

OPEN SOURCE

HIGH LEVEL TRAINING MATERIAL OF FOSS BRIDGE EU-VIETNAM



TÀI LIỆU ĐÀO TẠO
VỀ PHẦN MỀM TỰ DO & NGUỒN MỞ (FOSS/FLOSS)
CỦA DỰ ÁN FOSS BRIDGE EU-VIETNAM

Project funded by



Project implemented by



Networking partners



Part of the project



THÔNG TIN THÊM VỀ DỰ ÁN

Website: www.foss-bridge.org

Địa chỉ e-mail liên hệ: contact@foss-bridge.org

ĐIỀU HÀNH DỰ ÁN, INWENT CHLB ĐỨC

Balthas Seibold

Senior Project Manager - ICT for development,
Division 4.04 Business Development and Infrastructure

InWEnt - Internationale Weiterbildung und Entwicklung gGmbH
Capacity Building International, Germany

Friedrich-Ebert-Allee 40, 53113 Bonn, Germany
Phone: +49-228-4460-1253, Fax: +49-228-4460-2253
E-mail: balthas.seibold@inwent.org
<http://www.inwent.org> and <http://www.it-inwent.org>



QUẢN TRỊ DỰ ÁN TẠI VIỆT NAM

Project manager

Vũ Thế Bình

18 Hoàng Quốc Việt, Cầu Giấy, Hà Nội
Tel: +84-4-3756 1535
Fax: +84-4-3756 1888
E-mail: binh@netnam.vn
<http://www.ioit.ac.vn>

Mục lục/Content

❑ 5 tuyên bố về FLOSS	1
❑ Giấy phép và độc quyền sở hữu trí tuệ	9
❑ Văn hóa, nền tảng và lịch sử FLOSS.....	19
❑ Các giấy phép tự do nguồn mở	25
❑ Chi tiết các giấy phép FLOSS	35
❑ Các cộng đồng FLOSS	47
❑ Các mô hình sử dụng FLOSS	57
❑ Các mô hình kinh doanh	67
❑ FLOSS và các mô hình kinh doanh	75
❑ FLOSS và sự đổi mới	91
❑ Những rào cản với sự thích nghi nguồn mở	101
❑ Giới thiệu các tiêu chí lựa chọn phần mềm nguồn mở	119
❑ Kiến trúc dựa trên nguồn mở có khả năng liên thông	135
❑ FOSS-Bridge-Lab28	145
❑ Khám phá trung tâm lưu trữ (Forges) FLOSS tiềm năng	151
❑ Môi liên hệ và hiệu quả sử dụng của FLOSS Forges.....	159
❑ Tại sao và làm thế nào để xây dựng từ mã nguồn	169
❑ Kiểm soát phiên bản mã nguồn mở với Subversion.....	177
❑ Intellectual property and licenses.....	189
❑ FLOSS history, background, culture	201
❑ Free and open-source licenses	205
❑ Details of FLOSS Licenses	215
❑ FLOSS Communities	227
❑ Business Models	237

VỀ FOSS BRIDGE:

“FOSS Bridge EU-Vietnam - Hợp tác kinh doanh với Phần mềm Tự do và Mã nguồn mở” là một chương trình ASIA-Invest thực hiện bởi InWEnt (Tổ chức Nâng cao Năng lực CHLB Đức, INRIA (Viện nghiên cứu quốc gia về CNTT và Tự động hóa – Pháp) và IOIT (Viện Công nghệ thông tin - Việt Nam) với sự hỗ trợ tài chính của Ủy ban Châu Âu và Bộ Phát triển và Hợp tác kinh tế Liên bang Đức (BMZ). Xin xem thêm thông tin tại: <http://www.foss-bridge.org>



GIẤY PHÉP:

Creative Commons Attribution-Share Alike 3.0 Germany
Bản quyền: InWEnt Về giấy phép Creative Commons Attribution-Share Alike 3.0 Germany: Tư liệu này được cấp phép dưới giấy phép Creative Commons Attribution-Share Alike 3.0 Germany. Bạn được tự do sao chép, phân phối, truyền và thích ứng tác phẩm. Để xem bản sao của giấy phép này, hãy thăm <http://creativecommons.org/licenses/by-sa/3.0/de/deed.en> hoặc gửi thư tới Creative Commons, 171 phố Second, Suite 300, San Francisco, California, 94105, Mỹ. Xin lưu ý là giấy phép này đòi hỏi sự cho phép của tất cả các tác giả trong các phiên bản tương lai, và cần đưa vào lời công nhận: “Dựa trên tư liệu huấn luyện tạo bởi Christophe NEY cho dự án “FOSS Bridge EU-Vietnam - Hợp tác kinh doanh thông qua Phần mềm Tự do và Mã nguồn mở”, InWEnt. Số tham chiếu của tư liệu huấn luyện: [TRM1702400700]. Để có thêm thông tin mời xem <http://www.foss-bridge.org>. Căn cứ vào giấy phép, người giữ bản quyền (InWEnt) không chuyển nhượng các phiên bản tương lai của tư liệu này hay việc sử dụng lại nó.”

Xin lưu ý các chỉ dẫn và lời phủ nhận sau:

Tất cả thương hiệu được đề cập trong tài liệu này và được bảo vệ bởi các bên thứ ba tùy thuộc vào các điều khoản của luật thương hiệu thích hợp và các quyền sở hữu của chủ sở hữu đã đăng ký.

Căn cứ vào giấy phép, người giữ bản quyền (InWEnt) không chuyển nhượng các phiên bản tương lai của tư liệu này hay việc sử dụng lại nó.

Tài liệu này được tạo ra với sự giúp đỡ tài chính của Ủy ban Châu Âu và Bộ Phát triển và Hợp tác kinh tế Liên bang Đức (BMZ).

Các tác giả chịu trách nhiệm về nội dung của tài liệu này, và dưới bất cứ góc độ nào nó cũng không phản ánh quan điểm của Cộng đồng Châu Âu và Bộ Phát triển và Hợp tác kinh tế Liên bang Đức (BMZ).

Bên cạnh thẩm quyền dưới giấy phép, InWEnt trân trọng đề nghị nhận được thông báo ngắn gọn trong trường hợp việc sử dụng tư liệu chỉ rõ ngữ cảnh sử dụng/sửa đổi và số người đạt tới. Mọi phản hồi xin gửi về [ict\[at\]inwent.org](mailto:ict[at]inwent.org).

5 TUYÊN BỐ VỀ FLOSS

SOẠN BỞI: FRANÇOIS LETELLIER
CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
PHIÊN BẢN/ NGÀY: FOSS-BRIDGE-BLOCK1 /2008/04/17

Được tạo ra theo Dự án "FOSS Bridge EU-Vietnam - Hợp tác kinh doanh với Phần mềm Tự do & Nguồn mở" (<http://www.foss-bridge.org>) - do InWEnt (Tổ chức Nâng cao Năng lực CHLB Đức), INRIA (Viện nghiên cứu quốc gia về CNTT và Tự động hóa của Pháp) và IOIT (Viện Công nghệ thông tin - Việt Nam). Dự án được hỗ trợ tài chính của Ủy ban Châu Âu và Bộ Phát triển và Hợp tác kinh tế Liên bang Đức (BMZ). Xin xem thêm thông tin tại <http://www.foss-bridge.org>.

Số tham chiếu của tư liệu huấn luyện: [TRM1702400700].

Căn cứ vào giấy phép, người giữ bản quyền (InWEnt) không chuyển nhượng các phiên bản tương lai của tư liệu này hay việc sử dụng lại nó. Xin xem thêm các lưu ý ở trang cuối cùng của tài liệu này.

BÂY GIỜ CHÚNG TA SẼ HỌC...

- ☐ **Khái niệm thời đại số/xã hội thông tin**
- ☐ **Hàng hóa vật lý và hàng hóa thông tin**
- ☐ **Lật tẩy những chi tiết hoang đường hoặc quan niệm sai lầm về FOSS**

HÀNG HÓA PHI VẬT CHẤT (1/2)

- ☐ **Phần mềm là một dạng hàng hóa phi vật chất**
- ☐ **Vô hình**
 - ❖ Không sờ được
 - ❖ Phải được truyền tải qua vật trung gian
- ☐ **Có thể tái sinh**
 - ❖ Một cách hoàn hảo
 - ❖ Không bị giới hạn
- ☐ **Không cạnh tranh (non rival)**
 - ❖ Khi dùng bởi người này thì không ảnh hưởng đến việc sử dụng của người khác
 - ❖ Nhưng loại trừ

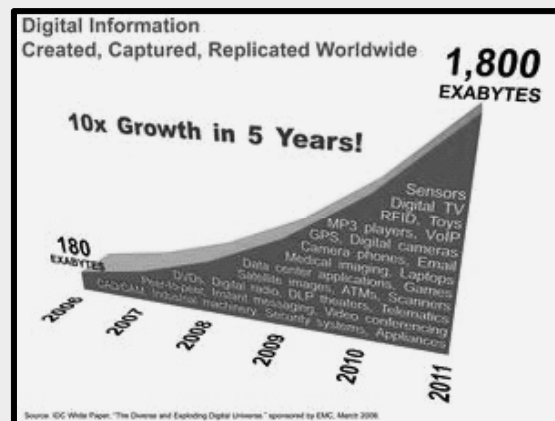
HÀNG HÓA PHI VẬT CHẤT (2/2)

- ☐ **Phần mềm là một dạng hàng hóa phi vật chất**
- ☐ **Nhưng lại cần phải được « thể hiện » ra trong thế giới vật chất**
- ☐ **Hàng hóa cần thử: khó biết trước chất lượng (với phần mềm thì điều này chỉ đúng 1 phần)**
- ☐ **Không có sự khan hiếm tự nhiên**
- ☐ **Các tác động ngoại lai mạng lưới**
- ☐ **Hệ quả là: khó xác định giá trị thực chất**

THỜI ĐẠI SỐ

- Hậu «thời đại công nghiệp»
- “Thổ dân số”: những thế hệ đầu tiên sinh ra có máy tính ở gia đình
- 281 EB thông tin số được tạo ra hoặc sao chép trong năm 2007 trên toàn thế giới. (Exabyte = 10 mũ 18 byte - đúng ra là 2 mũ 60)
- Tỷ lệ tăng trưởng 57% / năm (khoảng 1.000 EB đến 2010)
- ≈ 70% thông tin số được tạo ra bởi cá nhân vào năm 2010.

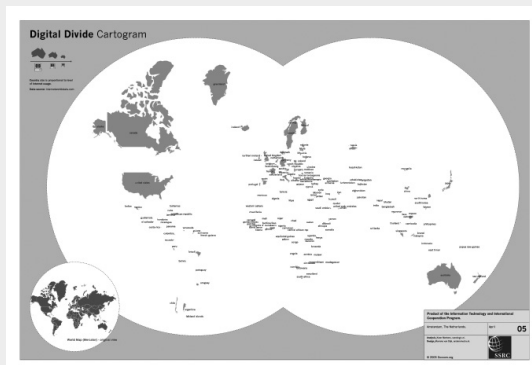
Nguồn: IDC «Sự đa dạng và bùng nổ của thế giới số», 2008



THỜI ĐẠI SỐ

❑ Mặt kia của đồng xu: Khoảng cách số

- ❖ 1,5 tỷ người kể nối Internet năm 2010 (còn những người khác thì sao?)
- ❖ Phần mềm tự do/nguồn mở có thể là một phần của giải pháp nhằm thu hẹp khoảng cách số

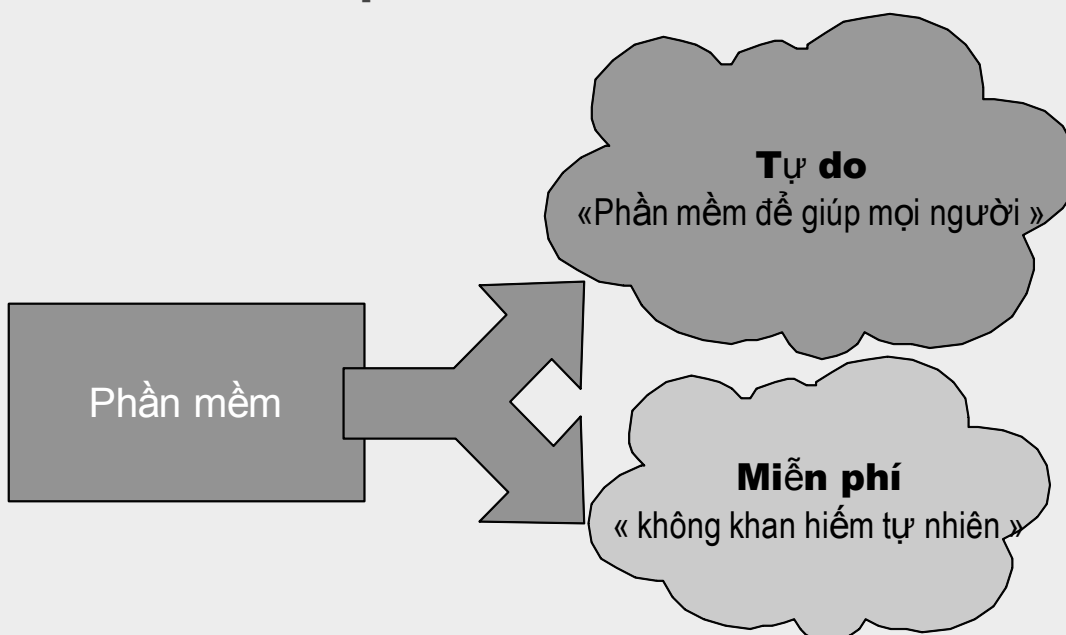


Nguồn: govcom.org - R. Rogers

TỪ KHÔNG KHAN HIỂM TỚI KHÔNG TỒN PHÍ

- ❑ **Khi các động lực của thị trường áp dụng...**
 - ❖ Các nền kinh tế có quy mô hướng tới giảm giá thị trường mục tiêu về không
 - ❖ Nếu giá = 0, hợp tác phát triển có ý nghĩa về kinh tế
- ❑ **Khi các động lực của thị trường không áp dụng (chẳng hạn như các sáng kiến được cộng đồng gây quỹ)**
 - ❖ Phần mềm có thể được phân phối rộng rãi với chi phí rất thấp
- ❑ **Hệ quả: có thể tìm được rất nhiều phần mềm mà không mất phí (với người dùng)**

PHẦN MỀM « TỰ DO »



THUẬT NGỮ

- FLOSS: Free / Libre / Open Source Software
- Free «theo nghĩa tự do ngôn luận, không phải bia uống miễn phí»
- Mã nguồn mở: thuật ngữ được đưa ra để tránh sự nhầm lẫn của từ «free» (trong tiếng Anh)
- Libre: từ gốc Latin, hay được dùng trong Cộng đồng chung Châu Âu thay cho «free»
- Free $\approx$ Tự do $\approx$ Phần mềm mã nguồn mở

NỀN KINH TẾ MỚI THỜI NAY, ĐÃ CHỌN ĐÚNG CHƯA?

❑ Các bong bóng dotcom vỡ...

- ❖ 2001: thị trường chứng khoán dotcom trải qua một giai đoạn đổ vỡ to lớn
- ❖ phương châm «phình to nhanh» đòi hỏi mức đầu tư đáng kể, nhưng các mô hình kinh doanh đã thất bại trong việc sinh ra lợi nhuận ổn định.



❑ Điều gì đã thay đổi?

- ❖ kỳ vọng thấp hơn nhiều (bài học đã trải qua)
- ❖ khoảng 20 năm dữ liệu lưu trữ về các mô hình kinh doanh đã được chứng minh.

QUAN NIỆM SAI LẦM VỀ PMTD&MNM (1/3)

- ❑ **Đó là sự đối đầu giữa Linux và Windows:**
 - ❖ Sự thật: hơn 400.000 dự án PMTD&MNM (18,000 đạt đến độ chín) trong mọi lĩnh vực CNTT
- ❑ **PMTD&MNM không ổn định và không được hỗ trợ:**
 - ❖ Sự thật: các giải pháp PMTD&MNM lớn ổn định hơn cả đối thủ độc quyền; có dịch vụ hỗ trợ chuyên nghiệp cho PMTD&MNM
- ❑ **Các công ty lớn không dùng PMTD&MNM**
 - ❖ Sự thật: khoảng 90% trong số 1000 công ty Fortune triển khai PMTD&MNM; IBM, Sun, HP, Oracle... tích cực ủng hộ PMTD&MNM

QUAN NIỆM SAI LẦM VỀ PMTD&MNM (2/3)

- ❑ **ĐPMTD&MNM vi phạm sở hữu trí tuệ**
 - ❖ Sự thật: các giấy phép PMTD&MNM là dựa trên luật bản quyền
- ❑ **Không thể làm ra tiền trên PMTD&MNM**
 - ❖ Sự thật: HP thu về 2,5 tỷ đô la năm 2003; RedHat được 400 triệu đô la năm 2006
- ❑ **Phong trào PMTD&MNM không công bằng và không bền vững, vì các lập trình viên không được trả tiền cho những nỗ lực của họ.**
 - ❖ Sự thật: > 50% các nhà phát triển PMTD&MNM được trả công, số còn lại là người tình nguyện và đóng góp vì động cơ của bản thân.

QUAN NIỆM SAI LẦM VỀ PMTD&MNM (3/3)

- ❑ **Nếu tôi khởi động một dự án, nhiều nhà phát triển sẽ bắt đầu làm việc cho tôi mà chẳng nhận lại gì**
 - ❖ Sự thật: sự tăng trưởng của cộng đồng cần có đầu tư đáng kể
- ❑ **PMTD&MNM là vấn đề của người lập trình, còn người dùng và các nhà lãnh đạo không cần phải lo lắng về nó**
 - ❖ Sự thật: PMTD&MNM hướng mũi nhọn vào một mô hình sáng kiến mới mà tất cả đều nên biết
- ❑ **PMTD&MNM luôn chơi trò rượt đuổi với thế giới thương mại, nơi khởi nguồn mọi sáng kiến**
 - ❖ Sự thật: tỷ lệ các dự án có tính đổi mới (12%) như nhau giữa PMTD&MNM và phần mềm độc quyền

TÓM TẮT

- ❑ **Xã hội chúng ta đang bước vào thời đại số**
- ❑ **Hàng hóa vật chất và phi vật chất khác nhau một cách cơ bản**
- ❑ **Tính kinh tế của phần mềm hơi khác so với hàng hóa hữu hình**
- ❑ **PMTD&MNM có ý nghĩa kinh tế đối với ngành phần mềm**
- ❑ **Quan niệm sai và sự thật về phần mềm tự do/mã nguồn mở**

GIẤY PHÉP VÀ ĐỘC QUYỀN SỞ HỮU TRÍ TUỆ

SOẠN BỞI: FRANÇOIS LETELLIER
CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 3 / 2008/04/17

BÂY GIỜ CHÚNG TA SẼ HỌC...

- ☐ **Khung điều chỉnh chung cho bảo hộ phần mềm**
- ☐ **Các quy tắc của quốc tế và từng quốc gia**
- ☐ **“Tài sản trí tuệ” phát minh, thương hiệu, bản quyền**
- ☐ **Nhắc lại về các giấy phép độc quyền**

DỰ ÁN TỔNG THỂ VỀ PMNM Ở VIỆT NAM

(ngày 2/3/2004)

❑ **Ứng dụng và phát triển phần mềm nguồn mở tại VN trong giai đoạn 2004-2008**

- ❖ Đẩy mạnh ứng dụng và phát triển PMNM
- ❖ Tăng cường bảo vệ bản quyền
- ❖ Cắt giảm chi phí mua phần mềm
- ❖ Khuyến khích phát triển CNTT của VN
- ❖ Hình thành một đội ngũ chuyên gia kỹ thuật có trình độ
- ❖ Tạo ra một số sản phẩm CNTT tiêu biểu

BẢO HỘ PHẦN MỀM: KHUNG PHÁP LÝ

❑ **Chẳng có cái gì là “luật quốc tế”**

- ❖ Các luật quốc gia
- ❖ Thỏa thuận giữa các nước
- ❖ Luật Mỹ khác luật “quốc tế”

❑ **Điều gì là khác biệt giữa các nước?**

- ❖ Có lẽ là mọi thứ
- ❖ Ngôn ngữ
- ❖ Những thành tố bắt buộc (VD: trong các hợp đồng)
- ❖ Sự chối bỏ về những đảm bảo, những giới hạn trách nhiệm pháp lý
- ❖ Quyền tác giả -> quyền sở hữu tác phẩm (VD: các quyền về đạo đức)
- ❖ Khả năng phần mềm được cấp sáng chế

BẢO HỘ PHẦN MỀM: KHUNG PHÁP LÝ - CÁC SÁNG CHẾ

- ❑ **Sắc lệnh bảo vệ quyền sở hữu công nghiệp (28/1/1989)**
- ❑ **Hiệp định ASEAN trong hợp tác sở hữu trí tuệ (1996) - Hiệp ước GATT/Trips của WTO; phần 5, khoản 27: «có thể dùng sáng chế vào bất cứ phát minh nào»**
- ❑ **Bị loại khỏi bảo hộ sáng chế (ở Việt Nam):**
 - ❖ Phần mềm máy tính, mạch tích hợp, mô hình toán, đồ thị tìm kiếm hay những thứ tương tự.
 - ❖ Ý tưởng khoa học, lý thuyết và các phát hiện; phương pháp tổ chức và quản lý kinh tế; giáo dục, dạy học, đào tạo; ngôn ngữ và các hệ thống thông tin, các ký hiệu quy ước, thời gian biểu, luật và quy tắc, biểu tượng...
- ❑ **Phần mềm có thể được bảo hộ ở một số nước (VD: Mỹ)**
 - ❖ Không như bản quyền: các phát minh được tạo ra một cách độc lập vẫn đòi hỏi giấy phép từ bất cứ bằng sáng chế khác trước khi sử dụng

BẢO HỘ PHẦN MỀM: KHUNG PHÁP LÝ - THƯƠNG HIỆU

- ❑ Từ, cụm từ, biểu tượng, thiết bị hay bất cứ sự kết hợp nào, dùng để phân biệt hàng hóa/dịch vụ/tư cách hội viên của một người
- ❑ Đăng ký (nguyên tắc đệ trình trước tiên) là bắt buộc - trừ các thương hiệu nổi tiếng
- ❑ Quyền ngăn chặn mọi cá nhân không được sự chấp thuận của chủ sở hữu:
 - ❖ Sử dụng các ký hiệu giống hệt hoặc tương tự
 - ❖ Cho các sản phẩm/dịch vụ giống hệt, tương tự hoặc có liên quan, là nguyên nhân dẫn đến nhiều khả năng nhầm lẫn
- ❑ Sử dụng trái phép là vi phạm pháp luật.



BẢO HỘ PHẦN MỀM: KHUNG PHÁP LÝ

- ❑ Phần mềm máy tính
 - ❖ Được coi như là một phần của "các tác phẩm văn học, nghệ thuật hoặc khoa học"
 - ❖ Không được cấp bằng sáng chế
 - ❖ Được bảo vệ bởi bản quyền
 - ❖ Tại Việt Nam: nhận thức của người dân còn hạn chế, mức phạt thấp, sự thi hành luật chưa nghiêm



Biểu tượng Bản Quyền

BẢO HỘ PHẦN MỀM: KHUNG PHÁP LÝ

- ❑ **Độc quyền dành cho người sở hữu**
 - ❖ Quyền đạo đức: quyền ghi công, giữ nguyên bản
 - ❖ Quyền cá nhân: xuất bản, sao chép, diễn xuất, trưng bày, phân phối, tạo ra tác phẩm
- ❑ **Có thể chuyển nhượng các quyền, tách rời hay cùng nhau (giới hạn ở một số nước)**
- ❑ **Vi phạm bản quyền:**
 - ❖ Giống nhau đáng kể và có tiếp cận tới tác phẩm bị vi phạm
 - ❖ Sử dụng bất hợp pháp

BẢO HỘ PHẦN MỀM: KHUNG PHÁP LÝ

- ❑ **Để sử dụng tác phẩm được bảo hộ (như phần mềm):**
 - ❖ Phải ký hợp đồng ("giấy phép") với tác giả hoặc người sở hữu.
 - Không được nhận bất cứ quyền nào nếu không ghi ra rõ ràng.
 - Người được cấp giấy phép không có nghĩa vụ phải chấp nhận nó.
 - ❖ Phải kèm theo các điều dự phòng
 - Phần mềm được sử dụng như thế nào
 - Phạm vi và khoảng thời gian
 - Phí bản quyền tác giả (nếu có)
 - Trách nhiệm của các bên nếu hợp đồng bị phá vỡ
- ❑ **Sử dụng các tác phẩm được bảo hộ đã xuất bản vào mục đích phi thương mại không được áp dụng với phần mềm (VN)**

BẢO HỘ PHẦN MỀM: PHẠM VI CÔNG CỘNG

- ❑ **Phần mềm (cũng như các tác phẩm khác) được đưa vào đó sẽ ở trạng thái không một ai nắm bất cứ quyền pháp lý nào**
 - ❖ Theo lựa chọn ("Tặng" cho Phạm vi Công cộng)
 - ❖ Sau khi thời hạn bảo hộ chấm dứt (như các tác phẩm văn học cổ điển)
 - ❖ Theo luật định (VD: dữ liệu được tạo ra từ dịch vụ hành chính công ở Mỹ)
- ❑ **Là miễn phí và hợp pháp dưới bất cứ hình thức sử dụng nào, bởi bất kỳ ai, cho bất cứ mục đích (hợp pháp) nào.**

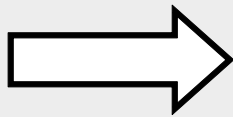
BẢO HỘ PHẦN MỀM: KHUNG PHÁP LÝ

- ❑ **Ai là "tác giả"?**
 - ❖ Người trực tiếp tạo ra toàn bộ hay một phần của tác phẩm văn học, nghệ thuật, khoa học
 - ❖ Người dịch tác phẩm đó
 - ❖ Người chỉnh sửa một tác phẩm đã có, viết lại hoặc chuyển thể một tác phẩm
 - ❖ Người biên soạn, chú giải hay chọn lọc các tác phẩm
- ❑ **Nếu có nhiều hơn một người: đồng tác giả**

LƯỢC ĐỒ TỔNG THỂ CÁC GIẤY PHÉP PHẦN MỀM

Được thuê bởi...
Nhượng lại bản quyền cho

Bán hoặc cấp giấy phép để :
sử dụng và/hoặc phân phối và/
hoặc sửa đổi... cho



Tác giả

Người sở hữu quyền

**Người được cấp
giấy phép**

Ví dụ:

Kỹ sư phần mềm Công ty (người tuyển dụng)

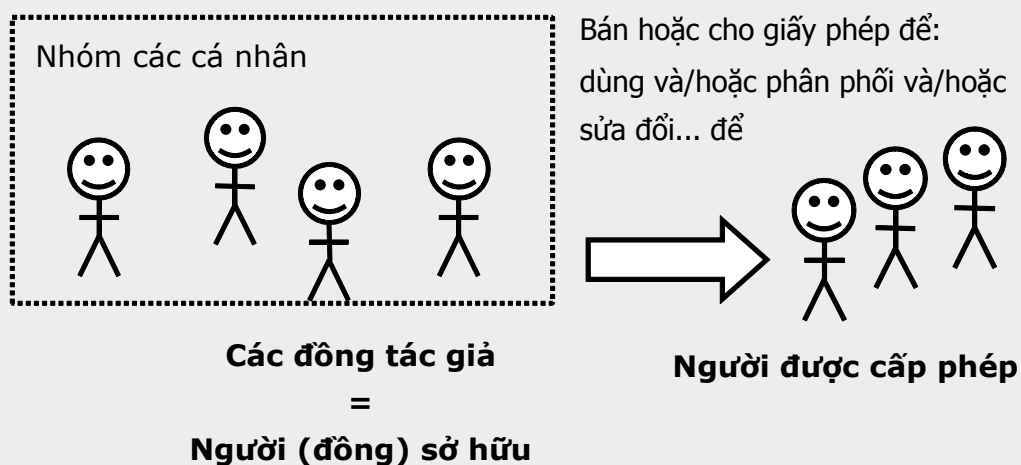
Người dùng cuối

BẢO HỘ PHẦN MỀM: KHUNG PHÁP LÝ

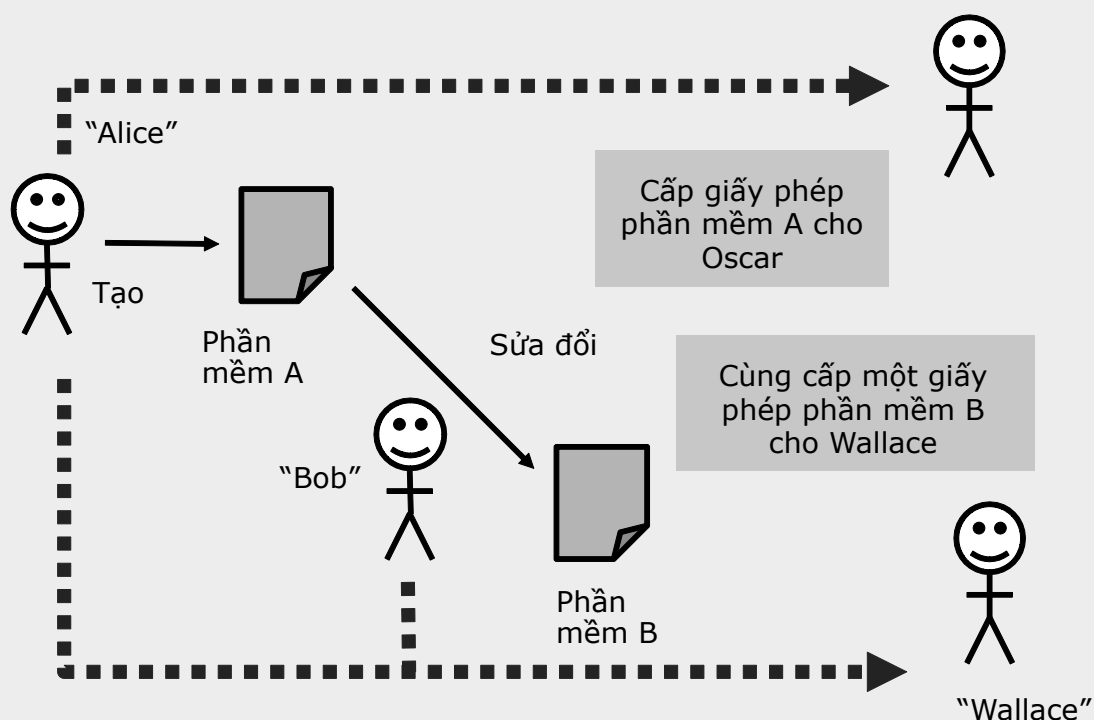
□ Ai là “người sở hữu” hợp pháp các quyền?

- ❖ Mặc định thì đó là tác giả -hoặc đồng tác giả
- ❖ Nhà chức trách hoặc tổ chức đã giao phó trách nhiệm cho một tác giả hoặc ký hợp đồng để tạo ra tác phẩm
- ❖ Những ngoại lệ có thể có (VD: quyền đạo đức)
- ❖ Người thừa kế di chúc hoặc người thừa kế theo luật định của tác giả
- ❖ Người được thừa hưởng theo chỉ định của người sở hữu tác phẩm

CÁC GIẤY PHÉP PHẦN MỀM KHI CÁC ĐỒNG TÁC GIẢ = CÁC CÁ NHÂN



CÁC GIẤY PHÉP PHẦN MỀM CHUỖI LIÊN DANH



TÓM TẮT

- ☐ Phần mềm chủ yếu được bảo hộ (khả năng loại trừ pháp lý) bởi luật bản quyền.
- ☐ Chẳng có gì giống với luật quốc tế, vì thế có sự phức tạp trong các ngữ cảnh quốc tế.
- ☐ Các (đồng) tác giả $\neq$ Người sở hữu các quyền
- ☐ Trừ các tác phẩm trong khu vực công cộng, không quyền nào được trao trừ phi nhận được giấy phép.

VĂN HÓA, NỀN TẢNG VÀ LỊCH SỬ FOSS

SOẠN BỞI: FRANÇOIS LETELLIER
CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 4 /2008/04/17

BÂY GIỜ CHÚNG TA SẼ HỌC...

- ☐ **FLOSS lịch sử, nền tảng, văn hóa**
- ☐ **Ai là ai - những câu đại thụ về quan điểm**

BILL GATES (NGƯỜI HÙNG NÀO CŨNG CẦN 1 KẼ XẤU)

- ❑ Đồng sáng lập Microsoft năm 1975
- ❑ Người đề xướng mạnh mẽ về sở hữu trí tuệ
- ❑ “ Ai có thể làm việc chuyên môn chẳng vì cái gì? ”
- ❑ 2007: người giàu nhất thế giới (59 tỷ đô la, Forbes)



RICHARD M. STALLMAN

- ❑ MIT 1971-1984
- ❑ “ Tự do ngôn luận, không phải bia miễn phí ”
- ❑ Sáng lập quỹ FSF (1984)
- ❑ Khởi đầu phát triển GNU (Emacs)
- ❑ Tạo ra giấy phép copylefted đầu tiên (GPL)



St. IGNUCIUS

ERIC S. RAYMOND

- ❑ Tác giả cuốn “Nhà thờ và chợ trời”
- ❑ “Câm miệng và đưa cho họ mã nguồn”
- ❑ Sáng lập Open Source Initiative (1998)



LINUS TORVALDS

- ❑ Linus -> “Linux”
- ❑ Nổi tiếng vì đã đưa nhân Linux đến dự án GNU
- ❑ Được OSDL thuê để lãnh đạo việc phát triển nhân Linux



BRUCE PERENST

- ❑ Tác giả chính của cuốn “Định nghĩa mã nguồn mở”
- ❑ Người thuyết giáo mã nguồn mở
- ❑ Cũng làm việc để thúc đẩy các chuẩn mở



LAWRENCE LESSIG

- ❑ Giáo sư luật tại Đại học Stanford
- ❑ Tác giả cuốn “Mã và các luật của không gian điều khiển”
- ❑ Sáng lập Creative Commons



TÓM TẮT

- ❑ **Lịch sử FLOSS khởi nguồn từ những ngày đầu của máy tính**
- ❑ **Phong trào “Phần mềm tự do” xuất hiện như một phản ứng đáp trả xu hướng đang lên của phần mềm thương mại**
- ❑ **Tự do khác với nguồn mở: khía cạnh quan trọng của phong trào là triết lý**
- ❑ **Những nhà lãnh đạo ngoan cố và lời cuốn**

CÁC GIẤY PHÉP TỰ DO NGUỒN MỞ

SOẠN BỞI: FRANÇOIS LETELLIER
CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 5 /2008/04/17

BÂY GIỜ CHÚNG TA SẼ HỌC...

- ☐ “Phần mềm tự do” là gì
- ☐ “Phần mềm mã nguồn mở” là gì
- ☐ “Copyleft” là gì và không là gì
- ☐ Phần mềm không phải “tự do” cũng không phải “mã nguồn mở” gọi là gì

PHẦN MỀM BIỂU KHÔNG

☐ **Freeware**

- ❖ Không mất tiền, mã đóng

☐ **Shareware**

- ❖ Không bắt buộc phải trả tiền, mã đóng

☐ **Bản chạy thử (demo)**

- ❖ Đôi khi cho không, sau đó người dùng phải trả tiền

☐ **Phần mềm Tự do, Phần mềm Mã nguồn mở**

- ❖ Không có nghĩa cho không - mặc dù thường không mất tiền
- ❖ Không có nghĩa "lĩnh vực công" - có áp dụng bảo hộ và cấp giấy phép

PHẦN MỀM TỰ DO - ĐỊNH NGHĨA

☐ **AFree trong "Tự do Ngôn luận" không phải "Bia Miễn phí"**

☐ **Bốn quyền "tự do" (nguồn: Free Software Foundation)**

- ❖ 0 - chạy chương trình, với bất cứ mục đích gì
- ❖ 1 - nghiên cứu cách chương trình làm việc, và sửa đổi cho phù hợp với nhu cầu của mình
- ❖ 2 - phân phối lại các bản sao để có thể giúp cả hàng xóm
- ❖ 3 - cải tiến chương trình, và đưa ra những cải tiến cho cộng đồng, để tất cả đều hưởng lợi

☐ **Truy xuất tới mã nguồn là điều kiện tiên quyết cho quyền tự do 1 và 3**

PHẦN MỀM MÃ NGUỒN MỞ - ĐỊNH NGHĨA (1/3)

Định nghĩa của “phần mềm mã nguồn mở”

(Nguồn: Open Source Initiative)

- ☐ **1. Tự do phân phối lại**
 - ❖ Kèm theo như là thành phần của phần mềm khác
- ☐ **2. Mã nguồn**
 - ❖ Kèm theo hoặc có sẵn
- ☐ **3. Các sản phẩm kế thừa**
 - ❖ Những thay đổi có thể được cấp phép y như vậy

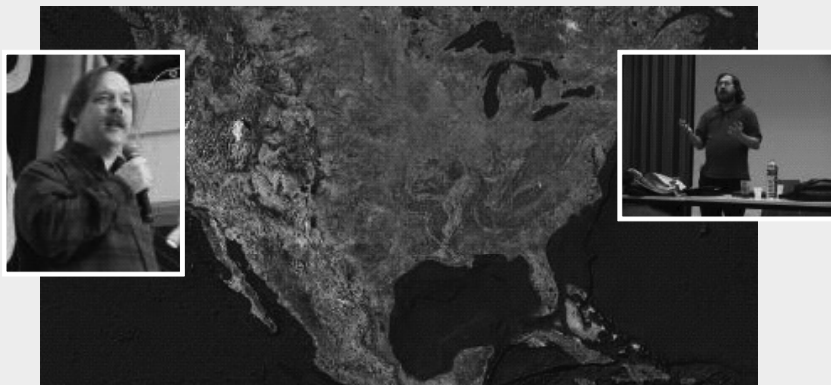
PHẦN MỀM MÃ NGUỒN MỞ - ĐỊNH NGHĨA (2/3)

- ☐ **4. Tính nguyên vẹn trong mã nguồn của tác giả**
 - ❖ Có thể phân phối các tác phẩm kế thừa, nhưng có thể cần thay đổi về tên hoặc sử dụng các tệp vá (patch)
- ☐ **5. Không phân biệt đối xử với người hoặc nhóm**
 - ❖ Cũng như không thể cho một công ty thêm các quyền
- ☐ **6. Không phân biệt đối xử các lĩnh vực**
 - ❖ Không thể cấm sử dụng vào những ngữ cảnh nhất định

PHẦN MỀM MÃ NGUỒN MỞ - ĐỊNH NGHĨA (3/3)

- ❑ **7. Phân phối giấy phép**
 - ❖ Không đòi hỏi giấy phép nào khác
- ❑ **8. Giấy phép không được riêng biệt cho một sản phẩm**
 - ❖ Giữ nguyên quyền nếu tách ra và sử dụng chỗ khác
- ❑ **9. Giấy phép không được hạn chế phần mềm khác**
 - ❖ Nếu chỉ đơn thuần phân phối trên cùng phương tiện
- ❑ **10. Giấy phép phải trung lập với công nghệ**
 - ❖ Không thể dựa vào công nghệ hoặc kiểu cách giao diện

KHÁC BIỆT VĂN HÓA TỰ DO VỚI MÃ NGUỒN MỞ



- ❖ Đạo đức (FSF) vs thực dụng / “thân thiện với kinh doanh” (OSI)
- ❖ Đồng nghĩa với FLOSS: Free / Libre / Open Source Software
- ❖ “Mã đóng” hoặc “tư hữu” gọi tắt = phi tự do
- ❖ FSF tạo ra các giấy phép của riêng mình, OSI phê chuẩn các giấy phép đệ trình

NGHE THÌ GIỐNG FOSS NHƯNG KHÔNG PHẢI FOSS

- ❑ **Cho không ≠ FOSS**
- ❑ **Truy cập được đến mã nguồn không có nghĩa là FOSS**
 - ❖ VD: phần mềm được đặt hàng; “nguồn chia sẻ” (Microsoft)
- ❑ **Được phát tán tự do không có nghĩa là FOSS**
- ❑ **Có mọi quyền FOSS nhưng hạn chế (VD: sử dụng với mục đích thương mại) cũng không có nghĩa FOSS**
 - ❖ VD: Scilab

“SỰ NÀY NỜ” GIẤY PHÉP

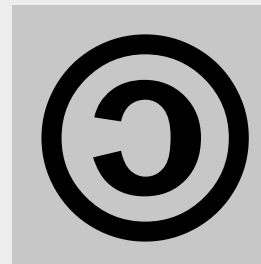
- ❑ **1000 (?) giấy phép phục tùng định nghĩa FOSS (Saugatuck 07)**
 - ❖ Nhiều triết lý / mô hình kinh doanh
 - ❖ Phạm vi quốc tế
- ❑ **61 được OSI xác nhận (APR 08)**
- ❑ **Những hạn chế**
 - ❖ Phức tạp với người sử dụng
 - ❖ Khả năng không tương thích
- ❑ **Câu trả lời của OSI: phân loại**

PHỔ BIẾN VÀ ĐƯỢC SỬ DỤNG RỘNG RÃI HOẶC CÓ CỘNG ĐỒNG MẠNH

- ☐ **Apache License, 2.0**
- ☐ **New BSD license**
- ☐ **GNU General Public License (GPL version 2)**
- ☐ **GNU Library or "Lesser" General Public License (LGPL version 2)**
- ☐ **MIT license**
- ☐ **Mozilla Public License 1.1 (MPL)**
- ☐ **Common Development and Distribution Licence (CDDL)**
- ☐ **Common Public Licence (CPL)**
- ☐ **Eclipse Public Licence (EPL)**

"COPYLEFT"

- ☐ **Dựa trên copy-right**
- ☐ **Người sở hữu quyền cấp quyền để**
 - ❖ Sử dụng, sửa đổi, phân phối lại
- ☐ **Dưới một số điều kiện "qua lại":**
 - ❖ Trong trường hợp phân phối...
 - ❖ ...phải giao lại đúng mọi quyền đã nhận được
 - ❖ Tùy vào giấy phép, nghĩa vụ này có áp dụng cho các đóng góp hay không
- ☐ **Tồn tại giấy phép copylefted cho cả phần mềm, âm nhạc, và nghệ thuật, ...**



QUAN NIỆM SAI VỀ COPYLEFT

Các niềm tin phổ biến sau đây là không đúng:

- ☐ **Nghĩa vụ phân phối các tác phẩm phái sinh**
- ☐ **Nghĩa vụ cho phép truy xuất công khai tới các tác phẩm phái sinh**
- ☐ **“Sự lây nhiễm” của hệ thống thông tin**
 - ❖ Thuật ngữ FUD: “tính vi-rút”
 - ❖ Nhầm lẫn giữa sản phẩm lắp ghép với dẫn xuất
 - ❖ Lẫn lộn giữa phát triển và thực thi
- ☐ **Không tương hợp với hoạt động thương mại**

CÁC GIẤY PHÉP FOSS COPYLEFTED TỨC CÁC GIẤY PHÉP “ĐÀO”

- ☐ **GPL - GNU Public licence**
 - ❖ v1 (1989), v2 (1991), v3 (2007)
 - ❖ VD: nhân Linux, PERL, MySQL
- ☐ **LGPL - Lesser GNU Public licence**
 - ❖ Được thiết kế cho các thư viện
 - ❖ V: glibc, JBoss AS
- ☐ **CPL - Common Public licence**
 - ❖ VD: Microsoft Windows Installer XML
- ☐ **EPL - Eclipse Public licence**
 - ❖ VD: nền Eclipse

GIẤY PHÉP NON-COPYLEFTED TỨC GIẤY PHÉP “HÀN LÂM”

- ☐ Lấy cảm hứng từ “tự do học thuật”
- ☐ Đầu tiên trong lịch sử: BSD (Berkeley, < 1980)
- ☐ ≈ “Người sở hữu bản quyền phần mềm này sau đây cấp phép cho bạn vì bất cứ mục đích nào”
- ☐ Thông thường, các phủ nhận về bảo đảm và trách nhiệm
- ☐ Các giấy phép BSD (mới), MIT, Apache 1.0, Artistic

TÓM TẮT

- ☐ FOSS ≠ phần mềm miễn phí, mặc dù thường có thể lấy miễn phí
- ☐ Phần mềm tự do = định nghĩa của FSF
- ☐ Nguồn mở = tiêu chuẩn của OSI
- ☐ Phân phối FOSS không phải là một nghĩa vụ
- ☐ “copyleft” là một nguyên tắc nghịch đảo
- ☐ Một số giấy phép FOSS không phải là copylefted

CHI TIẾT CÁC GIẤY PHÉP FLOSS

SOẠN BỞI: FRANÇOIS LETELLIER
CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 6 /2008/04/17

BÂY GIỜ CHÚNG TA SẼ HỌC...

- ☐ **Nghiên cứu chi tiết các giấy phép FLOSS**
- ☐ **Xem xét các vấn đề về tính tương thích giữa các giấy phép**
- ☐ **Khám phá các giấy phép tài liệu tự do**

GNU GENERAL PUBLIC LICENSE 2

- ❑ **Phạm vi: tính copyleft mạnh**
 - ❖ Sự đền đáp bao trùm các sản phẩm chứa/dẫn xuất
- ❑ **Giấy phép con:**
 - ❖ Không có giấy phép con; giấy phép được cấp từ tác giả gốc trên mỗi lần phân phối
- ❑ **Cấp sáng chế:**
 - ❖ Không cấp sáng chế rõ ràng, nhưng bản thân giấy phép hàm ý 1 sáng chế
- ❑ **Trả đũa:**
 - ❖ Cấm phân phối trừ phi tất cả có thể phân phối miễn phí

GNU LESSER GENERAL PUBLIC LICENSE 2.1

- ❑ **Phạm vi: copyleft**
 - ❖ Bao trùm các sản phẩm chứa/dẫn xuất, chỉ hạn chế (chứ không yêu cầu phải có sự đền đáp lại của) các giấy phép.
- ❑ **Giấy phép con:**
 - ❖ Giống như GPL
- ❑ **Cấp sáng chế:**
 - ❖ Giống như GPL
- ❑ **Trả đũa:**
 - ❖ Giống như GPL

COMMON PUBLIC LICENSE 1.0 ECLIPSE

PUBLIC LICENSE 1.0

❑ **Phạm vi: copyleft**

- ❖ Sự đền đáp không được áp dụng cho các đơn thể phân phối cùng với phần mềm nhưng lại không phải là các sản phẩm dẫn xuất của phần mềm

❑ **Giấy phép con:**

- ❖ Có thể cấp phép lại cho mã đối tượng nếu giấy phép gồm có các lời báo trước và tuyên bố phủ nhận.

❑ **Cấp sáng chế:**

- ❖ Cấp trực tiếp cho phần đóng góp và kết hợp

❑ **Trả đũa: đòi hỏi phi quyền lợi có thể thu hồi**

- ❖ (riêng với CPL) kiện tụng về sáng chế chống lại người đóng góp đối với một sáng chế có thể ứng dụng vào phần mềm - khi đó bất cứ giấy phép sáng chế được cấp bởi người đóng góp cho người nhận sẽ chấm dứt
- ❖ Kiện tụng về sáng chế chống lại bất cứ thực thể nào viện ra rằng bản thân chương trình vi phạm sáng chế của người nhận – khi đó quyền của những người nhận như thế đã được cấp bởi giấy phép sẽ chấm dứt.

APACHE LICENSE 2.0

- ❑ **Phạm vi: hàn lâm**
 - ❖ Không đền đáp lại
- ❑ **Giấy phép con:**
 - ❖ Có thể cung cấp các điều khoản trong giấy phép thêm hoặc khác và các điều kiện chỉnh sửa
- ❑ **Cấp sáng chế:**
 - ❖ Cấp rõ ràng cho những đóng góp hoặc kết hợp
- ❑ **Trả đũa:**
 - ❖ Rất giống EPL

MOZILLA PUBLIC LICENSE 1.1

- ❑ **Phạm vi:**
 - ❖ Sự đền đáp dựa trên tệp tin
- ❑ **Giấy phép con:**
 - ❖ Phân phối lại bản thực thi dưới điều khoản khác
- ❑ **Cấp sáng chế:**
 - ❖ Cấp sáng chế một cách rõ ràng cho "phiên bản của người đóng góp"
- ❑ **Trả đũa:**
 - ❖ Mất sáng chế của người đóng góp và các nhượng quyền có xác nhận chống lại người đóng góp
 - ❖ Mất các nhượng sáng chế của người đóng góp nếu có xác nhận chống lại bất cứ "phần mềm, phần cứng, hoặc thiết bị"

NEW BSD

- ❑ **Phạm vi: hàn lâm**
 - ❖ Không đền đáp lại
- ❑ **Giấy phép con:**
 - ❖ Không đề cập đến giấy phép con, nhưng hay được thừa nhận
- ❑ **Cấp sáng chế:**
 - ❖ Không cấp sáng chế, chỉ có từ “sử dụng”
- ❑ **Trả đũa:**
 - ❖ Không

MIT

- ❑ **Phạm vi: học thuật**
 - ❖ Giống như BSD
- ❑ **Giấy phép con:**
 - ❖ Giấy phép con thì rõ, cấp phép lại thì ngầm ẩn
- ❑ **Cấp sáng chế:**
 - ❖ Không có các điều khoản rõ ràng
- ❑ **Trả đũa:**
 - ❖ Y như BSD

GIẤY PHÉP CÔNG CỦA EU

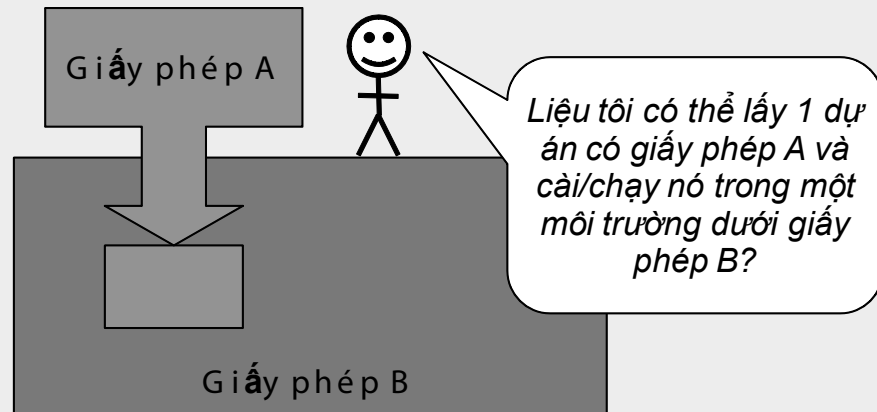
- ❑ Đề xuất bởi IDABC, Tháng Một 2007, chấp thuận bởi Ủy ban Châu Âu
- ❑ Tuân theo luật của tất cả các quốc gia Châu Âu
- ❑ Các phiên bản ở 23 ngôn ngữ chính thức của EU (January 2008) có cùng giá trị pháp lý
- ❑ Copylefted, tương thích với CPL, EPL, GPL, OSL, và CeCILL
- ❑ Hiện đang được OSI duyệt lại

CÁC ĐIỀU KHOẢN SÁNG CHẾ TRONG GIẤY PHÉP FLOSS

- ❑ **Nhắc lại: phần mềm được cấp sáng chế ở một số nước (Mỹ)**
- ❑ **Những loại chính của các điều khoản sáng chế trong giấy phép FLOSS**
 - ❖ Không (VD: BSD mới, MIT)
 - ❖ Có thể thu hồi những gì không phải là đòi quyền lợi (VD: MPL, CPL, EPL, CDDL)
 - ❖ Vĩnh viễn, toàn cầu, không độc chiếm, miễn phí, không tiền tiền quyền, không thể thu hồi (VD: Giấy phép Apache 2.0)
 - ❖ Những ràng buộc về phân phối trong trường hợp có thể có vi phạm sáng chế (VD: *GPL)
- ❑ **Nhận xét: việc cấp các giấy phép sáng chế bởi nhà phát triển đầu tiên hoặc người đóng góp không thể đảm bảo người dùng không vi phạm bất cứ sáng chế nào của bên thứ ba ở bất cứ nước nào**

TÍNH (KHÔNG) TƯƠNG THÍCH CỦA GIẤY PHÉP (1/5)

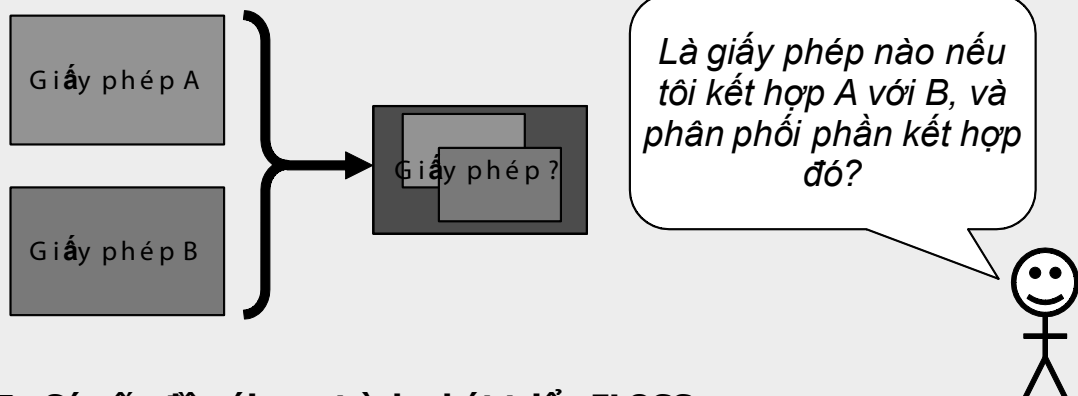
❑ Gộp phần mềm



❑ Có vấn đề cho người dùng cuối của FLOSS

TÍNH (KHÔNG) TƯƠNG THÍCH CỦA GIẤY PHÉP (2/5)

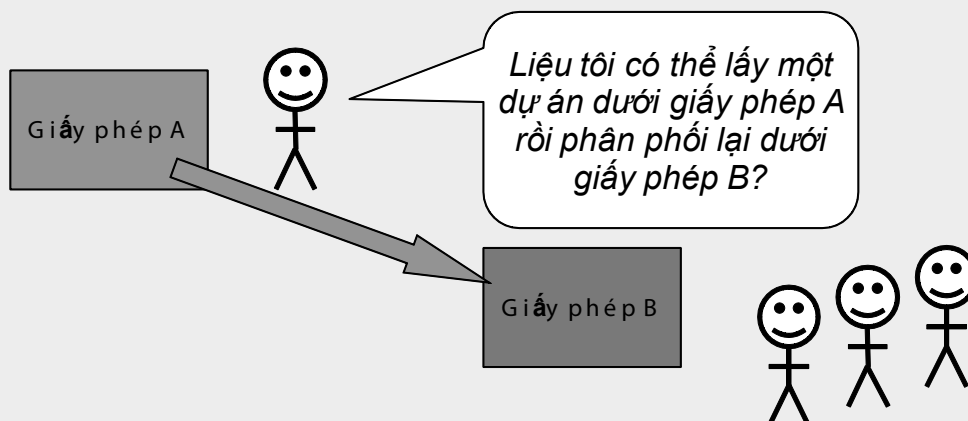
❑ Kết hợp phần mềm dưới các giấy phép khác nhau



❑ Có vấn đề với quy trình phát triển FLOSS

TÍNH (KHÔNG) TƯƠNG THÍCH CỦA GIẤY PHÉP (3/5)

❑ Thay đổi giấy phép để phân phối lại



❑ Có vấn đề với các mô hình kinh doanh dựa trên FLOSS

TÍNH (KHÔNG) TƯƠNG THÍCH CỦA GIẤY PHÉP (4/5)

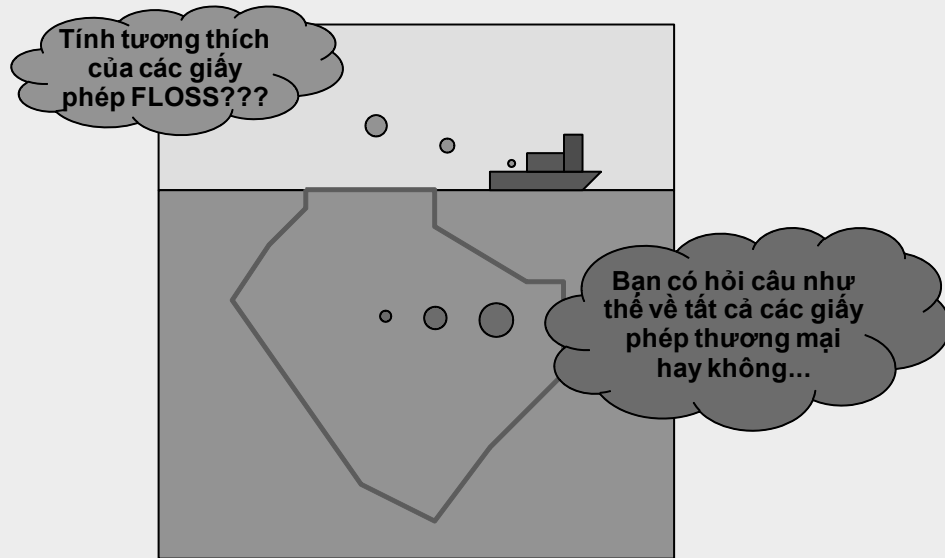
❑ Không có câu trả lời đơn giản – tùy trường hợp

- ❖ Chi tiết, phân tích pháp lý các giấy phép
- ❖ Các quốc gia, luật có thể áp dụng
- ❖ Các bên có dính líu (pháp lý/vật lý chẳng hạn)

		DERIVATIVE WORK				
		GPL	MPL	CPL	OSL	Academic
CONTRIBUTION	BSD	yes	no ¹	no ²	yes	yes ³
	MIT	yes	no ¹	no ²	yes	yes ³
	Apache	yes ⁴	no ¹	no ²	yes	yes ³
	AFL	yes ⁴	no ¹	no ²	yes	yes ³

Nguồn: « Open Source Licensing - Software Freedom and Intellectual Property Law » bởi Lawrence Rosen

TÍNH (KHÔNG) TƯƠNG THÍCH CỦA GIẤY PHÉP (5/5)



CÁC GIẤY PHÉP TÀI LIỆU (1/3)

- ☐ Tài liệu về khái niệm thường được cấp phép với phần mềm
- ☐ Các loại tài liệu khác:
 - ❖ Sách, báo, sổ tay, wiki, web site, slideshows, tác phẩm nghệ thuật, nội dung đa phương tiện...
 - ❖ Bảo vệ bởi luật bản quyền
 - ❖ thường không được cấp phép với phần mềm

CÁC GIẤY PHÉP TÀI LIỆU (2/3)

❑ **GNU Free Documentation License (GFDL)**

- ❖ Tạo bởi quỹ FSF
- ❖ Tương đương với GPL cho các sản phẩm không phải là phần mềm.
- ❖ Copylefted
- ❖ Khái niệm “bản sao trong suốt” (= mã nguồn) và “bản sao mờ” (= mã khả thi)

CÁC GIẤY PHÉP TÀI LIỆU (3/3)

❑ **Creative Commons (CC)**

- ❖ Người tạo nên: Lawrence Lessig
- ❖ Bảng phân công các giấy phép tạo bởi các luật sư cho phù hợp:
 - Nhiều nước / ngôn ngữ
 - Từ “giữ tất cả quyền” tới “giữ một số quyền”
 - Người sở hữu quyết định các quyền và ngôn ngữ – và đảm bảo cho giấy phép có lời lẽ pháp lý ăn khớp

TÓM TẮT

- ❑ **Phổ rộng các thuật ngữ giấy phép, từ copyleft mạnh cho tới các giấy phép “hàn lâm” rất rộng rãi về quyền**
- ❑ **Những vấn đề tương thích giấy phép xuất hiện khi sử dụng, phát triển và kinh doanh trên FLOSS – đồng thời nó cũng đúng với phần mềm thương mại.**
- ❑ **Các nguyên tắc của FLOSS cũng đã được chuyển sang cho các giấy phép tài liệu**

CÁC CỘNG ĐỒNG FOSS

SOẠN BỞI: FRANÇOIS LETELLIER

CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT

PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 9 /2008/04/17

BÂY GIỜ CHÚNG TA SẼ HỌC...

- ☐ Các cộng đồng FLOSS tiêu biểu
- ☐ Hacker và văn hóa FLOSS
- ☐ Mô tả sơ lược về các thành viên cộng đồng

DỰ ÁN FLOSS

- ❑ **Dự án – đơn vị đo cơ bản trong các cộng đồng FLOSS**
 - ❖ Phát triển cộng tác
 - ❖ Hướng công nghệ – chủ yếu là tạo ra mã
- ❑ **Các lãnh đạo dự án**
 - ❖ Thường có mặt ngay từ khi dự án khai sinh, giám sát được những hướng đi chính của nó.
 - ❖ Cá tính và uy tín là chìa khóa
- ❑ **Thành viên trụ cột “đôi khi được tổ chức như PMC”**
 - ❖ Đóng góp đáng kể qua thời gian
 - ❖ Thường nhỏ hơn 15 người (lớn hơn giới hạn này, có nhiều khả năng dự án phải chia thành các dự án con)
- ❑ **Người dùng tích cực (aka “những người đóng góp”)**
 - ❖ Báo lỗi, viết tài liệu, thỉnh thoảng vá lỗi

KỸ NGHỆ CỘNG TÁC CDE (1/3)



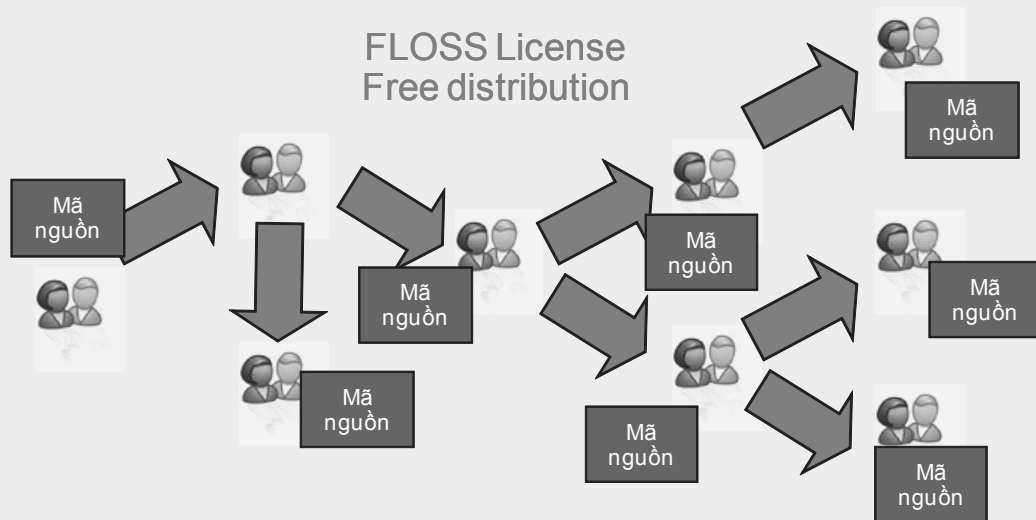
KỸ NGHỆ CỘNG TÁC CDE (2/3)

- ☐ **Mailing lists, diễn đàn – hình thành kho tri thức**
- ☐ **Wikis – tài liệu trực tuyến**
- ☐ **Quản lý mã nguồn – cho phép phát triển song song**
 - ❖ CVS, SVN
- ☐ **Săn lỗi**
 - ❖ Bugzilla, TRAC
- ☐ **Các công cụ quản trị, quản lý người dùng, thống kê, tích hợp liên tục, ...**

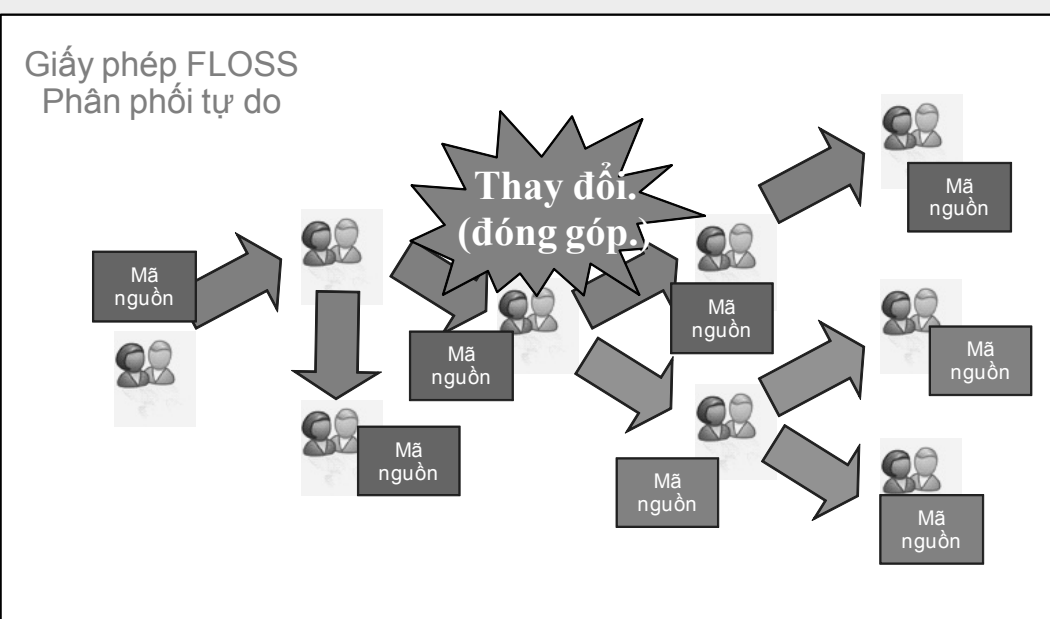
KỸ NGHỆ CỘNG TÁC CDE (3/3)

- ☐ **Cảnh báo: việc tập trung mã nguồn được thực hiện bởi các lý do thực tế**
- ☐ **Giấy phép không yêu cầu điều đó ít (bên cạnh các ngoại lệ)**
- ☐ **Hệ quản là:**
 - ❖ Không thể truy xuất công khai vào cơ sở mã nguồn của một số dự án.
 - ❖ Một số dự án có cơ sở mã nguồn phân tán
 - ❖ Các dự án có thể rẽ nhánh...

RỄ NHÁNH DỰ ÁN (1/3)



RỄ NHÁNH DỰ ÁN (2/3)



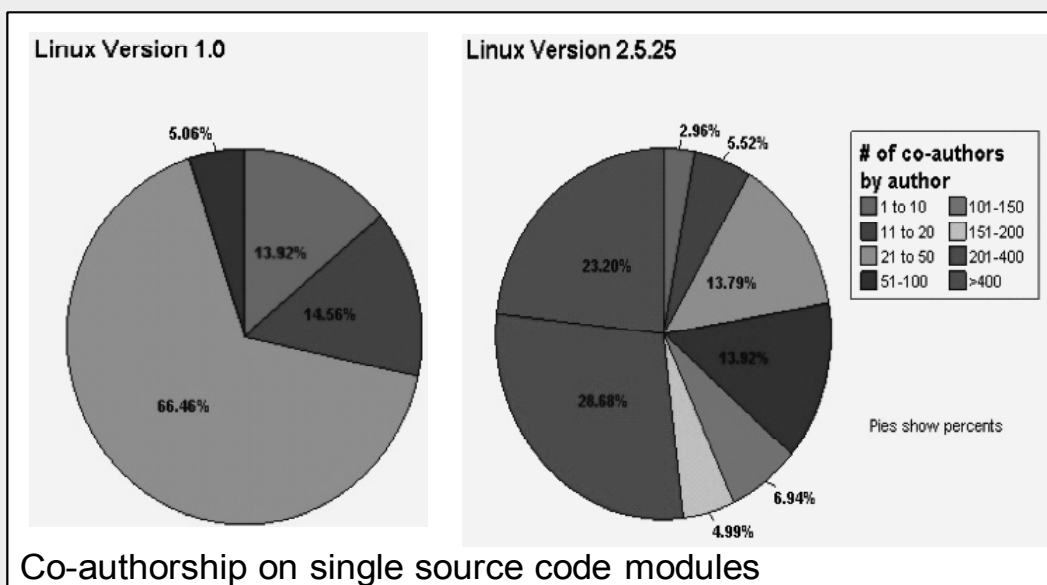
RỄ NHÁNH DỰ ÁN (3/3)

- ❑ **Khi một dự án sinh ra hai biến thể, thường bất đồng dần dần**
- ❑ **Nhiều lý do có thể**
 - ❖ Kỹ thuật (VD: chuyển sang các nền khác)
 - ❖ Con người (VD: xung đột trong lãnh đạo)
 - ❖ Kinh doanh (VD: mở đầu một đề nghị thương mại mới)
- ❑ **Rễ nhánh là tốt nhất... và cũng là điều tồi tệ nhất.**
- ❑ **“Cái phù hợp nhất sẽ sống sót”**

CÁC CỘNG ĐỒNG FLOSS Ự TIẾN HÓA

- ❑ **Các dự án tiến hóa**
 - ❖ Bằng cách bổ sung thêm nhà phát triển mới
 - ❖ Qua di chuyển các nhà phát triển từ các dự án đang tồn tại
- ❑ **Các dự án lớn nhất tăng trưởng nhanh hơn**
 - ❖ Do đó mới có hiện tượng của “khối lượng tới hạn” (critical mass)
- ❑ **Các dự án hình thành quanh một nòng cốt các nhà phát triển có kết nối với nhau.**
- ❑ **Thông tin chảy theo những phụ thuộc của dự án**

CỘNG TÁC PHÁT TRIỂN NHÂN LINUX



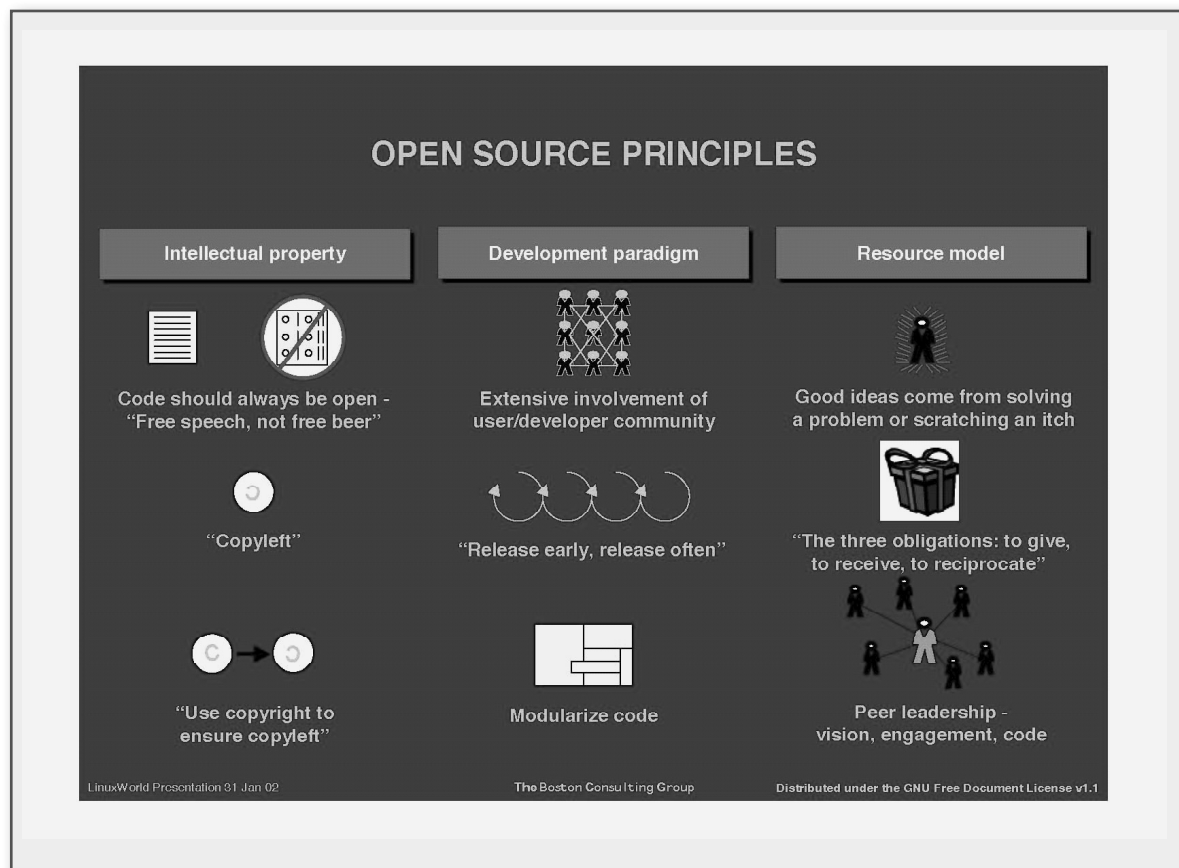
FLOSS & VĂN HÓA HACKER

❑ Đạo đức hacker (Steven Levy)

- ❖ Truy xuất vào máy tính phải không bị hạn chế
- ❖ Mọi thông tin phải được tự do
- ❖ Nghi ngờ nhà chức trách, khuyến khích phân quyền
- ❖ Hacker phải được đánh giá thông qua hành động, chứ không phải các tiêu chí như địa vị, tuổi, chủng tộc, vị trí
- ❖ Bạn có thể tạo ra nghệ thuật và cái đẹp trên máy tính
- ❖ Máy tính có thể thay đổi cuộc sống của bạn, làm tốt hơn

❑ Ý thức trách nhiệm như bản sao của tự do

❑ Ý nghĩa của từ "hacker" đã bị méo mó trong những năm 90



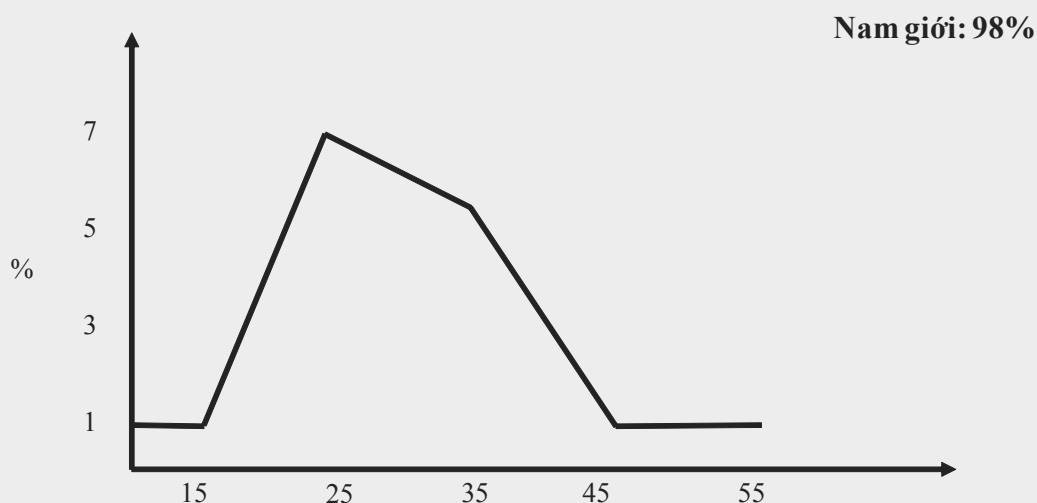
ĐỘNG CƠ VÀ SƠ LƯỢC VỀ CÁC NHÀ PHÁT TRIỂN FOSS

❑ Các động cơ chính (BCG Hacker Survey 2002)

- ❖ Người thích học – viết mã để cải thiện kỹ năng và cả thú vui (29%)
- ❖ Người thích làm – đều bị kích thích bởi cơ hội phát triển các kỹ năng cho các hoạt động ngoài công việc (27%)
- ❖ Tay chuyên nghiệp – tham gia cộng đồng OSS để nâng cao kỹ năng cá nhân, chuyên nghiệp (25%).
- ❖ Người tin vào cộng đồng – tin tưởng mạnh mẽ rằng nên mở mã nguồn (19%)

❑ 65% không đóng góp từ cơ quan – hoặc không được các cấp bậc lưu tâm

MỘT NHÀ PHÁT TRIỂN FLOSS TIÊU BIỂU (NĂM 2002)



THIÊN LỆCH GIỚI TÍNH TRONG CÁC CỘNG ĐỒNG FLOSS

- ❑ **% nữ giới: <2% trong FLOSS / 28% trong phần mềm độc quyền.**
- ❑ **Phụ nữ chủ động bị loại ra khỏi các cộng đồng FLOSS (nếu vô tình) thay vì vụ lợi thụ động**
 - ❖ Đạo đức 'hacker', đặt mình bên ngoài tính xã hội 'chủ đạo' sociality, và làm cân bằng phụ nữ với xu hướng chủ đạo đó
 - ❖ FLOSS thường công bằng tạo ra mã thay vì tạo ra phần mềm
 - ❖ Phụ nữ có xu hướng bắt đầu với máy tính muộn hơn. Để tham gia, họ có một khối lượng lớn công việc phải làm để bắt kịp
 - ❖ Chấp nhận khẩu chiến, với một hiệu ứng gây khó chịu, tỏ ý đặc thù trong trường hợp của nữ
 - ❖ Nam giới thường tự do hơn khi tham gia so với nữ giới nhiều giờ làm việc tập trung với máy tính

TÓM TẮT

- ☐ Dự án là đơn vị đo cơ bản trong các cộng đồng FLOSS
- ☐ Các dự án tiến hóa, rẽ nhánh
- ☐ Thường được tổ chức quanh một cơ sở mã nguồn, thành viên của cộng đồng đóng góp ở nhiều mức độ
- ☐ Các nhà phát triển tiêu biểu theo đuổi các động cơ cá nhân (không phải chủ yếu vì lợi nhuận)
- ☐ Đa đa số là nam giới, 25-35 tuổi, văn hóa hacker

CÁC MÔ HÌNH SỬ DỤNG FOSS

SOẠN BỞI: FRANÇOIS LETELLIER

CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT

PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 13 /2008/08/08

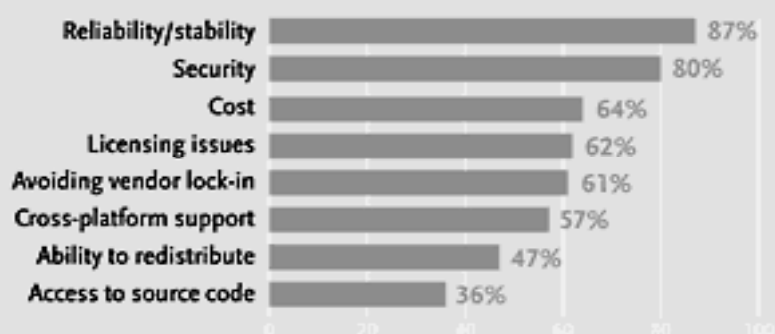
BÂY GIỜ CHÚNG TA SẼ HỌC...

- ☐ **Nhận thức về sử dụng FLOSS**
- ☐ **Hòa trộn FLOSS / độc quyền**
- ☐ **Các mô hình sử dụng FLOSS**
- ☐ **Làm sao để quyết định mô hình sử dụng và lựa chọn FLOSS**

LỢI ÍCH CỦA FLOSS TỪ LẬP TRƯỜNG CỦA NGƯỜI DÙNG

Business Goes Back to the Source

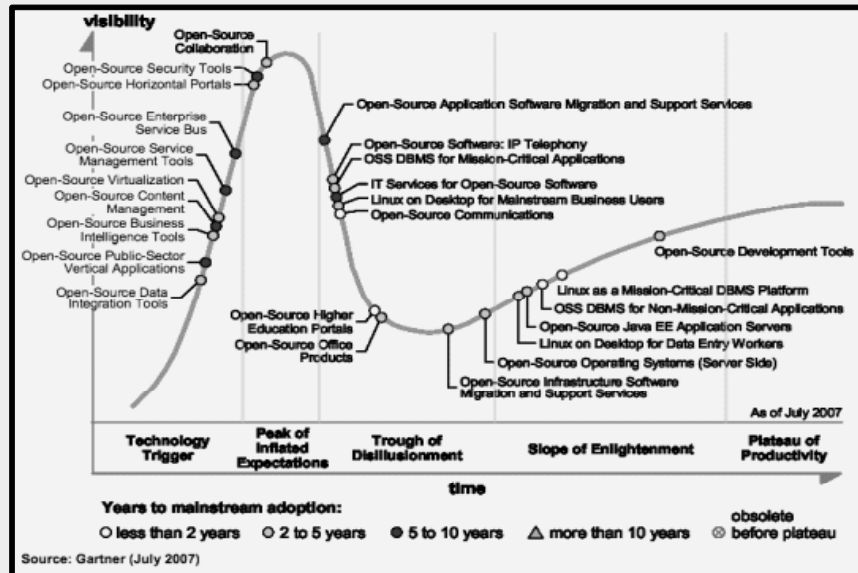
In a July 2004 *InfoWorld*/IDG survey, respondents rated a number of criteria as being important or very important to the appeal of open source — but source code itself hardly topped the list.



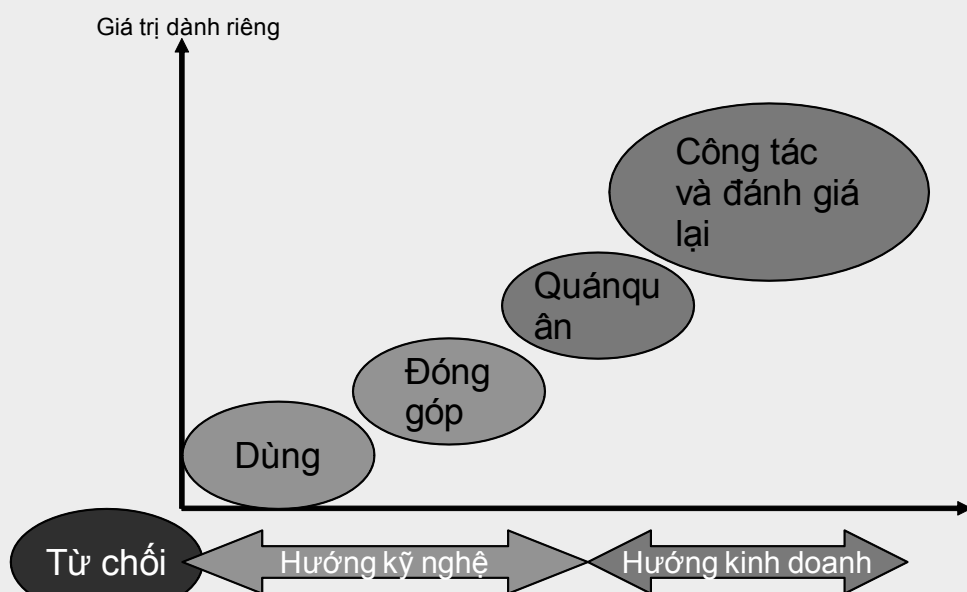
NHẬN THỨC VIỆC SỬ DỤNG FLOSS

- ❑ « Chúng tôi không dùng FLOSS, nhưng chúng tôi dùng Linux, Apache, MySQL! »
- ❑ Châu Âu, khu vực công, 2005:
 - ❖ 80% sử dụng FLOSS
 - ❖ 30% không biết về nó...
 - ❖ Tỷ lệ tương tự trong công nghiệp
- ❑ Trong các công ty tin học lớn, quản lý các giấy phép độc quyền có thể lớn gấp 10-20 lần so với quản lý các giấy phép FLOSS

CHU TRÌNH CƯỜNG ĐIỀU CHO PHẦN MỀM MÃ NGUỒN MỞ CỦA GARTNER, 2007



THANG CHẤP NHẬN FLOSS



Nguồn: dự án FLOSSMetrics

SỬ DỤNG FOSS - CÁC TRỌNG TÂM CHI PHÍ

❑ Dùng

- ❖ Xác định phần mềm tiềm năng đang quan tâm
- ❖ Chỉnh sửa, chuyển đổi, huấn luyện

❑ Đóng góp

- ❖ Thời gian phát triển

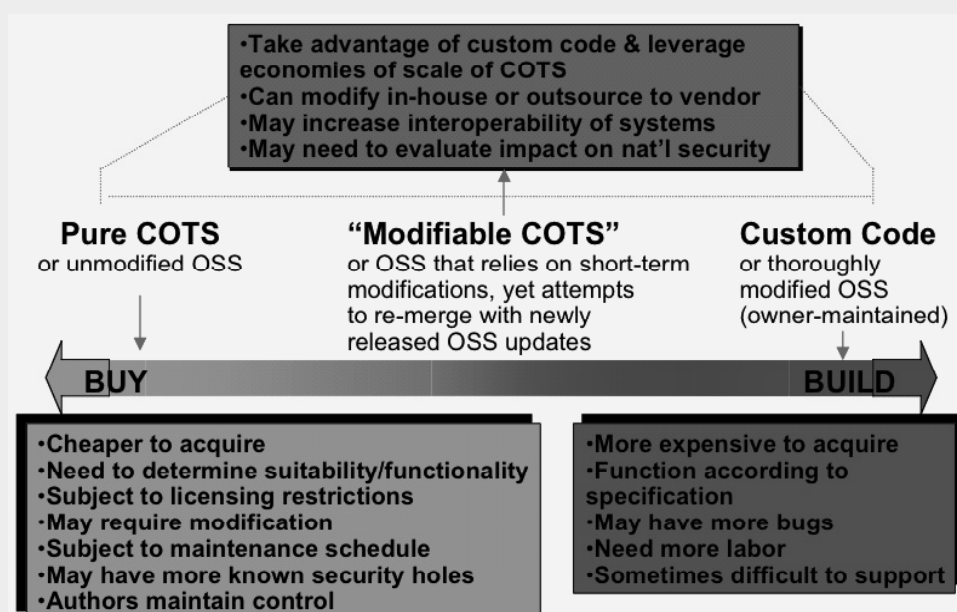
❑ Quán quân

- ❖ Thời gian phát triển
- ❖ Tương tác cộng đồng và tài trợ
- ❖ Hỗ trợ các bên thứ ba

❑ Đánh giá lại

- ❖ Thời gian phát triển
- ❖ Dự án, kết hợp hệ sinh thái

MUA HAY XÂY



Nguồn: *A Business Case Study of Open Source Software*, Carolyn A. Kenwood, 2001

CÁC MÔ HÌNH SỬ DỤNG DÙNG LẠI (XÀI CHÙA)

- ❑ **Dùng FLOSS trong sản xuất**
 - ❖ Ngăn chặn chi phí (TCO)
 - ❖ Sự linh động
 - ❖ Dùng các chuẩn thực tế (de facto)
- ❑ **Chiếm đại đa số người dùng**
- ❑ **Chấp nhận là một dạng đóng góp**
- ❑ **Các vấn đề chính:**
 - ❖ Đi bước đầu tiên từ từ chối sang sử dụng
 - ❖ Các kỹ năng có sẵn
 - ❖ Niềm tin rằng FLOSS là không mất chút chi phí nào hết

CÁC MÔ HÌNH SỬ DỤNG CUNG ỨNG KÉP

- ❑ **Giống-tốt-nhất / chiến thuật thực dụng**
 - ❖ Quyền lực cạnh tranh
 - ❖ Tránh bẫy khóa-cứng
 - ❖ Khả năng mở rộng không giới hạn
 - ❖ Chuyển đổi từng bước qua các nền tảng lai ghép
- ❑ **Những vấn đề chính:**
 - ❖ Quản lý sự hỗn tạp, không đồng nhất
 - ❖ Hiệu ứng « danh sách tính năng »
 - ❖ Bối rối vì các quy trình mua sắm phần mềm đã thiết lập

CÁC MÔ HÌNH SỬ DỤNG - SÀNG LỌC

❑ **Mở đầu bằng phát triển bên trong**

- ❖ Đóng góp mã nguồn mở để chuyển công đoạn bảo trì và phát triển ra bên ngoài
- ❖ Cộng đồng có thể đóng góp các cải tiến
- ❖ Tập trung hoạt động của công ty vào các giá trị gia tăng

❑ **Những vấn đề chính:**

- ❖ Rủi ro vì đánh giá thấp nỗ lực phải bỏ ra
- ❖ Quyết định và thuyết phục phần mã nguồn đã được đầu tư có thể được tung ra cho cộng đồng một cách an toàn.
- ❖ Sự nhút nhát của các nhà phát triển nội bộút chi phí nào hết

CÁC MÔ HÌNH SỬ DỤNG CHIA SẺ NGHIÊN CỨU + PHÁT TRIỂN

❑ **FLOSS được dùng trong cộng tác xuyên tổ chức**

- ❖ Nền linh động cho kỹ thuật cộng tác
- ❖ FLOSS cung cấp một khung pháp lý chìa khóa trao tay.
- ❖ Có thể được bảo hộ thông qua các phương tiện khác (VD: NDA) khi cần thiết
- ❖ Chiến lược đổi mới mở
- ❖ Thúc đẩy công nghệ và các chuẩn
- ❖ Giới hàn lâm và hành chính công dễ tham gia

❑ **Những vấn đề chính:**

- ❖ Cân bằng giữa các quyết định gốc rễ và các lộ trình từ trên xuống.

CÁC MÔ HÌNH SỬ DỤNG - LỢI ÍCH CỦA FLOSS TRONG CÁC HOẠT ĐỘNG

Tham gia hằng hái hơn



- ☐ Tái sử dụng FLOSS
- ☐ Tạo nguồn kép
- ☐ Mở ra các phát triển từ trong
- ☐ Chia sẻ nghiên cứu và phát triển



Các cơ hội kinh doanh trực tiếp hơn

TÌM VÀ LỰA CHỌN FLOSS

- ☐ **Xác định yêu cầu của bạn**
 - ❖ Danh sách ngắn các chức năng đòi hỏi và hữu dụng
 - ❖ Thêm các yêu cầu về dịch vụ (VD: hỗ trợ)
- ☐ **Tìm các FLOSS khớp với các yêu cầu**
 - ❖ Forge sites, tìm trên web, công cụ tìm kiếm trong mỗi bản phân phối
- ☐ **Chọn tập thích hợp các giải pháp FLOSS**
- ☐ **Tùy chọn, tinh lọc mô hình sử dụng của mình**

CHỌN CÁC GIẢI PHÁP FLOSS THÍCH HỢP (1/2)

❑ Dự án FLOSS, bản thân nó có

- ❖ số nhà phát triển, người đóng góp
- ❖ Hoạt động trên mailing lists, diễn đàn, tốc độ phát hành, số lượng tải xuống
- ❖ Chất lượng mã và độ trưởng thành, tài liệu sẵn có, các lỗi được báo và sửa, lộ trình
- ❖ Danh tiếng, sự nổi danh, tham chiếu
- ❖ Sự tuân theo các chuẩn mở (mặc dù không có trong yêu cầu)
- ❖ Kiểm thử, điểm chuẩn, thử nghiệm mới – đừng tin vào tờ rơi giới thiệu
- ❖ Giấy phép (& hợp pháp)

CHỌN CÁC GIẢI PHÁP FLOSS THÍCH HỢP (2/2)

❑ Cộng đồng

- ❖ Đăng sau dự án là ai? Các cá nhân hay một công ty?
- ❖ Mô hình kinh doanh là gì?
- ❖ Dự án có phải là một phần của cộng đồng có tổ chức?
- ❖ Các kỹ năng sẵn có
- ❖ Các dịch vụ chuyên nghiệp sẵn có

❑ Các phương pháp hình thức

- ❖ OpenBRR
- ❖ QS-OS

CHUYỂN ĐỔI SANG FLOSS CÁC THÓI QUEN HAY NHẤT)

- ☐ Phải chắc chắn về cam kết của cấp quản lý trong bước quá độ
- ☐ Phải rõ ràng về các kỳ vọng; làm sao có thể đo đếm được và chuẩn bị các điểm chuẩn
- ☐ Hãy thực tế... (chi phí, thời biểu)
- ☐ Luôn ưu tiên độ ổn định hơn là tính năng
- ☐ Duyệt lại thủ tục mua sắm và hỗ trợ CNTT
- ☐ Tránh các thay đổi lớn – tốt nhất là chuyển đổi dần
- ☐ Dành riêng nhân viên để liên lạc với cộng đồng
- ☐ Huấn luyện nhân viên trong quá trình FLOSS được phát triển
- ☐ Sử dụng tính linh động của FLOSS để cải tiến tại địa phương

TÓM TẮT

- ☐ Ngày nay, một công ty không thể lơ lửng chọn FLOSS
- ☐ Các công ty lớn nhất cũng đã sử dụng FLOSS – dù không phải lúc nào cũng ý thức được về nó
- ☐ Vài mô hình sử dụng – bước đầu tiên là từ chối bỏ sang xài chùa (dùng lại)
- ☐ Lựa chọn FLOSS và quản lý thay đổi là chìa khóa để chuyển đổi từ độc quyền sang FLOSS

CÁC MÔ HÌNH KINH DOANH

SOẠN BỞI: FRANÇOIS LETELLIER
CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 14 /2008/04/17

BÂY GIỜ CHÚNG TA SẼ HỌC...

- ☐ **“Mô hình kinh doanh” là gì**
- ☐ **Các công ty xây dựng các bản chào hàng thương mại như thế nào?**

“MÔ HÌNH KINH DOANH” LÀ GÌ

☐ **Mức hoạch định: Chiến lược**

☐ **Mức kiến trúc: Mô hình Kinh doanh**

☐ **Vận hành: Các tiến trình kinh doanh**

“MÔ HÌNH KINH DOANH” LÀ GÌ

☐ **Không chỉ là một mô hình lợi nhuận**

☐ **Bốn khía cạnh:**

- ❖ Quản lý hạ tầng
- ❖ Đổi mới sản phẩm
- ❖ Quan hệ khách hàng
- ❖ Tài chính

“MÔ HÌNH KINH DOANH” LÀ GÌ



- ❑ -> **lôgic kinh doanh của một công ty tạo ra lợi nhuận theo một cách bền vững**

“MÔ HÌNH KINH DOANH” LÀ GÌ

Sáng tạo sản phẩm	• Phân khúc khách hàng mục tiêu • Chuỗi giá trị • Khả năng
Quan hệ khách hàng	• Chiến lược thông tin • Cảm giác và phục vụ • Tin cậy và trung thành
Quản trị hạ tầng	• Nguồn lực • Cấu hình hành động • Mạng lưới đối tác
Tài chính	• Mô hình thu nhập • Cấu trúc chi phí • Lợi nhuận/thua lỗ

MÔ HÌNH CỔ ĐIỂN TRONG PHẦN MỀM CÔNG TY PHẦN MỀM

❑ Một cách ngắn gọn – và chuẩn:

- ❖ Một công ty phần mềm (có thể là một “ISV”) phát triển các sản phẩm phần mềm...
- ❖ trên các nghiên cứu nội bộ của riêng nó;
- ❖ Nhưng sản phẩm này được đóng gói như phần mềm COTS (có vỏ bọc thương mại)...
- ❖ Bảo vệ bản quyền (& các tùy chọn thương hiệu và bằng sáng chế);
- ❖ COTS được cấp giấy phép cho khách hàng với một chi phí nào đó;
- ❖ Thêm nữa, các dịch vụ cơ bản (hỗ trợ, bảo trì, v.v.).

MÔ HÌNH CỔ ĐIỂN TRONG PHẦN MỀM CÔNG TY PHẦN MỀM

- ❖ Lợi nhuận cao khi khối lượng tới hạn đạt được
- ❖ Nhưng dấu hiệu rủi ro của phản hồi không tốt trên sự đầu tư nghiên cứu phát triển
- ❖ Sự quan trọng của các hiệu ứng mạng đối với việc phát triển thị trường
- ❖ Nghiên cứu phát triển nội bộ cần đủ để giữ tính đổi mới
- ❖ Chiến lược khóa cứng là không phổ biến

MÔ HÌNH CỐ ĐIỂN TRONG PHẦN MỀM CÁC DỊCH VỤ CHUYÊN NGHIỆP

❑ Một cách ngắn gọn – và chuẩn:

- ❖ Một công ty cung cấp các dịch vụ chuyên nghiệp...
- ❖ Phân phối các dịch vụ có sức mạnh con người như đặt trước phát triển phần mềm/ tư vấn kỹ thuật / thuê ngoài...
- ❖ Tới khách hàng không chuyên IT hoặc như thầu phụ của các công ty IT.
- ❖ Mỗi hợp đồng dựa trên yêu cầu chi tiết của khách hàng...
- ❖ Và giá dựa trên ngày công

MÔ HÌNH CỐ ĐIỂN TRONG PHẦN MỀM CÁC DỊCH VỤ CHUYÊN NGHIỆP

- ❖ Đầu tư ban đầu giới hạn
- ❖ Giảm lợi nhuận thuần nhưng giới hạn rủi ro
- ❖ Nghiên cứu phát triển bên trong rất hạn chế, phần lớn sức mạnh con người kỹ thuật dành phục vụ khách hàng
- ❖ Thường là sự không thông hiểu kỹ thuật và điều chỉnh dựa trên các công nghệ khách hàng ưa thích
- ❖ Các kỹ năng/sự tinh thông (và sự lanh lợi) cần để giữ tính cạnh tranh và giữ sự trung thành của khách hàng
- ❖ Vấp ngã sự cạnh tranh bên ngoài

MÔ HÌNH CỔ ĐIỂN TRONG PHẦN MỀM BÁN LẠI GIÁ TRỊ GIA TĂNG

❑ Một cách ngắn gọn – và chuẩn:

- ❖ Một đơn vị bán lại giá trị gia tăng (VAR) đưa ra các giải pháp đã được tùy biến, trọn gói ...
- ❖ Tối chuyên viên không chuyên IT, các công ty người dùng cuối.
- ❖ Các giải pháp được xây dựng vỏ bọc thương mại (COTS), tùy biến theo các yêu cầu, ...
- ❖ Và đưa ra các dịch vụ chuyên nghiệp thêm vào, như tư vấn, đào tạo, bảo trì.
- ❖ Đơn vị bán lại giá trị gia tăng được giảm giá từ các giấy phép COTS do thỏa thuận với các công ty phần mềm ...
- ❖ Và tạo ra lợi nhuận thuần với việc bán lại giấy phép và các dịch vụ.

MÔ HÌNH CỔ ĐIỂN TRONG PHẦN MỀM BÁN LẠI GIÁ TRỊ GIA TĂNG

- ❖ Các đơn vị bán lại giá trị gia tăng cùng tiếp thị với công ty phần mềm và hành động như các kênh phân phối cho họ
- ❖ Các quyết định chiến lược phải đưa ra như công ty phần mềm nào với đối tác khi một vài đối thủ cạnh tranh đưa ra các sản phẩm tương tự
- ❖ Trong quan hệ giữa rủi ro và lợi nhuận, nhóm cân bằng giữa các công ty dịch vụ và phần mềm

TÓM TẮT

- ❑ **Các mô hình kinh doanh không chỉ là các mô hình thu nhập**
- ❑ **Bốn khía cạnh:**
 - ❖ Quản trị hạ tầng, đổi mới sản phẩm, các quan hệ khách hàng, tài chính
- ❑ **Công ty phần mềm, dịch vụ chuyên nghiệp, bán lại giá trị gia tăng là các mô hình chuẩn trong ngành công nghệ thông tin (IT)**

FLOSS VÀ CÁC MÔ HÌNH KINH DOANH

SOẠN BỞI: FRANÇOIS LETELLIER
CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 15 /2008/04/17

BÂY GIỜ CHÚNG TA SẼ HỌC...

- ☐ **Mô hình kinh doanh FLOSS**
- ☐ **Mối liên hệ giữa giấy phép và mô hình kinh doanh**
- ☐ **Cách để bạn tham gia vào mô hình kinh doanh FLOSS**

FLOSS ẢNH HƯỞNG ĐẾN CÁC MÔ HÌNH KINH DOANH NHƯ THẾ NÀO?

❑ Quản lý hạ tầng

- ❖ Các tài nguyên
 - Va chạm thấp hơn khi xây dựng các nhóm thông qua sự luyện tập văn hóa chung
- ❖ Cấu trúc hoạt động
 - Hoạt động có thể nhằm gia tăng giá trị
 - Giảm bất đồng bộ thông tin, hướng quan hệ nhà cung cấp/khách hàng tới cộng đồng hơn
- ❖ Mạng lưới đối tác
 - Thuê ngoài bên trong mạng lưới

FLOSS ẢNH HƯỞNG ĐẾN CÁC MÔ HÌNH KINH DOANH NHƯ THẾ NÀO?

❑ Đổi mới sản phẩm

- ❖ Phân đoạn khách hàng mục tiêu
 - Nâng cao danh tiếng và nhãn hiệu bằng cách cung cấp nhãn hiệu mạnh hơn
- ❖ Đề xuất giá trị
 - Extension of geographic coverage
 - Ability to leverage local expertise and customization
- ❖ Các năng lực
 - Mở rộng khả năng quản lý dự án và quan hệ khách hàng

❑ Chiến lược đổi mới mở

FLOSS ẢNH HƯỞNG ĐẾN CÁC MÔ HÌNH KINH DOANH NHƯ THẾ NÀO?

❑ Quan hệ khách hàng

- ❖ Chiến lược thông tin
 - Chuyển tới sự lãnh đạo
 - Chia sẻ hiểu biết và kinh nghiệm
- ❖ Thái độ và sự phục vụ
 - Thương hiệu chung tạo sự thuận tiện xây dựng hồ sơ năng lực
- ❖ Sự tin tưởng và lòng trung thành
 - Truy cập sự tinh thông của người viết phần mềm để cải thiện sự tin tưởng của khách hàng

LÀM THẾ NÀO ĐỂ GẮN FLOSS VÀO MÔ HÌNH KINH DOANH

❑ Các vấn đề tài chính

- ❖ Mô hình thu nhập
 - Tăng quy mô kinh doanh bằng cách tạo toàn bộ sản phẩm
- ❖ Cấu trúc chi phí
 - Chia sẻ chi phí giữa các thành viên trong mạng lưới
- ❖ Được / Mất
 - Lợi nhuận lớn hơn là kết quả của mô hình thu nhập và cấu trúc chi phí

NHỮNG MÔ HÌNH KINH DOANH DỰA TRÊN FLOSS

❑ Thông điệp chính:

- Không có mô hình kinh doanh đơn lẻ cho FLOSS
- Nhiều mô hình kinh doanh có thể dựa trên FLOSS hoặc liên quan 1 phần tới FLOSS
- Chìa khóa là không chọn giữa sở hữu và FLOSS, nhưng luôn đánh giá FLOSS như 1 tùy chọn trong môi trường kinh doanh

NHỮNG MÔ HÌNH KINH DOANH DỰA TRÊN FLOSS

❑ Thông điệp chính:

- Những mô hình kinh doanh đi từ độc quyền tới phần mềm tự do
- Hiện có nhiều mô hình kinh doanh như là các công ty
- Bạn có thể tạo cho riêng mình: Đổi mới cũng xảy ra trong mô hình kinh doanh

PHÂN LOẠI NGƯỜI THAM GIA FLOSS ?

- ❑ **Người chơi thuần FLOSS**
 - ❖ Các công ty phần mềm
 - ❖ FLOSS là trung tâm trong mô hình kinh doanh của họ
- ❑ **Cơ hội đến từ những chiến lược FLOSS**
 - ❖ Các công ty phần mềm và dịch vụ
 - ❖ Có thể sử dụng FLOSS khi thích hợp
- ❑ **FLOSS dựa trên đổi mới mở của**
 - ❖ Các nền công nghiệp: phần cứng, phần mềm, ...
 - ❖ Phần mềm mạnh
 - ❖ Đòn bẩy FLOSS trong chiến lược đổi mới

MÔ HÌNH KINH DOANH NGƯỜI CHƠI THUẦN FLOSS

- ❑ **Dịch vụ**
- ❑ **Phân phối giá trị gia tăng (« distro »)**
- ❑ **Cấp phép kép**
- ❑ **Kết hợp giữa nghiên cứu và phát triển (R&D)**

CƠ HỘI ĐẾN TỪ NHỮNG CHIẾN LƯỢC FLOSS?

- ☐ Những cơ hội bên ngoài
- ☐ VARs và những công ty dịch vụ không được biết trước
- ☐ (Sự quảng cáo)
- ☐ FLOSS dựa trên đổi mới mở sẽ được đề cập riêng
- ☐ Đòn bẩy FLOSS trong chiến lược đổi mới

NGƯỜI CHƠI THUẦN FLOSS NHỮNG CÔNG TY DỊCH VỤ (1/4)

- ☐ Được quảng bá bởi Eric Raymond
- ☐ Đào tạo, hỗ trợ, duy trì và biến đổi
- ☐ Các tiêu điểm của FLOSS
 - ❖ Chiến lược kinh doanh chuyên biệt
 - ❖ Các kỹ năng
 - ❖ Đóng góp cho cộng đồng

NGƯỜI CHƠI THUẦN FLOSS CÁC CÔNG TY DỊCH VỤ (2/4)

- ☐ **Các loại dịch vụ gì? Cơ bản, bất kỳ...**
 - ❖ Sự đào tạo
 - ❖ Sự biến đổi
 - ❖ Hỗ trợ kỹ thuật
 - ❖ Giám sát, cập nhật
 - ❖ Kiểm tra, kiểm chuẩn, chứng nhận, đảm bảo
- ☐ **Cung cấp 1 lần...**
- ☐ **Hoặc đã được bán như dịch vụ bổ sung (thu nhập định kỳ)**

NGƯỜI CHƠI THUẦN FLOSS CÁC CÔNG TY DỊCH VỤ (3/4)

- ☐ **Nỗ lực hạn chế R & D**
 - ❖ Như trong bất kỳ công ty dịch vụ nào
 - ❖ Nhưng: những nỗ lực đáng kể cần thiết để cắt giảm khó khăn và duy trì lực đòn bẩy của hiểu biết như 1 lợi thế cạnh tranh
- ☐ **Những cơ hội lớn cho phát triển kinh tế địa phương**

NGƯỜI CHƠI THUẦN FLOSS CÁC CÔNG TY DỊCH VỤ (4/4)

- ☐ **1 biến thể: lưu thông tiền tệ gián tiếp**
 - ❖ Các công ty đầu tư nghiên cứu và phát triển trên sự phát triển FLOSS
 - ❖ Nhưng lợi nhuận dựa trên phân phối các dịch vụ chuyên nghiệp
 - ❖ Hiểu biết tốt ở đây có thể đẩy bạn tiến xa

NGƯỜI CHƠI THUẦN FLOSS

PHÂN PHỐI GIÁ TRỊ GIA TĂNG (1/2)

- ❑ **Bắt nguồn từ thời kỳ phát triển của Linux trong quá khứ khi việc lắp ghép các hệ thống của bạn gặp khó khăn**
- ❑ **Xuất phát từ mô hình kinh doanh của công ty dịch vụ**
 - ❖ Dịch vụ = sự lựa chọn, đảm bảo, tích hợp, ổn định, và phân phối các gói FLOSS
 - ❖ Thu nhập = phân phối 1 lần duy nhất và/hoặc thêm các dịch vụ bổ sung (thường là cập nhật)

NGƯỜI CHƠI THUẦN FLOSS

PHÂN PHỐI GIÁ TRỊ GIA TĂNG (2/2)

- ❑ **Công ty hành động như công ty phần mềm, với nghiên cứu phát triển bên trong được dành riêng**
- ❑ **Nhưng không thu phí giấy phép phần mềm**
- ❑ **Tiếp thị dựa trên gói chuẩn**
 - ❖ Thường một « bản cộng đồng », FLOSS, miễn phí, không có dịch vụ thêm vào
 - ❖ Các phiên bản khác nhau (đồng/bạc/bạch kim; doanh nghiệp; ...) Hợp đồng dịch vụ rõ ràng
 - ❖ Truy cập các dịch vụ thêm vào có thể là điều kiện cho các thuê bao của một vài gói

NGƯỜI CHƠI THUẦN FLOSS

GIẤY PHÉP KÉP (1/3)

- ❑ **Một công ty phần mềm phát triển vỏ bọc thương mại**
 - ❖ phân phối trong FLOSS dưới một giấy phép copylefted
 - ❖ Cũng có thể thanh toán dưới giấy phép độc quyền
- ❑ **Phiên bản FLOSS phổ biến tới:**
 - ❖ thu được sự hiểu biết cao trong cộng đồng
 - ❖ cùng lúc, là một sự ngăn cản với:
 - LORGs không thoải mái với giấy phép copylefted
 - Những người bán lại OEM
 - ❖ ... vì họ thích hơn việc trả và nhận các điều kiện giấy phép ko có tính copylefted

NGƯỜI CHƠI THUẦN FLOSS

GIẤY PHÉP KÉP (2/3)

- ❑ **Giấy phép kép thỉnh thoảng bị chỉ trích như một vụ trộm các nguyên lý phần mềm tự do nhằm đưa vào giấy phép độc quyền**
- ❑ **Sự quản lý cộng đồng đúng đắn là cần thiết**
 - ❖ Nhận các phân phối thậm chí thông qua công ty đứng đằng sau dự án có thể thu lợi nhuận trên các phân phối: sự khuyến khích nhà phân phối là cần thiết
 - ❖ Để các công ty thu được quyền lợi trên tất cả các mã nguồn (vì vậy là hợp pháp để thực hành giấy phép kép): gán những bản quyền cần thiết

NGƯỜI CHƠI THUẦN FLOSS

GIẤY PHÉP KÉP (3/3)

- ☐ Trong một biên thể, công ty phân phối một bản « nhẹ hơn » trong FLOSS...
- ☐ Và một bổ sung với dự án FLOSS
- ☐ Hoặc một đầu cuối cao hơn, bản cho « doanh nghiệp », với nhiều đặc điểm, dưới giấy phép độc quyền
 - ❖ Yêu cầu gắn một bản quyền...
 - ❖ Hoặc (không giống chuẩn giấy phép kép) một giấy phép cho phép/hàn lâm cho dự án FLOSS

NGƯỜI CHƠI THUẦN FLOSS

NGHIÊN CỨU PHÁT TRIỂN CHUNG (1/3)

- ☐ Có thể là một mô hình tạo ra lợi nhuận...
- ☐ Nhưng cũng thường là trung tâm trong thể hệ thứ ba, các tổ chức FLOSS phi lợi nhuận (ObjectWeb, Eclipse, MMBase, ...)
- ☐ Trung tâm trên một FLOSS (hoặc một phần của) các dự án FLOSS
- ☐ Các khách hàng trả cho sự nâng cao, mà thậm chí đóng góp vào dự án...
- ☐ Có thể sau một sự chậm trễ (mô hình tiên chuộc)

NGƯỜI CHƠI THUẦN FLOSS

NGHIÊN CỨU PHÁT TRIỂN CHUNG (2/3)

- ❑ **Mô hình này là phù hợp khi:**
 - ❖ Dự án là phức tạp và phòng chống với nhiều mở rộng
 - ❖ Đó là thị trường riêng, nơi một vị trí thống trị có thể nhanh chóng quickly giành được và duy trì
 - ❖ Công nghệ có thể lạc hậu nhanh chóng; vì vậy các đặc điểm vượt trội khác là một ưu điểm thực sự

NGƯỜI CHƠI THUẦN FLOSS

NGHIÊN CỨU PHÁT TRIỂN CHUNG (3/3)

- ❑ **Khách hàng phải cảm thấy thoải mái trong chia sẻ với họ**
« ngang hàng » - như vậy một cộng đồng thực sự được tạo dựng
- ❑ **Dễ làm hơn cả với công nghệ quan trọng, nhưng không khác biệt**
(v.d. an ninh middleware)
- ❑ **Các điểm tương đồng với « chia sẻ nghiên cứu phát triển » mô hình sử dụng**

NGƯỜI CHƠI THUẦN FLOSS

CÁC NHÂN TỐ THÀNH CÔNG THEN CHỐT (1/2)

- ❑ **Thị trường: tùy thuộc vào cửa sổ cơ hội**
- ❑ **Vươn tới số lượng lớn trong cộng đồng những người phát triển**
 - ❖ Những người phát triển hành động người dùng dẫn dắt và như những người thích nghi sớm
- ❑ **Hạ bệ những người ra quyết định ngò vức**
- ❑ **Efficient commercial infrastructure**

NGƯỜI CHƠI THUẦN FLOSS

CÁC NHÂN TỐ THÀNH CÔNG THEN CHỐT (2/2)

- ❑ **Sự quan trọng của hệ sinh thái kinh doanh**
 - ❖ Từ khách hàng/nhà cung cấp liên quan đến đối tác partnership với khách hàng – và các đối tác khác
 - ❖ Hiệu ứng mạng lưới mạnh trong thích ứng công nghệ
 - ❖ Không thể làm đơn độc trong FLOSS - vì FLOSS là chia sẻ và cộng tác
 - ❖ Thông tin qua các cổng tổ chức: đúng cho gợi ý kỹ thuật, đúng cho danh tiếng
 - ❖ Một người chơi FLOSS thành công thường là nhân tố then chốt của hệ sinh thái

CÁC CHIẾN LƯỢC FLOSS CÓ CƠ HỘI MỞ RỘNG VIỆC ĐẦU TƯ ĐƯỢC TÀI TRỢ (1/3)

- ❑ **Phát triển FLOSS được tài trợ bởi một vài tổ chức bên ngoài**
 - ❖ Quỹ công cộng
 - ❖ Các viện nghiên cứu hỗ trợ sự đổi mới
- ❑ **Thường thấy trong hàn lâm, trong quản trị công cộng, hoặc phần tư nhận khi chính sách công cộng nhằm phát triển FLOSS**
- ❑ **Trong kết nối với chuẩn mở: phát triển một mô hình tham chiếu được triển khai**

CÁC CHIẾN LƯỢC FLOSS CÓ CƠ HỘI MỞ RỘNG VIỆC ĐẦU TƯ ĐƯỢC TÀI TRỢ (2/3)

- ❑ **Chiến lược « mất người dẫn dắt »**
 - ❖ Một công ty đầu tư trong FLOSS để phổ biến công nghệ
 - ❖ Không tìm kiếm bất cứ phản hồi trực tiếp
 - ❖ Như là cạnh tranh giá rẻ hơn và phát triển thị trường bổ sung

CÁC CHIẾN LƯỢC FLOSS CÓ CƠ HỘI MỞ RỘNG VIỆC ĐẦU TƯ ĐƯỢC TÀI TRỢ (3/3)

☐ Chiến lược « mất người dẫn dắt» (tiếp)

- ❖ Không thực sự mở rộng tạo quỹ: trong một công ty đơn lẻ, đầu tư được làm trong FLOSS (trung tâm giá thuần) để tạo ra thị trường với chào hàng khác
- ❖ Dễ hơn nên công ty là ở vị trí vượt trội trên thị trường cũng như, nếu không rủi ro đem lại kết quả ngoài mong đợi là có thực
- ❖ Chuẩn sử dụng là những màn hình khói thuốc, như là các cộng đồng nguồn mở

CÁC CHIẾN LƯỢC FLOSS CÓ CƠ HỘI

Dịch vụ cho người không có chuyên môn và bán lại các giá trị gia tăng

☐ Cơ bản « doanh nghiệp như thường lệ »

- ❖ những nhà tích hợp
- ❖ các dịch vụ chuyên nghiệp
- ❖ bán lại các giá trị gia tăng

☐ Trả lời các yêu cầu khách hàng không có phát triển một việc trí đặc biệt trên FLOSS

TÓM TẮT

- ❑ **Không có một mô hình kinh doanh FLOSS riêng lẻ, có nhiều, bạn có thể tự phát minh**
- ❑ **Các mô hình kinh doanh tùy thuộc cửa sổ cơ hội. Chúng sẽ phát triển cho phù hợp với hoàn cảnh**
- ❑ **Các mô hình kinh doanh người chơi thuần và các chiến lược có cơ hội được tiếp cận khác nhau**
- ❑ **Giấy phép và mô hình kinh doanh là một cặp ràng buộc**

FLOSS VÀ SỰ ĐỔI MỚI

SOẠN BỞI: FRANÇOIS LETELLIER

CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT

PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 16 /2008/06/30

BÂY GIỜ CHÚNG TA SẼ HỌC...

- ☐ **Sử dụng FLOSS như thế nào trong một chiến lược đổi mới**
- ☐ **Các nguyên lý của “đổi mới mở”**
- ☐ **Thêm các mô hình kinh doanh, dùng FLOSS**
- ☐ **Và những gợi ý xây dựng mô hình một công ty theo chiến lược đổi mới mở**

TẠI SAO ĐỔI MỚI?

- ❑ **“Đổi mới là việc hướng tới sự tăng trưởng kinh tế, năng xuất, tạo việc làm và nâng cao chất lượng cuộc sống.**
- ❑ **Đổi mới cũng khuyến khích tính cạnh tranh ICT; tiếp đến, cạnh tranh dẫn đến sản phẩm tốt hơn, cải thiện lựa chọn người tiêu dùng và lý tưởng, số lượng người chấp nhận ICT lớn hơn nhiều”**

Source: European Task-Force on ICT Sector competitiveness & ICT uptake, WG on innovation in R&D, manufacturing and services http://ec.europa.eu/enterprise/ict/policy/taskforce/wg/wg3_report.pdf

500 DỰ ÁN HÀNG ĐẦU TRONG SOURCEFORGE

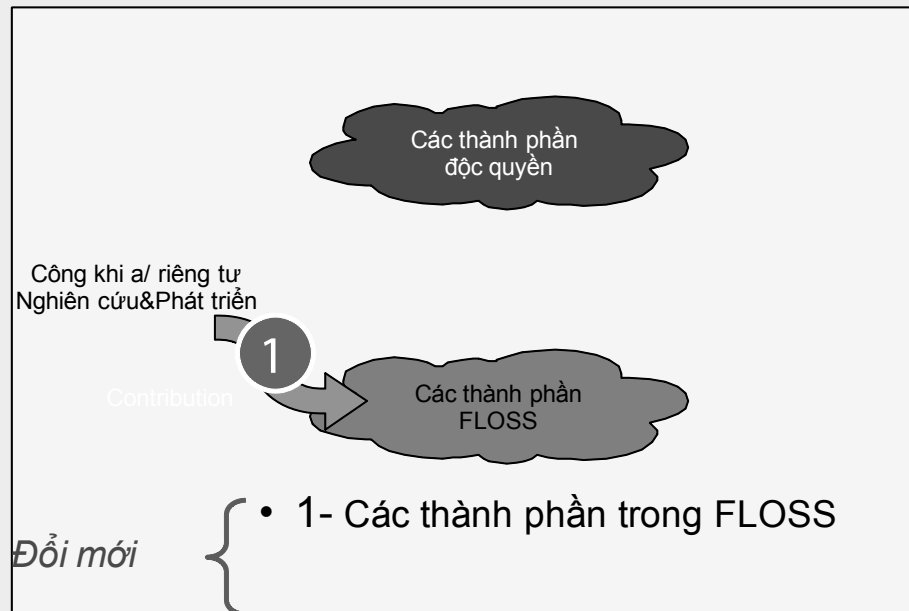
	Công nghệ mới	N. cho một nền tảng	Công nghệ đang có
Thị trường mới	1% Đổi mới căn bản		<1%
Thị trường đang có	<1%	10%	87% không đổi mới

Nguồn: Tính đổi mới của các dự án phần mềm nguồn mở, K. Klineciewicz 2005

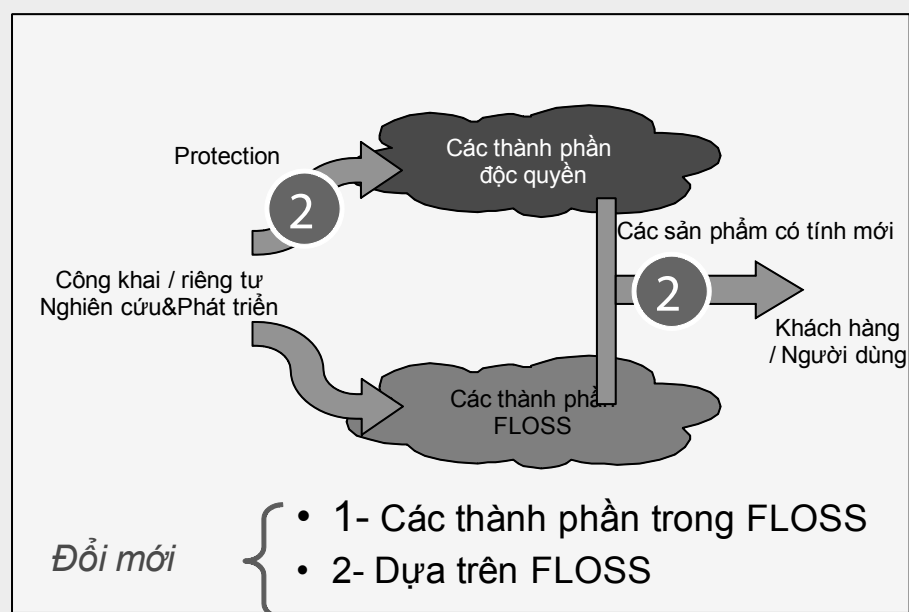
ĐỔI MỚI DIỄN RA NHƯ THẾ NÀO TRONG FLOSS?

- ❑ **Đổi mới đóng góp vào một dự án FLOSS**
- ❑ **Sự kích hoạt đổi mới trong FLOSS đang diễn ra ở “nơi khác”**
- ❑ **FLOSS như là một hình mẫu phát triển đổi mới – “đổi mới mở”**

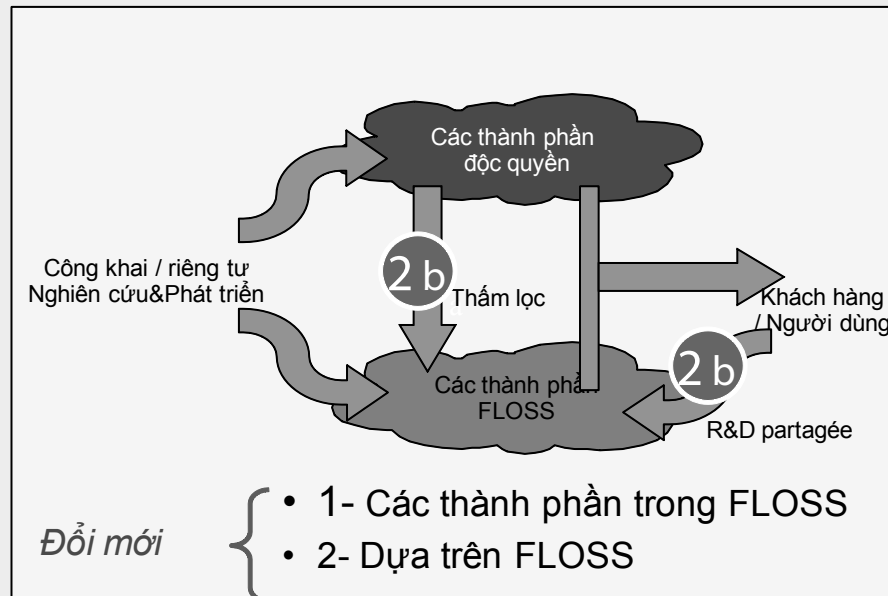
ĐỔI MỚI TRONG FLOSS



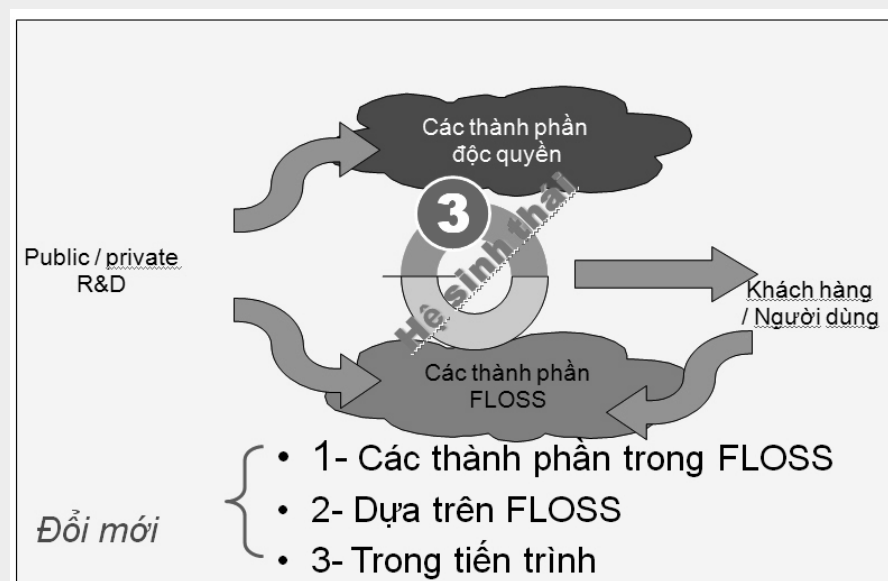
ĐỔI MỚI DỰA TRÊN FLOSS



ĐỔI MỚI ĐÓNG GÓP TRỞ LẠI FLOSS



QUÁ TRÌNH TẠO GIÁ TRỊ ĐỔI MỚI



MÔ HÌNH KHÔNG CÓ ĐỔI MỚI PHẦN MỀM BỊ BỎ RƠI

- ❑ **Thực tế tồi tệ nhất: chúng ta không biết phải làm gì với dự án này ... hãy mở mã nguồn nó ra!**
 - ❖ Lỗi thời
 - ❖ Vô ích
 - ❖ Không có thị trường
 - ❖ Không có cộng đồng
- ❑ **Đầu vào rác, đầu ra rác!**

CÁC MÔ HÌNH KINH DOANH FOSS XÀI CHÙA HAY OEM

- ❑ **FLOSS phân phối dưới giấy phép mang tính chấp nhận được...**
- ❑ **Từng xây dựng các sản phẩm độc quyền**
- ❑ **Những người xài chùa / OEM là các đồng minh của các công ty hoạt động theo chiến lược LOSS**
- ❑ **Tính phi pháp được thực hiện bởi phần mềm sử dụng giấy phép copylefted**
 - ❖ Dẫn tới mô hình kinh doanh giấy phép song hành “dual-license”

CÁC MÔ HÌNH KINH DOANH FLOSS - NHÚNG

- ☐ **Bán phần cứng**
- ☐ **Nhúng phần mềm / firmware dựa trên FLOSS**
- ☐ **Việc nghiên cứu và phát triển chủ yếu nhằm tới việc thiết kế các sản phẩm cuối cùng, hơn là tới phần mềm**

CÁC MÔ HÌNH KINH DOANH FLOSS INFOWARE/SAAS

- ☐ **Dịch vụ trực tuyến**
- ☐ **Dựa trên FLOSS + sự cải thiện bên trong tổ chức**
- ☐ **Truy cập vào mã nguồn cho việc tùy biến, và giảm chi phí giấy phép đóng góp vào sự linh hoạt của công ty**
- ☐ **Một số giấy phép có tính copyleft mạnh (VD. Affero GPL) trong trường hợp này yêu cầu rằng mã nguồn được thực hiện sẵn cho người dùng**

FLOSS: SỰ ĐỔI MỚI MỞ TRONG PHẦN MỀM

- ☐ Bởi vì sớm hay muộn, nhà cung cấp phần mềm độc quyền phải đối mặt với việc cạnh tranh của hầu hết những phần mềm miễn phí thay thế
- ☐ Bởi vì chi phí sản xuất phần mềm có thể được giảm thông qua kỹ thuật cộng tác
- ☐ Bởi vì việc mở mã nguồn là cách tốt nhất để tối đa hóa tiềm năng hợp tác
- ☐ Và bởi vì khi sự cạnh tranh trở nên lạc lõng, sự cộng tác là một cách tốt để duy trì đổi mới

TỪ SỰ ĐỘC QUYỀN ĐẾN FLOSS SỐC VĂN HÓA

- ☐ Hội chứng NIH
- ☐ Hội chứng NSH
- ☐ Chủ nghĩa cầu toàn: Không đủ để là FLOSS
- ☐ Những nhà phát triển phải tìm hiểu gì:
 - ❖ Sử dụng lại trước khi phát minh lại cái bánh xe
 - ❖ Đóng góp các bản cải tiến và vá lỗi
 - ❖ Trao đổi ngang hàng
- ☐ Chiến lược tự vẫn
 - ❖ Phát minh lại, phát triển lại mọi thứ, coi nhẹ các tiêu chuẩn và khách hàng

BẢO TỒN CÁC LỢI NHUẬN HẤP DẪN

- ☐ Phần mềm được phát triển dưới dạng các mô-đun
- ☐ Bất cứ lúc nào, các phần giá trị được phổ dụng hóa, những phần khác thì không
- ☐ Khi một phần mềm trưởng thành, một số thành phần sẽ “được hàng hóa hóa”
- ☐ Cơ hội cho những giá trị hấp dẫn đi ra thị trường

CÁC KIẾN NGHỊ FLOSS DỰA TRÊN ĐỔI MỚI MỞ

- ☐ FLOSS cung cấp một hành lang pháp lý “turnkey”
- ☐ Bám vào điều kiện thuận lợi, giảm thiểu rủi ro trong việc thương lượng với nhà cung cấp và nhà thầu phụ
- ☐ Luôn đảm bảo bạn có thể tìm được cái thay thế cho bất kỳ công nghệ chính nào bạn sử dụng
- ☐ Đánh giá rủi ro và hậu quả của việc thay đổi theo mô hình kinh doanh của bạn
- ☐ Các công cụ có sẵn trong FLOSS – hãy sử dụng nó
- ☐ Được tham gia vào cộng đồng, tuân thủ các quy tắc và nhận biết các giới hạn của nó

TÓM TẮT

- ❑ **Sự đổi mới diễn ra**
 - ❖ Trong các dự án FLOSS
 - ❖ Trên các sản phẩm dựa trên FLOSS
 - ❖ Thông qua “sự sàng lọc”
 - ❖ Và trong quá trình tự thân: đổi mới mở
- ❑ **Hàng hóa được liên kết chặt chẽ với sự đổi mới thông qua việc bảo tồn những lợi nhuận hấp dẫn**
- ❑ **Chuẩn bị với sốc văn hóa khi thay đổi từ độc quyền sang FLOSS**

NHỮNG RÀO CẢN VỚI SỰ THÍCH NGHI NGUỒN MỞ

SOẠN BỞI: FRANÇOIS LETELLIER

CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT

PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 23 /2008/06/30

XÁC ĐỊNH NHỮNG RÀO CẢN ĐỂ THÍCH NGHI FLOSS

BÂY GIỜ CHÚNG TA SẼ HỌC...

- ☐ **Những động cơ thúc đẩy thích nghi theo FLOSS trong doanh nghiệp**
- ☐ **Những rào cản trong việc thích nghi FLOSS**
- ☐ **Những cách để giải quyết chúng**
- ☐ **Cách chúng mang đến những cơ hội kinh doanh như thế nào**

SỬ DỤNG FLOSS TRONG XU HƯỚNG CHỦ ĐẠO

❑ 80% P.A. và LOGs dùng FLOSS

- ❖ 30% có thể không hiểu biết về việc
- ❖ 78% Linux
- ❖ 74% cơ sở hạ tầng / máy chủ
- ❖ 61% công cụ phát triển
- ❖ 45% các ứng dụng desktop
- ❖ 29% các ứng dụng doanh nghiệp (groupware, CRM, ERP...)

❑ Công cụ và sáng kiến trong điều tra về nguồn mở

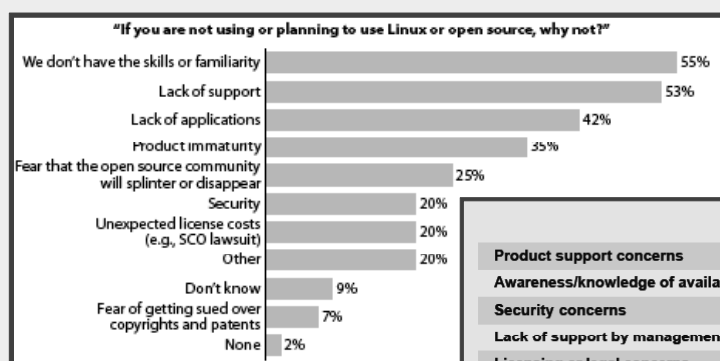
Source: CIO.com APR 2008



Firefox	82.41%
Xerces	76.99%
Zlib	75.98%
Xalan	73.77%
Prototype	59.21%
PERL	57.63%
OpenSSL	57.57%
OpenOffice	55.61%
Ghostsript	53.53%
Wget	48.74%
Gcc	47.6%

(June 08)

NHỮNG RÀO CẢN ĐỂ THÍCH NGHI THEO FLOSS 2004-2008

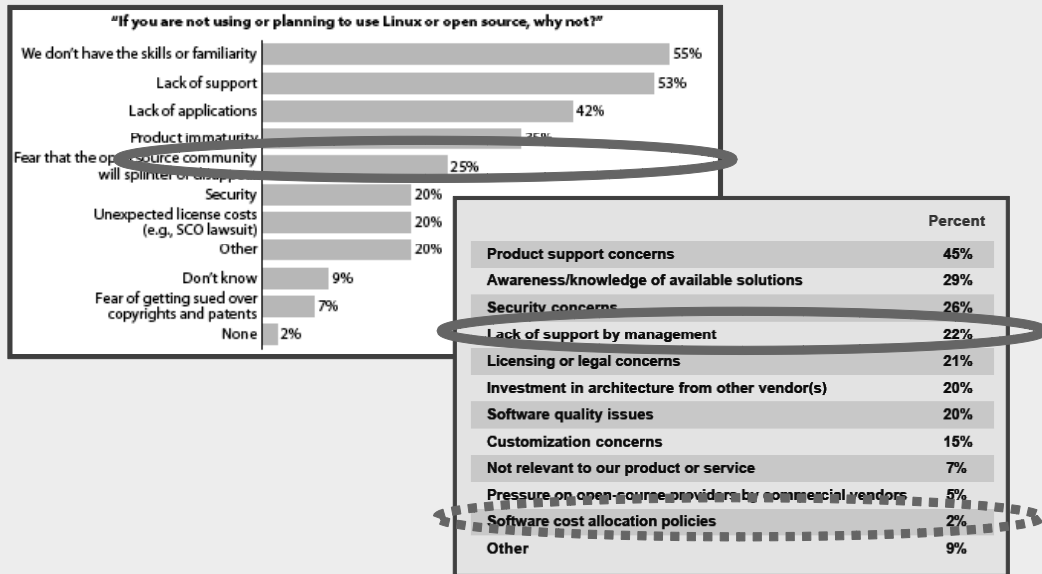


Source: « the Costs and Risks of Open

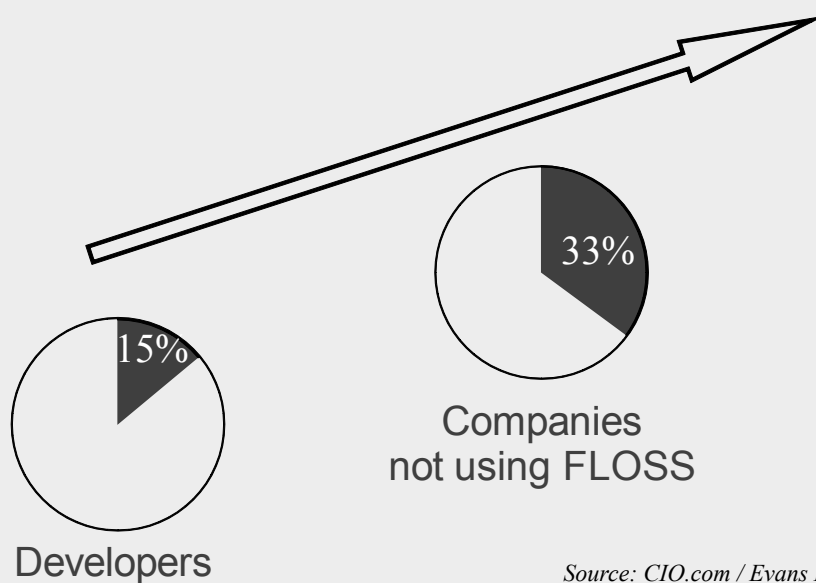
Source », Forrester, 2004

	Percent
Product support concerns	45%
Awareness/knowledge of available solutions	29%
Security concerns	26%
Lack of support by management	22%
Licensing or legal concerns	21%
Investment in architecture from other vendor(s)	20%
Software quality issues	20%
Customization concerns	15%
Not relevant to our product or service	7%
Pressure on open-source providers by commercial vendors	5%
Software cost allocation policies	2%
Other	9%

#1 RÀO CẢN KHẢ NĂNG HỖ TRỢ

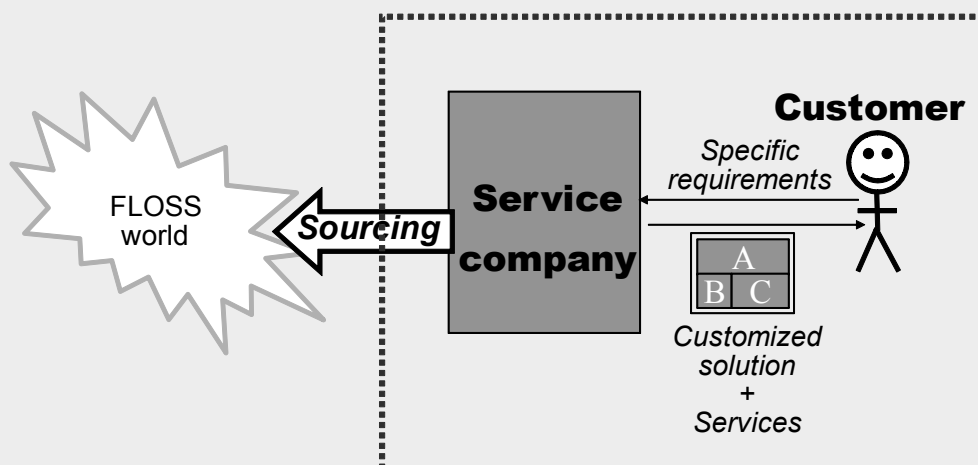


RÀO CẢN THỨ 1 KHẢ NĂNG SẴN SÀNG HỖ TRỢ



Source: CIO.com / Evans Data - 2008

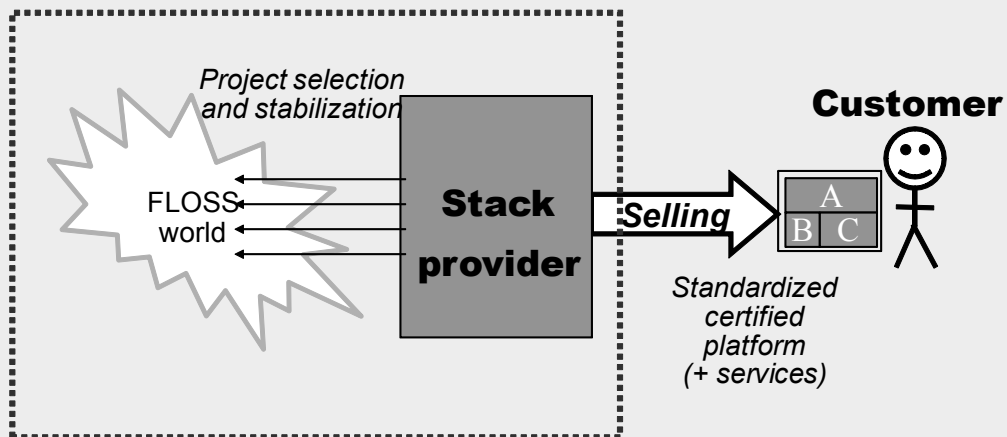
PHÁ VỠ RÀO CẢN ĐỂ ÁP DỤNG FLOSS CÔNG NGHỆ



CÁC CÔNG TY HỖ TRỢ

- ☐ **Loại dịch vụ? cơ bản, bất kỳ...**
 - ❖ Đào tạo
 - ❖ Tùy biến
 - ❖ Bảo trì và hỗ trợ kỹ thuật
 - ❖ Giám sát, cập nhật
 - ❖ Kiểm tra, chuẩn hóa, cấp chứng chỉ, bảo hành
- ☐ **Bán hàng trả một lần hoặc trả góp**
- ☐ **Những thứ cần thiết cho các dịch vụ là cơ chế kinh doanh và cơ chế phát triển kinh tế địa phương**

PHÁ VỠ RÀO CẢN ĐỂ ÁP DỤNG FLOSS CÔNG NGHỆ

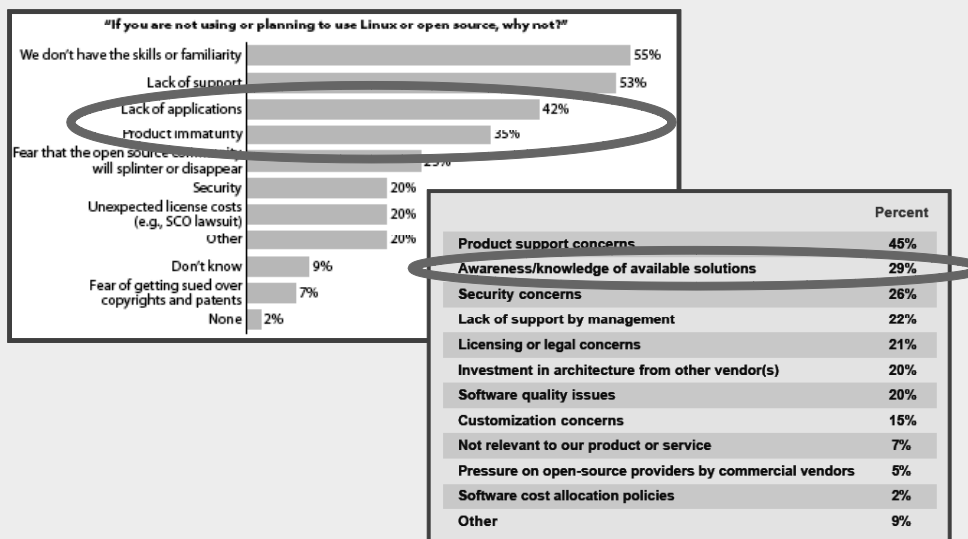


GIÁ TRỊ GIA TĂNG CỦA FLOSS

- ❑ **Những công ty trong lĩnh vực phần mềm đóng góp cho sự nghiên cứu và phát triển trên thế giới**
- ❑ **Mà không tính phí bản quyền phần mềm**
- ❑ **Tiếp thị dựa trên các chuẩn đóng gói**
 - ❖ Thông thường, một "phiên bản cộng đồng", FLOSS, Free, kèm với những giá trị gia tăng
 - ❖ Nhiều phiên bản (bronze/gold/platinum; enterprise;...) với thỏa thuận mức độ dịch vụ rõ ràng
 - ❖ Truy cập vào các dịch vụ gia tăng có thể liên quan tới các hợp đồng dài hạn của một số gói dịch vụ

RÀO CẢN THỨ 2

KHẢ NĂNG SẴN SÀNG ÁP DỤNG



THIẾU CÁC ỨNG DỤNG FLOSS LÀ MỘT CƠ HỘI KINH DOANH

❑ Vấn đề chính: Làm thế nào để gây quỹ R&D?

❖ Góp tiền gián tiếp

- Công ty đầu tư vào R&D trên việc phát triển FLOSS (VD dự án FLOSS)
- Thu nhập được tạo ra nhờ việc cung cấp các dịch vụ chuyên nghiệp

THIỆU CÁC ỨNG DỤNG FLOSS LÀ MỘT CƠ HỘI KINH DOANH

☐ **Vấn đề chính: Làm thế nào để gây quỹ R&D?**

- ❖ Giấy phép song hành
 - Một công ty phần mềm phát triển COTS
 - Phân phối vào FLOSS dưới giấy phép copyleft
 - Cũng sẵn sàng trả tiền cho các giấy phép độc quyền

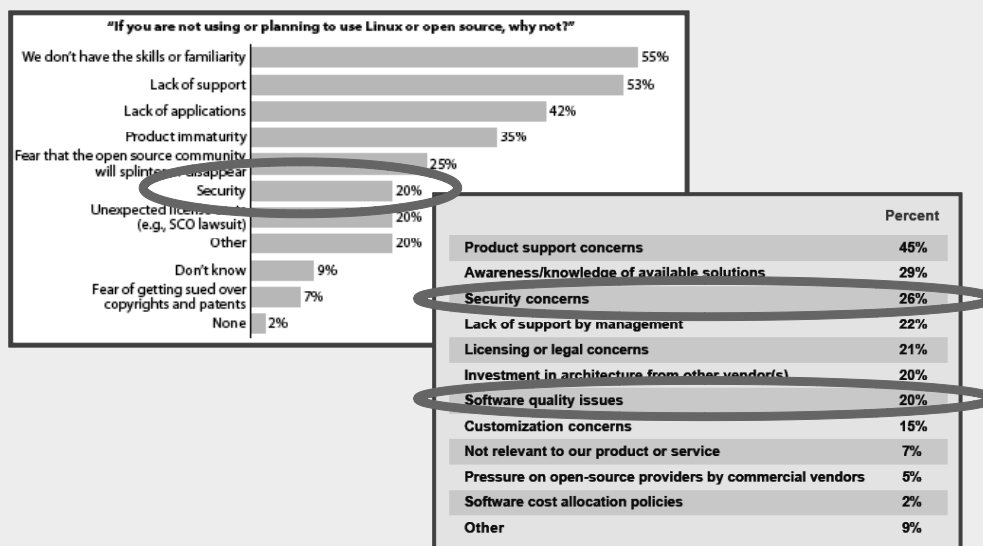
THIỆU CÁC ỨNG DỤNG FLOSS LÀ MỘT CƠ HỘI KINH DOANH

☐ **Vấn đề chính: Làm thế nào để gây quỹ R&D?**

- ❖ Cộng đồng hóa R&D
 - Công ty dẫn đầu một dự án FLOSS
 - Khách hàng trả tiền cho nâng cấp, cải tiến, mà thực ra là đóng góp cho dự án...

☐ **... Thêm vào ý tưởng của chính bạn...**

RÀO CẢN THỨ 3 CHẤT LƯỢNG & AN NINH



VẤN ĐỀ CHẤT LƯỢNG

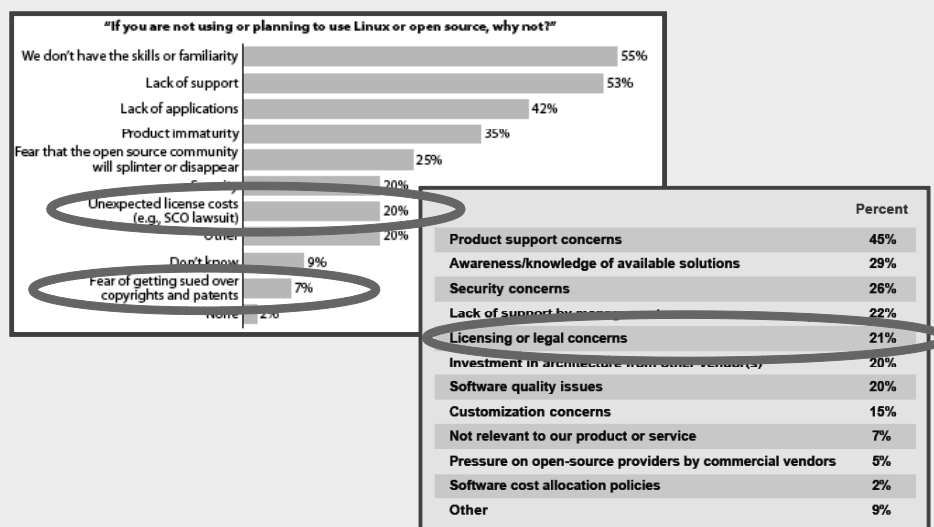
- ❑ **FLOSS có thể thực sự tốt**
 - ❖ Hiệu ứng "many eyeballs"
- ❑ **Nhưng không luôn là vậy**
 - ❖ Các dự án với cộng đồng nhỏ lẻ (hoặc chỉ do một công ty tài trợ) có thể chỉ đủ tốt hoặc tệ như phần mềm nguồn đóng

PHẦN MỀM FLOSS XÁC ĐỊNH VÀ LỰA CHỌN

❑ Phương pháp lựa chọn FLOSS hướng đến:

- ❖ Tìm các dự án mà đáp ứng được các nhu cầu của bạn
- ❖ So sánh chúng với sản phẩm độc quyền
- ❖ Lựa chọn một (vài) sản phẩm và loại bỏ những thứ khác
- ❖ Suy xét những báo cáo phân tích tiếp thị, thổi phồng, giả mạo một cách khách quan
- ❖ Thoải mái với quyết định và sẵn sàng báo cáo điều quản lý

RÀO CẢN THỨ 4 GIẤY PHÉP/HỢP PHÁP



VÍ DỤ SCO VS IBM

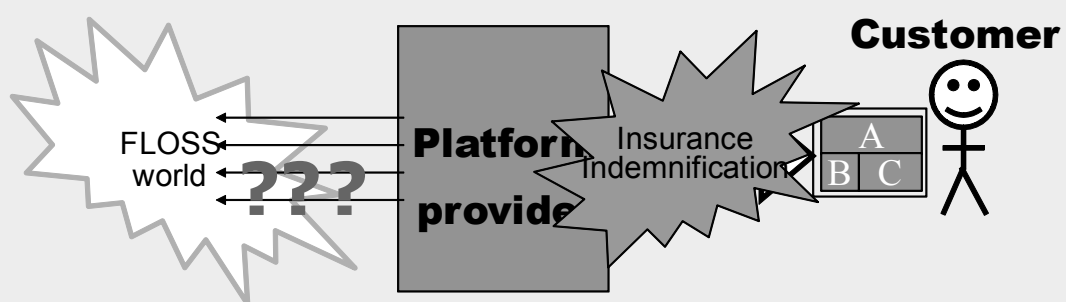
- 06/03/2003 - SCO kiện IBM (vi phạm sở hữu trí tuệ) – giá cổ phiếu SCO: tăng 40%
- 14/05/2003 SCO dừng bán các sản phẩm Linux products với lý do “sản phẩm bắt nguồn từ UNIX nhưng chưa hợp pháp”
- 14/05/2003 – SCO gửi thư đến 1,500 quản trị CNTT
- 19/05/2003 - Microsoft mua một giấy phép Unix từ SCO – trị giá \$ triệu USD
- 4/08/2003 – Red Hat kiện SCO (về giả mạo)
- 19/08/2003 – Mã “ăn trộm” của SCO được cho là bắt nguồn từ Unix 1979 (pub. Dom/BSD?)
- 24/09/2003 – HP đảm bảo bồi thường cho các khách hàng Linux
- 26/09/2003 – IBM kiện SCO về việc vi phạm GPL
- 17/10/2003 – VC “BayStar” (sân sau của MS) đầu tư 50 triệu USD vào SCO
- 31/05/2007 - CEO của SCO dự đoán doanh thu từ bán các “giấy phép Linux” có thể đạt hàng tỷ đô-la
- 10/08/2007 – Tòa tuyên rằng Novell là chủ sở hữu của bản quyền Unix – chấm dứt các nỗ lực của SCO
- 14/09/2007 - SCO đệ đơn xin phá sản theo điều chương 11 của Luật Phá sản Hoa kỳ
- 23/10/2007 - SCO đệ đơn đề nghị bán mảng kinh doanh UNIX cho York Capital Management với giá 36 triệu USD – Novell và IBM phản đối
- 12/11/2007 – SCO bị một tòa án Đức cáo buộc có tội do quảng bá thông tin sai lệch về Linux
- 23/02/2008 - SCO đệ trình kế hoạch tái cơ cấu lên Tòa án Phá sản Hoa Kỳ ở Delaware

Source: <http://www.linux.org/news/sco/timeline.html>

BỒI THƯỜNG VÀ BẢO HIỂM HỢP PHÁP

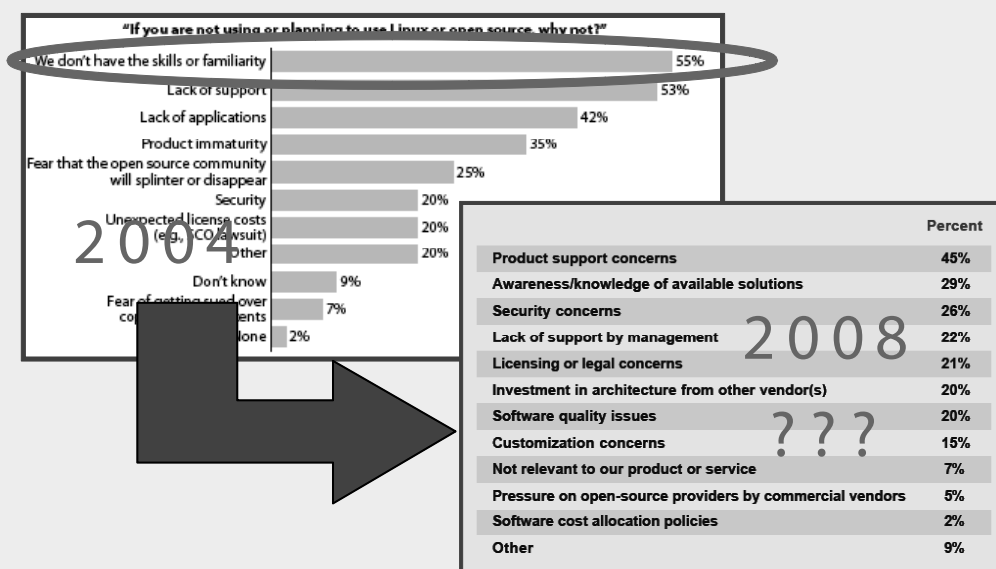
- ❑ Một số mã nguồn có thể vi phạm vào sự sở hữu trí tuệ của người khác (©, các bằng sáng chế, nhãn hiệu đăng kí) thì người sử dụng cũng dính dáng đến sự vi phạm này
- ❑ Khi mã nguồn được công bố thì những thông tin quyền sở hữu được ghi chú
- ❑ (Không như sản phẩm độc quyền) FLOSS đến từ rất nhiều nguồn (không xác định)
- ❑ Khe hẹp nào để len vào?

BỒI THƯỜNG VÀ BẢO HIỂM HỢP PHÁP

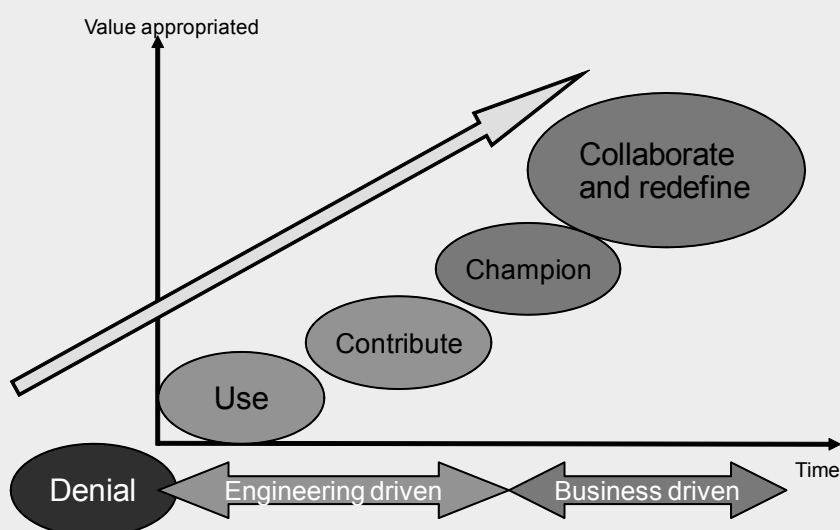


- ❖ Một công ty (VD: Red Hat, OpenLogic, JBoss, Novell...) có thể đang lách luật để kiếm lời
- ❖ Bồi thường khách hàng trong trường hợp tranh chấp liên quan đến các sản phẩm FLOSS của nhà cung cấp
- ❖ Do điều luật là không có bằng chứng, nên sự bồi thường có thể là vô ích

VIỆC TRIỆT TIÊU CÁC RÀO CẢN CÁC KỸ XẢO

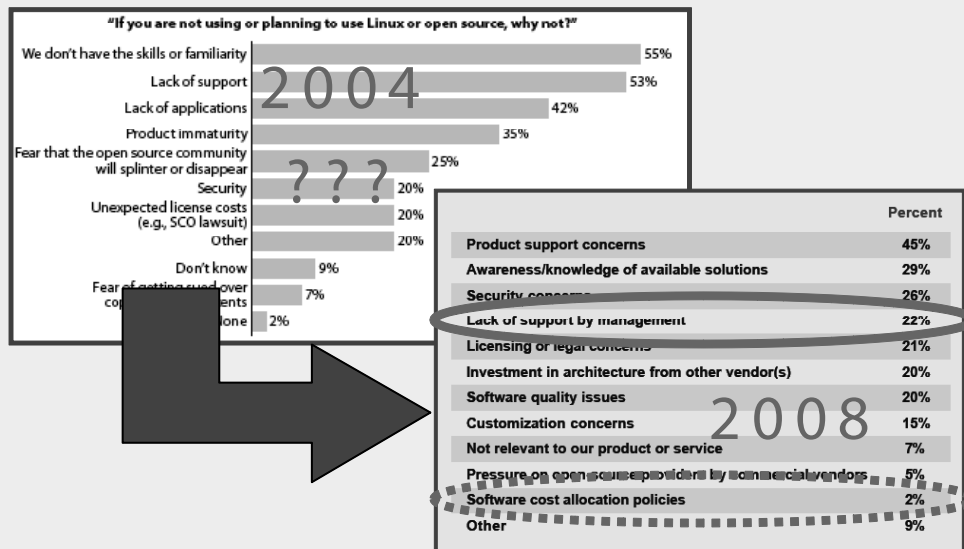


TIẾN DẦN LÊN CÁC "NẮC THANG" ỨNG DỤNG FLOSS



Source: FLOSSMetrics project

CÁC VẤN ĐỀ XUẤT HIỆN HỖ TRỢ TỪ LÃNH ĐẠO



TỔNG GIÁ TRỊ CỦA QUYỀN SỞ HỮU (TCO)

❑ Các vấn đề hiện tại...

- ❖ Không chỉ tính mỗi chi phí giấy phép
- ❖ Phải tính đến cả những chi phí tiềm ẩn khác...

❑ "Total Cost of Ownership" / "Total Economic Impact"

- ❖ Chi phí trực tiếp
 - Giá sản phẩm
 - Phụ phí (phát triển, bảo trì, hỗ trợ, ...)
- ❖ Chi phí gián tiếp (lên đến 60%?)
 - Thời gian gián đoạn, chất lượng dịch vụ...

❑ TCO/TEI không mang lợi nhuận

LINUX VÀ WINDOWS SỰ TRANH CÃI VỀ TCO ...

*Nghiên cứu 2002 của IDC (MS funded)

- Linux TCO > Win 2000 TCO cho hạ tầng
- Linux TCO < Win 2000 TCO cho dịch vụ web

Nghiên cứu 2002 Robert Frances Group (IBM funded)

- Linux TCO < 1/2 Windows TCO

Nghiên cứu 2003 Forrester Giga Group (MS funded)

- Win/.NET TEI = 4/5 Linux/J2EE TEI for LORGS

Nghiên cứu 2004 Cybersource (độc lập)

- Linux TCO = 2/3 Windows TCO

Nghiên cứu 2004 Yankee Group (độc lập)

- Linux TCO < Windows TCO cho các công ty mới, nhỏ
- Windows TCO < Linux TCO cho các "Windows shops"

Nghiên cứu 2005 Meta Group/Gartner

- Chi phí một năm Linux > tