phải tính đến transition and deployment code: • Thời gian system down để triển khai phần mềm mới. • Chi phí mua bán thiết bị để phục vụ cho phần mềm mới. • Chi phí nhân công để triển khai phần mềm.
- **Availability of support:** Các dự án phần mềm mở đa số là những dự án tình nguyện, trừ một số dự án có các công ty thương mại đứng đằng sau. Vì thế khi quyết định chuyển đổi phần mềm sang open source, công ty hay tổ chức phải cân nhắc: • Dự án phần mềm mở mình định dùng có phát triển ổn định hay không. • Phần mềm này có phổ biến rộng rãi trong cộng đồng hay không. • Có dễ tìm chuyên gia tư vấn, đào tạo và hỗ trợ kĩ thuật cho phần mềm này hay không.

### 3. Mã nguồn mở và tính pháp lý của phần mềm

Mã nguồn mở và tính pháp lý của phần mềm Cấu thành cơ bản của một giải pháp phần mềm là một loại giá trị phi vật chất, một sản phẩm phần mềm (software, program) mang nhiều nét đặc thù của một công trình nghiên cứu khoa học những đồng thời cũng mang trong mình nhiều dấu ấn của quá trình phát triển và đặc biệt là của các công dụng cụ đã được sử dụng. Đặc điểm nhị nguyên này gắn liền với đặc thù của quy trình phát triển phần mềm hiện đại: vừa là quá trình tìm tòi khoa học, vừa là một ngành công nghiệp với thiết kế mẫu, quy trình chuẩn và các linh kiện dụng cụ có thể sử dụng và khả năng tự động hóa cao

Tính khoa học thể hiện rõ nhất ở mối quan hệ liên hoàn giữa tất cả các khâu trong quá trình tìm tòi, phát triển giải pháp: từ việc xác định nội dung bản chất vấn đề (requirement analysis) đến việc xây dựng các mô hình làm việc (modeling) - xác định kiến trúc tổng thể cho sản phẩm (architecture) - thiết kế (design) - tìm giải pháp cụ thể (lập trình – implementation) đến thử nghiệm (test) và khắc phục lỗi (debug). Kết quả của một bước trước ảnh hưởng quyết định đến các bước sau và với trình độ phát triển ngày nay vẫn chưa có cách nào để từ một yêu cầu cho trước không tốn nhiều công suy nghĩ có ngay được kết quả cuối cùng - chương trình máy tính. Rất nhiều khi do kết quả của công đoạn trước sai hoặc không tốt mà tất cả nỗ lực ở công đoạn sau đều trở thành vô nghĩa: chương trình không chạy được hoặc tuy có chạy nhưng sai (có lỗi) và không thể đáp ứng yêu cầu đã đề ra. Điểm duy nhất ngày nay có thể tự động hóa được hoàn toàn là việc dịch chương trình từ chỉ dẫn chi tiết (mã nguồn) ra ngôn ngữ máy – tập hợp các bits và byte mà chúng ta nhìn thấy dưới dạng file EXE, DLL v.v. trong máy tính. Nếu định nghĩa tính sáng tạo là những gì nhất thiết đòi hỏi phải có tư duy của con người thì mọi quá trình có thể tự động hóa đều không thể coi là có tính sáng tạo. Như vậy tuy giá trị sử dụng của một chương trình luôn được đo bởi kết quả sử dụng cụ thể của chương trình đấy, nhưng giá trị lý luận, tính sáng tạo luôn được đo qua kiến trúc, thiết kế và các giải pháp trước công đoạn cuối cùng – công đoạn dịch ra ngôn ngữ máy. Một công đoạn khác cũng có thể thực hiện bởi máy móc đó là việc dịch các lời thoại, menu, chỉ dẫn trong chương trình ra các ngôn ngữ khác nhau. Chính vì lý do này mà mọi người đều cho rằng việc “Việt hóa” một phần mềm để đem đi thi sáng tạo là việc làm không thể chấp nhận được (mặc dù đôi khi “Việt hóa” cũng đòi hỏi phải có những giải pháp nhất định). Bên cạnh đòi hỏi cần phải có tư duy, sáng tạo thực sự còn hàm ý phát minh, tìm tòi ra giải pháp mới, cách làm mới. Như vậy để xứng đáng là giải pháp trí tuệ, bên cạnh việc phải có nỗ lực đáng kể trong quá trình tư duy thiết kế xây dựng giải pháp, những kết quả đạt được - cụ thể là kiến trúc, thiết kế, giải thuật, cấu trúc dữ liệu của chương trình - cũng phải có tính mới, tính hữu dụng. Tùy

vào giải pháp mới tìm được nhiều hay ít, tính quyết định của nó đến giải pháp tổng thể cao hay thấp cũng như tính hữu ích và độc đáo của nó mà chúng ta nói giải pháp có tính sáng tạo cao hay không. Trong ngành công nghệ phần mềm, mọi người thường thiên về ý kiến cho rằng các công đoạn trước như phân tích, thiết kế, tìm giải thuật, cấu trúc dữ liệu đòi hỏi tư duy và kinh nghiệm nhiều hơn các công đoạn sau như việc thể hiện chúng trong một ngôn ngữ cụ thể (lập trình) hay thử và cải tiến, khắc phục lỗi[1]. Bên cạnh là một hoạt động mang tính nghiên cứu, sáng tạo, sản xuất phần mềm còn là một ngành công nghiệp và vì vậy người ta tìm mọi cách để rút ngắn quá trình sản xuất và nâng cao năng suất lao động. Một trong những cách được áp dụng rộng rãi nhất là sử dụng các khuôn mẫu có sẵn (framework), thư viện lập trình và linh kiện lắp ghép (component). Các framework thường chỉ bao gồm các cấu trúc cơ bản như kiến trúc hệ thống, các mẫu thiết kế tối ưu hay các giải pháp mang tính lý thuyết còn thư viện thường bao gồm 1 tập hợp các giải pháp cụ thể (giải thuật, cấu trúc dữ liệu, mã nguồn) cho một vấn đề cụ thể (ví dụ tìm giá trị cực tiểu của một hàm bất kỳ). Các component tuy cũng là những giải pháp cụ thể nhưng cho phép người thiết kế tùy biến chức năng qua việc xác định các tham số hoạt động (properties) và ghép nối chúng lại với nhau (lập trình) thành một giải pháp hoàn chỉnh. Do khả năng tùy biến cao nên một component có thể “chỉ” là một giải pháp hiển thị kết quả dưới dạng đồ thị nhưng cũng có thể bao gồm toàn bộ chức năng phức tạp của một chương trình soạn thảo văn bản như Word®. Như vậy với linh kiện và công cụ thích hợp người ta có thể nhanh chóng bổ xung tính năng cho 1 phần mềm mà nếu tự phát triển sẽ phải mất hàng tháng trời nếu không phải là hàng năm! Trong ngành công nghiệp phần mềm ngày nay, sử dụng linh kiện là giải pháp rất hay được áp dụng, nhất là trong phương pháp phát triển nhanh (RAD – rapid application development) như Delphi, Visual Basic hay .NET. Vì vậy việc xác định tính sáng tạo của một chương trình được xây dựng bởi các linh kiện đòi hỏi phải xác định xem bản thân các linh kiện đó đã cho phép giải quyết phần lớn bài toán được đặt ra hay chưa. Mã nguồn mở, GPL và tính pháp lý của phần mềm Xuyên suốt các cuộc tranh luận về iCMS và nguồn gốc của nó là các khái niệm như “mã nguồn mở”, “luật GPL” hay là “Copyright”; nhiều khi lẫn lộn, nhiều khi sai lầm và đối với các độc giả ngoại đạo thì chắc chắn là rất khó hiểu. Vậy thực chất chúng là gì và phải hiểu chúng thế nào cho đúng? Mã nguồn mở (open-source software (<http://en.wikipedia.org/wiki/Open-source>)) hiểu theo nghĩa rộng là một khái niệm chung được sử dụng cho tất cả các phần mềm mà mã nguồn của nó được công bố rộng rãi công khai và cho phép mọi người tiếp tục phát triển phần mềm đó. Điều này không có nghĩa là chúng có thể được sao chép, sửa chữa thoải mái hay sử dụng vào mục đích nào cũng được. Mã nguồn mở được công bố dưới rất nhiều điều kiện khác nhau (license), một số trong đó cho phép phát triển, sử dụng và bán tùy ý miễn là giữ nguyên các dòng về nguồn gốc sản phẩm (BSD), một số bắt buộc tất cả các sản phẩm làm ra từ đó cũng phải là

open-source (GPL), một số khác đòi hỏi phải công bố trọn vẹn mã nguồn (Mozilla), một số khác không cho phép sử dụng vào mục đích thương mại (Sun Solaris Source Code License), một số khác lại không có ràng buộc gì đáng kể (public domain, MIT X11 license) v.v. Qua đó ta thấy khái niệm open source không thể chuẩn xác mà muốn nói đến tính pháp lý của việc sử dụng các phần mềm mã nguồn mở, chúng ta phải xem xét đến điều kiện sử dụng (license) cụ thể mà dưới đó chúng được công bố. Một điều kiện hay được áp dụng nhất là GPL: GNU General Public License (<http://www.fsf.org/licenses/gpl.html>) của tổ chức Free Software Foundation. GPL license có 2 đặc điểm phân biệt, đó là:

- Tác giả gốc giữ bản quyền về phần mềm nhưng cho phép người dùng rất nhiều quyền khác, trong đó có quyền tìm hiểu, phát triển, công bố cũng như quyền khai thác thương mại sản phẩm.
- Tác giả sử dụng luật bản quyền để bảo đảm các quyền đó không bao giờ bị vi phạm đối với tất cả mọi người, trên mọi phần mềm có sử dụng mã nguồn của mình.

Đặc biệt điểm thứ 2 thường được gọi là hiệu ứng virus (viral effect) vì nó biến tất cả các phần mềm có dùng mã nguồn GPL cũng biến thành phần mềm GPL. Trên thực tế điều này có ý nghĩa: bất kỳ tác giả nào sử dụng dù chỉ 1 phần rất nhỏ mã nguồn GPL trong chương trình của mình cũng phải công bố chương trình đó dưới điều kiện GPL. Điều kiện này quy định ví dụ:

- Mọi phần mềm GPL đều phải công bố mã nguồn của mình rộng rãi công khai và phải tạo điều kiện cho mọi người truy cập được mã nguồn ấy (ví dụ qua web hoặc qua việc bán CD giá rẻ)
- Giữ nguyên mọi dòng chú thích về nguồn gốc tác giả, bản quyền của họ cũng như điều kiện được áp dụng đối với phần mềm (trong 1 file có tên LICENSE)
- Cấm việc bán mã nguồn nhưng cho phép kinh doanh chương trình được tạo ra từ mã nguồn ấy hoặc là các dịch vụ hỗ trợ liên quan.

#### **4.Xã hội**

##### **❖ Tổ chức cộng đồng :**

###### **➤ Người sử dụng**

Những người sử dụng là các thành viên cộng đồng có một nhu cầu đối với dự án. Họ là những thành viên quan trọng nhất của cộng đồng và không có họ thì dự án có thể không có mục đích. Bất kỳ ai cũng có thể là một người sử dụng; không có những yêu cầu đặc biệt nào.

Dự án yêu cầu những người sử dụng của mình tham gia trong dự án và cộng đồng càng nhiều có thể càng tốt. Những đóng góp của những người sử dụng xúc tác cho đội dự án đảm bảo rằng họ đang làm thỏa mãn các nhu cầu của những người sử dụng đó.

Những đóng góp chung của những người sử dụng bao gồm (như không bị hạn chế):

việc truyền bá về dự án (như, một đường liên kết trên một website và việc nâng cao nhận thức bằng truyền khẩu).

thông tin cho các lập trình viên về những điểm mạnh và yếu từ quan điểm của người sử dụng.

đưa ra sự hỗ trợ tinh thần (một câu 'cảm ơn' đi cùng)

đưa ra sự hỗ trợ tài chính (phần mềm là nguồn mở, nhưng các lập trình viên cần phải ăn)









Những người sử dụng mà tiếp tục tham gia với dự án và cộng đồng của nó thường sẽ trở nên có liên quan ngày một nhiều hơn nữa. Những người sử dụng như vậy tự thấy họ đang trở thành những người đóng góp, như được mô tả trong phần sau.

Những người đóng góp

Những người đóng góp là các thành viên của cộng đồng mà đóng góp theo những cách thức cụ thể cho dự án. Bất kỳ ai cũng có thể trở thành một người đóng góp, và những đóng góp có thể nằm ở nhiều dạng, như được mô tả trong một tài liệu riêng rẽ (bản dịch tiếng Việt).

Không có mong đợi về cam kết cho dự án, không có những yêu cầu kỹ năng đặc biệt và không có qui trình lựa chọn.

Bổ sung thêm vào những hành động của họ như những người sử dụng, những người đóng góp cũng có thể thấy họ đang làm một hay nhiều hơn những điều sau đây:

-  hỗ trợ cho những người sử dụng mới (những người sử dụng đang tồn tại thường là những người tốt nhất để hỗ trợ cho những người sử dụng mới)
-  báo cáo các lỗi
-  xác định các yêu cầu
-  cung cấp các hình đồ họa và thiết kế web
-  lập trình
-  viết tài liệu
-  sửa các lỗi
-  bổ sung thêm các tính năng

Những người đóng góp tham gia với dự án thông qua trình theo dõi vấn đề và danh sách thư, hoặc bằng việc viết hoặc soạn thảo tài liệu. Họ đề trình những thay đổi cho bản thân

dự án thông qua các bản vá (bản dịch tiếng Việt), chúng sẽ được những người đệ trình (xem phần sau) cân nhắc để đưa vào trong dự án. Danh sách thư của các lập trình viên là nơi thích hợp nhất để yêu cầu sự trợ giúp khi tiến hành sự đóng góp đầu tiên đó.

Như những người đóng góp giành được kinh nghiệm và sự quen thuộc với dự án, hồ sơ của họ trong, và cam kết tới, cộng đồng sẽ gia tăng. Ở giai đoạn nào đó, họ có thể thấy chính họ đang trở thành những ứng viên cho chế độ của người đề xuất.

### ➤ Những người đề xuất

Những người đề xuất là các thành viên của cộng đồng mà đã chỉ ra được rằng họ tận tâm cho sự phát triển liên tục của dự án thông qua sự cam kết liên tục với cộng đồng. Chế độ của người đề xuất cho phép những người đóng góp dễ dàng hơn triển khai các hoạt động có liên quan tới dự án của họ bằng việc trao cho họ sự truy cập trực tiếp tới các tài nguyên của dự án. Đó là, họ có thể thực hiện những thay đổi trực tiếp tới các kết quả đầu ra của dự án, mà không phải đệ trình những thay đổi đó thông qua các bản vá.

Điều này không có nghĩa là một người đệ trình là tự do làm những gì họ muốn. Trên thực tế, những người đề xuất không có quyền nhiều hơn so với những người đóng góp đối với dự án. Trong khi chế độ của những người đề xuất chỉ định một thành viên được quý trọng của cộng đồng, người đã thể hiện sự tôn trọng lành mạnh đối với các mục tiêu và mục đích của dự án, thì công việc của họ vẫn tiếp tục sẽ được cộng đồng rà soát lại trước khi chấp nhận trong một phát hành chính thức. Sự khác biệt chính giữa một người đệ trình và một người đóng góp là khi sự phê chuẩn này được thấy từ cộng đồng. Một người đệ trình tìm kiếm sự phê chuẩn sau sự đóng góp được làm, hơn là trước đó.

Việc tìm kiếm sự phê chuẩn sau khi thực hiện một đóng góp được biết tới như là một qui trình đệ trình sau đó rà soát lại. Nó có hiệu lực hơn để cho phép những người được tin cậy thực hiện những đóng góp trực tiếp, khi mà đa số những đóng góp đó sẽ được dự án chấp nhận. Dự án sử dụng các cơ chế giao tiếp khác nhau (bản dịch tiếng Việt) để đảm bảo rằng tất cả những đóng góp được cộng đồng như một tổng thể rà soát lại. Với thời gian, một người đóng góp được mời để trở thành một người đệ trình, họ sẽ trở nên quen thuộc hơn với các công cụ khác nhau của dự án như một người sử dụng và sau đó như một người đóng góp.

Bất kỳ ai cũng có thể trở thành một người đệ trình; không có những yêu cầu đặc biệt, khác với phải chỉ ra được một thiện ý và khả năng để tham gia trong dự án như một tay chơi của đội. Điển hình, một người đề xuất tiềm tàng cần phải thể hiện rằng họ có một sự hiểu biết về dự án, các mục đích và chiến lược của nó. Họ cũng sẽ đã đưa ra được những đóng góp có giá trị cho dự án qua một giai đoạn thời gian.

Những người đề xuất mới có thể được bất kỳ người đề xuất đang tồn tại nào đề cử. Một khi họ đã được đề cử, sẽ có một phiếu bầu của ban quản lý dự án – PMC (Project Management Committee) (xem bên dưới). Việc biểu quyết của những người đề xuất là một trong những hoạt động hiểm hoi diễn ra trong danh sách quản lý riêng tư của dự án. Điều này là để cho phép các thành viên của PMC tự do bày tỏ những ý kiến của họ về một ứng viên mà không gây ra sự bối rối lúng túng nào. Một khi cuộc biểu quyết được tổ chức, thì các kết quả biểu quyết được tổng hợp sẽ được xuất bản trong danh sách thư công khai. Ứng viên có quyền yêu cầu một sự giải thích đối với bất kỳ phủ quyết 'không' nào chống lại họ, bất kể kết quả của biểu quyết là thế nào. Sự giải thích này sẽ được Chủ tịch của PMC đưa ra (xem bên dưới) và sẽ là nặc danh và có tính xây dựng một cách tự nhiên. Các ứng viên có thể từ chối sự chỉ định của họ như một người đề xuất. Tuy nhiên, điều này là không thường thấy, khi mà dự án không mong đợi bất kỳ thời điểm đặc thù hay cam kết về tài nguyên nào từ các thành viên cộng đồng của nó. Ý định đằng sau vai trò của người đề trình là để cho phép mọi người đóng góp cho dự án dễ dàng hơn, không trói họ vào dự án theo bất kỳ cách chính thức nào.

Điều quan trọng để nhận thức rằng chế độ của người đề xuất là một sự ưu tiên, không phải một quyền. Sự ưu tiên đó phải giành được và một khi giành được thì nó có thể bị PMC loại bỏ được (xem phần dưới) trong những hoàn cảnh đặc biệt. Tuy nhiên, theo những hoàn cảnh thông thường thì chế độ của người đề xuất tồn tại lâu cho tới khi người đề xuất mong muốn tiếp tục tham gia với dự án.

Một người đề xuất mà thể hiện một mức đóng góp trên trung bình cho dự án, đặc biệt với sự tôn trọng đối với đường hướng chiến lược và sự lành mạnh dài hạn của nó, có thể được đề cử để trở thành một thành viên của PMC. Vai trò này được mô tả bên dưới.

### ➤ **Ban Quản lý Dự án**

Ban quản lý dự án bao gồm những cá nhân được xác định như là 'những người chủ dự án' trên site phát triển. PMC có các trách nhiệm bổ sung đối với và trên cả những trách nhiệm của một người đề xuất. Các trách nhiệm đó đảm bảo sự điều hành trơn tru của dự án. Các thành viên PMC được kỳ vọng sẽ rà soát lại những đóng góp mã nguồn, tham gia trong việc lên kế hoạch chiến lược, phê chuẩn những thay đổi cho mô hình điều hành và quản lý các bản quyền trong các kết quả của dự án.

Các thành viên của PMC không có quyền đáng kể đối với các thành viên khác của cộng đồng, dù đây là PMC mà biểu quyết về những người đề xuất mới. Nó cũng ra các quyết định khi sự đồng thuận của cộng đồng không thể đạt được. Bổ sung thêm, PMC có sự truy cập tới danh sách thư riêng và của dự án và kho lưu trữ của nó. Danh sách này được

sử dụng cho các vấn đề nhạy cảm, như các cuộc biểu quyết đối với những người đề xuất mới và các vấn đề pháp lý mà không thể được thảo luận công khai. Nó không bao giờ được sử dụng để quản lý hoặc lên kế hoạch dự án.

Membership of the PMC is by invitation from the existing PMC members. A nomination will result in discussion and then a vote by the existing PMC members. PMC membership votes are subject to consensus approval of the current PMC members.

Cơ chế thành viên của PMC là được mời từ các thành viên PMC đang tồn tại. Một ứng viên sẽ gây ra sự thảo luận và sau đó cuộc biểu quyết từ các thành viên PMC đang tồn tại. Các cuộc biểu quyết cơ chế thành viên của PMC tuân theo sự phê chuẩn đồng thuận của các thành viên PMC hiện hành.

### ➤ **Chủ tịch PMC**

Chủ tịch PMC là một cá nhân duy nhất, được các thành viên PMC bầu. Một khi ai đó được chỉ định thành Chủ tịch, họ vẫn giữ trong vai trò đó cho tới khi họ chọn về hưu, hoặc PMC biểu quyết với đa số hơn 2/3 để loại bỏ họ.

Chủ tịch PMC không có quyền bổ sung nào đối với các thành viên khác của PMC: vai trò đó là vai trò của nhà điều phối và người tạo thuận lợi. Chủ tịch cũng được kỳ vọng đảm bảo rằng tất cả các quy trình điều hành sẽ được gắn kết mạch lạc, và có quyền biểu quyết khi dự án không đạt được sự đồng thuận.

### Sự hỗ trợ

Tất cả các thành viên trong cộng đồng được khuyến khích cung cấp sự hỗ trợ cho những người sử dụng mới trong cơ sở hạ tầng quản lý dự án. Sự hỗ trợ này được cung cấp như một cách thức phát triển cộng đồng. Những người tìm cách hỗ trợ nên nhận thức được rằng tất cả hoạt động hỗ trợ trong dự án là tự nguyện và vì thế được cung cấp như và khi thời gian cho phép. Một người sử dụng có yêu cầu về số lần hoặc các kết quả trả lời được đảm bảo vì thế nên tìm cách mua một hợp đồng hỗ trợ từ một thành viên cộng đồng. Tuy nhiên, đối với những người có thiện ý tham gia với dự án trong các điều khoản riêng của mình, và có thiện ý để giúp hỗ trợ cho những người sử dụng khác, thì các kênh hỗ trợ cộng đồng là lý tưởng.

### Quy trình đóng góp

Bất kỳ ai cũng có thể đóng góp cho dự án, bất kể các kỹ năng của họ, khi mà có nhiều cách thức để đóng góp. Ví dụ, một người đóng góp có thể là tích cực trong danh sách thư của dự án và trình theo dõi vấn đề, hoặc có thể cung cấp các bản vá (bản dịch tiếng Việt). Các cách thức đóng góp khác nhau được mô tả chi tiết hơn trong một tài liệu riêng rẽ (bản dịch tiếng Việt).



Danh sách thư của các lập trình viên của dự án được thực hiện thông qua thảo luận với tất cả các thành viên của cộng đồng, từ người sử dụng mới nhất cho tới thành viên có kinh nghiệm nhất của PMC. Tất cả các thảo luận quản lý dự án không nhảy cảm diễn ra trong danh sách thư của những người đóng góp cho dự án. Thỉnh thoảng, thảo luận nhảy cảm diễn ra trong một danh sách thư riêng.

Để đảm bảo rằng dự án không bị sa lầy vì những thảo luận bất tận và biểu quyết liên miên, dự án vận hành một chính sách đồng thuận lười. Điều này cho phép đa số các quyết định được thực hiện mà không dùng tới phương sách của một biểu quyết chính thống.

❖ ***Thành phần xã hội trong cộng đồng nguồn mở và tỉ lệ nhân lực và đóng góp***

✚ Tỉ lệ nhân lực:

- Lập trình viên chuyên phát triển phần mềm nguồn mở: 10%
- Tình nguyện viên tự do :70%
- Học sinh, sinh viên học IT:10%
- Nhân viên công ty phần mềm: 5%
- Nghiên cứu viên: 5%

✚ Tỉ lệ đóng góp :

- Lập trình viên chuyên phát triển phần mềm nguồn mở: 20%
- Tình nguyện viên tự do :50%
- Học sinh, sinh viên học IT:10%
- Nhân viên công ty phần mềm: 5%
- Nghiên cứu viên: 15%

❖ ***Một số định hướng vĩ mô, kế hoạch cho sự phát triển phần mềm nguồn mở***

Từ kinh nghiệm của một số quốc gia trong khu vực, chúng ta có thể tìm hiểu học tập một số mô hình sau về đào tạo nhân lực, tăng cường tiềm lực phát triển phần mềm nguồn mở:

- Xây dựng các phòng máy tính chỉ trang bị phần mềm nguồn mở trong các cơ sở đào tạo, phổ cập tin học. Tiến tới đầu tư xây dựng những trung tâm tài năng về nguồn mở cấp thành phố, cấp quốc gia.
- Thay đổi, điều chỉnh chương trình đào tạo về sử dụng máy tính nói riêng và công nghệ thông tin nói chung không dựa trên một môi trường cứng nhắc (hiện nay phần lớn chương trình phổ cập tin học, kể cả chương trình quốc gia đều được xây dựng dựa trên hệ điều hành và sản phẩm của Microsoft)

- Mở rộng các hoạt động tuyên truyền xã hội, tổ chức các cuộc thi sản phẩm phần mềm nguồn mở. (cuộc thi Trí tuệ Việt nam năm 2004 đã có định hướng ưu tiên các sản phẩm nguồn mở hay sử dụng các thư viện nguồn mở)
- Mở rộng các chương trình đào tạo và cấp chứng chỉ, kể cả chứng chỉ quốc tế theo định hướng nguồn mở. Khuyến khích các công ty đầu tư, xây dựng chương trình và triển khai các hoạt động đào tạo này.
- Khuyến khích các dự án nghiên cứu phát triển phần mềm nguồn mở, ứng dụng nguồn mở trong các lĩnh vực nghiên cứu khác nhau như Sinh học, Vật lý, Hoá học, Môi trường, Y- dược, v.v... Các dự án nghiên cứu nhận kinh phí nhà nước khi mua sắm phần mềm ứng dụng cần ưu tiên lựa chọn phần mềm nguồn mở.
- Phần mềm nguồn mở không phải cây gậy thần giải quyết mọi vấn đề của quốc gia trong phát triển CNTT. Các lợi ích mà phần mềm nguồn mở mang lại tuy rõ nét, nhưng đòi hỏi phải có quyết tâm và sự phối hợp đồng bộ trong triển khai.
- Đó là thách thức lớn nhất và cũng là cơ hội lớn nhất cho chúng ta, một cơ hội đòi hỏi phải nỗ lực chạy đua với thời gian và tập trung tiềm lực của cả cộng đồng mới có thể trở thành hiện thực.

#### ❖ **Tổ chức cơ sở hạ tầng :**

##### ➤ Tài liệu :

- Rất nhiều.
- Phong phú, đa dạng.
- Diễn đàn : • Mạng xã hội coding GITHUB.COM. • Diễn đàn Ubuntu việt nam (forum.ubuntu-vn.org).
- Chat :

#### ❖ **Quản lý lỗi, quản lý mã nguồn :**

- CSV (Concurrent Versions System) hệ thống các phiên bản đồng thời : đây là hệ thống quản lý các phiên bản phần mềm mã nguồn mở được đề xuất và phát triển bởi Dick Grune vào thập niên 1980.
- SVN (Subversion) : là một phần mềm mã nguồn mở dùng để quản lý và kiểm tra các phiên bản mã nguồn khác nhau trong quá trình phát triển phần mềm.

- GIT quản lý mã nguồn phân tán được thiết kế bởi Linus Torvalds vào 2005. Trong khi CVS và SVN lấy project làm gốc, từ đó rẽ ra các nhánh là developer, thì Git lấy developer làm gốc, từ đó rẽ ra các nhánh là project. Hai nhánh project gặp nhau có nghĩa hai developer cùng hợp tác để làm cùng project.

#### ❖ **Các giai đoạn phát triển phần mềm**

- Giai đoạn 1: Thu thập thông tin khách hàng
  - Tìm hiểu yêu cầu thực tế của khách hàng
  - Xác nhận các yêu cầu của khách hàng về phần mềm và các thông tin khách hàng cung cấp.
- Giai đoạn 2: Giai đoạn phân tích
  - Dựa vào những thông tin và yêu cầu thiết kế phần mềm của khách hàng, phân tích rõ ràng mục đích, yêu cầu, nguyện vọng của khách hàng
  - Lập kế hoạch thực hiện dự án
  - Căn cứ trên kết quả khảo sát, thông tin do khách hàng cung cấp thiết kế lập cấu trúc cho phần mềm và thiết kế chung cho toàn hệ thống.
- Giai đoạn 3: Giai đoạn thiết kế
  - Xem xét chi tiết yêu cầu và tiến hành thiết kế
  - Tiến hành xử lý , thu thập tài liệu liên quan
  - Kiểm tra sản phẩm để đảm bảo thống nhất.
- Giai đoạn 4: Giai đoạn xây dựng
  - Xây dựng cơ sở dữ liệu dựa trên thiết kế.
  - Tích hợp hệ thống: lắp ghép phân tích thiết kế, nội dung lập trình thành một sản phẩm.
  - Tiến hành kiểm tra, chỉnh sửa và thực hiện nghiệm thu nội bộ
- Giai đoạn 5: Giai đoạn chạy thử

- Tổng hợp nội dung, xây dựng hệ thống theo thiết kế.
  - Kiểm tra và sửa lỗi.
  - Lắp ghép thiết kế với phần mềm, đảm bảo không sai lệch với thiết kế và phần mềm hoạt động tốt.
  - Chạy thử hệ thống trong vòng 01 tuần.
  - Kiểm tra và sửa.
- Giai đoạn 6: Giai đoạn nghiệm thu
- Khách hàng duyệt dự án: khách hàng duyệt chất lượng dự án.
  - Đăng tải hệ thống: đảm bảo hệ thống chạy tốt.
- Giai đoạn 7: Giai đoạn chuyển giao
- Bàn giao cho khách hàng (có đĩa CD mã nguồn)
  - Đào tạo và huấn luyện khách hàng dùng chương trình một cách hiệu quả
  - Đánh giá, bảo trì.

### **III. Tìm hiểu cài đặt và phát triển plugin cho Wordpress**

#### ***1. Plugin là gì***

Plugin là một phần mềm, hay nói cách khác nó là một hay nhiều tập tin PHP có chứa nhiều dòng lệnh hoặc hàm (function) do người dùng viết ra, trong đó ngoài các hàm do người dùng tạo ra thì còn các hàm kế thừa từ **thư viện có sẵn của WordPress**.

Đối với WordPress, nếu ta cần một plugin nào đó thì ta có thể tìm kiếm trước trên mạng thông qua Google hoặc tìm kiếm trực tiếp trên **kho plugin của WordPress**.

Khi đến với kho plugin này, ta sẽ dễ dàng tìm kiếm cho mình một plugin vừa ý bởi hệ thống có hỗ trợ đánh giá xếp hạng theo sự phổ biến của plugin hoặc sắp xếp plugin theo từng chức năng.

## 2. Tại sao lại phải dùng plugin

Nếu như ta xem WordPress là shop bán đồ chơi thì một trang blog WordPress mà không **cài đặt plugin** thì coi như shop đồ chơi đó chỉ có cái vỏ chứ không có đồ chơi được chứa đựng bên trong.

Có rất nhiều plugin với mục đích sử dụng khác nhau, nó cung cấp cho người dùng một số chức năng và tiện ích nhất định. Ta có thể tìm kiếm plugin để phục vụ một vài sở thích nào đó của ta hoặc plugin đó sẽ làm cho việc quản lý blog của ta tốt hơn.

Ta không thể sử dụng WordPress mà không có plugin, ví dụ như plugin **WordPress SEO** sẽ giúp cho ta dễ dàng thực hiện công việc **SEO** cho blog WordPress hơn.

## 3. Ta cần phải có những kiến thức gì trước khi bắt đầu?

- Kiến thức về HTML và CSS cơ bản
- Kiến thức về PHP cơ bản
- Ngoài 2 loại kiến thức bên trên thì ta có thể bổ sung thêm kiến thức về MySQL, Javascript,... nếu cần.
- Kiến thức về Action Hook và Filter Hook.



Kiến thức cần có khi viết plugin

## 4. Công cụ cần thiết

- Phần mềm hỗ trợ lập trình PHP (Đề nghị dùng Notepad++ hoặc Sublime Text 2).
- Máy tính của ta đã cài đặt giả lập localhost (Đề nghị dùng XAMP hoặc AppServ).
- Ta đã cài đặt sẵn một blog WordPress trên localhost.







## 5. Cấu trúc thư mục và tập tin

Tạo thư mục và tập tin là hai bước ta phải làm đầu tiên trong quá trình viết plugin. Thư mục của plugin phải được đặt trong thư mục mà WordPress dành riêng cho các plugin hoạt động, bắt buộc

ta phải bỏ plugin vào thư mục này thì mới có thể kích hoạt và sử dụng, đường dẫn đó ta có thể xem bên dưới.

wp-content/plugins

Như vậy, thư mục plugin của ta phải được đặt bên trong thư mục **plugins** với đường dẫn như bên trên, sau đó các thư mục con và các tập tin sẽ được đặt nằm trong thư mục gốc plugin của ta.

Name	Date modified	Type	Size
 languages	27/04/2014 10:40 SA	File folder	
 plugins	04/06/2014 9:13 CH	File folder	
 themes	04/06/2014 9:27 CH	File folder	
 upgrade	04/06/2014 9:27 CH	File folder	
 uploads	26/04/2014 8:41 CH	File folder	
 index.php	08/01/2012 5:01 CH	PHP File	1 KB

## 6. Tên của plugin

Phần tên của plugin phải được ta suy nghĩ và đặt ra trước, tên của plugin phải thể hiện rõ công dụng của plugin mà ta có ý định tạo. Tên của plugin phải rõ ràng và không được trùng tên với các plugin đã có sẵn, trước tiên ta nên tìm kiếm trên Google xem tên ta muốn đặt có hợp lý hay không. Có thể sử dụng nhiều từ để diễn tả cho tên của plugin nhằm tránh sự trùng lặp khi đặt tên.

## 7. Tên thư mục và tên của các tập tin

Tên của thư mục cũng là tên của plugin mà ta viết thường không có dấu Tiếng Việt và các từ được nối với nhau bằng dấu gạch ngang. Tên của thư mục sẽ đại diện cho plugin của ta, sau này nếu như ta truy vấn tới các tập tin hình ảnh, css hay javascript trong thư mục của plugin thì ta sẽ phải sử dụng đến tên này, WordPress gọi tên của thư mục chứa plugin là **text-domain**.

Đối với tên của tập tin PHP cũng vậy, ta không được đặt trùng tên với các plugin khác. Thông thường ta cũng sẽ lấy tên của thư mục chứa plugin đặt tên cho tập tin PHP chính của plugin. Một plugin phải có ít nhất một tập tin PHP, ngoài ra thì ta có thể tạo thêm bao nhiêu tập tin PHP phụ cũng được.

## 8. Bắt đầu viết code cho plugin

Đối với tập tin PHP chính của plugin, ta phải đặt một vài ghi chú ở phía trên cùng của tập tin, những ghi chú này bao gồm tên của plugin, tên của tác giả, mô tả cho plugin, phiên bản hiện tại của plugin và đường dẫn trang chủ của plugin với trang chủ của tác giả. Cụ thể hơn, những phần cơ bản này ta sẽ phải tạo giống như bên dưới, thay thông tin lại cho phù hợp với plugin của ta.

```
<?php
/**
 * Plugin Name: 5TN Slider// Tên của plugin
 * Plugin URI: // Địa chỉ trang chủ của plugin
```

```

* Description: Slider Component for WordPress // Phần mô tả cho plugin
* Version: 1.0 // Đây là phiên bản đầu tiên của plugin
* Author: 5T&N // Tên tác giả, người thực hiện plugin này
* Author URI: // Địa chỉ trang chủ của tác giả
* License: GPLv2 or later // Thông tin license của plugin, nếu không quan tâm thì ta cứ để
GPLv2 vào đây
*/
?>

```

Vậy là bây giờ ta đã có thể vào bảng điều khiển và kích hoạt plugin.

## **9. Viết code cho plugin**

Hook, Action và Filter

- Tìm hiểu thông tin Plugin API để lập trình plugin.

### ***Hook là gì***

Là khái niệm do WordPress đưa ra để cho phép ta thực thi một hàm nào đó vào một thời gian xác định. Có 2 loại Hook chính đó là: Action và Filter.

### ***Action là gì***

Action là một loại Hook, và nó là một hàm PHP nào đấy. Action được đặt ở một vị trí nhất định cho phép ta gọi một lệnh nào đấy tại vị trí ta đặt Action.

Ví dụ như ta muốn thêm một đoạn mã javascript nào đấy vào phần footer trước khi kết thúc thẻ `</body>` thì ta có thể sử dụng Action `wp_footer`.

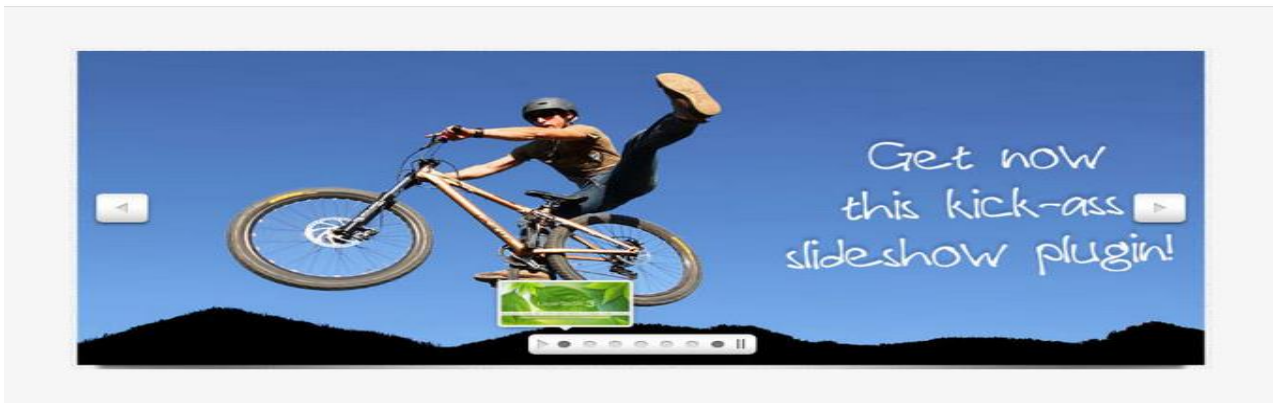
### ***Filter là gì***

Filter cũng là một loại Hook, nó được đặt nằm giữa cơ sở dữ liệu và trình duyệt web, khi ta thực thi một câu lệnh nào đó thì đều phải thông qua ít nhất một Filter, và dĩ nhiên đây cũng là một hàm PHP. Filter đóng vai trò là người quản lý, nó sẽ lọc những thông tin người dùng muốn đưa vào cơ sở dữ liệu và ngược lại.

Các tập tin ngoại

Đối với các tập tin .css và .js thì ta phải tải về và lưu vào các thư mục con bên trong thư mục của plugin, đây là quy định bắt buộc của WordPress nếu như ta muốn plugin của mình được WordPress chấp nhận và đăng tải trên trang chủ của họ.

Sản phẩm:



*Front-end:*

Back- end:

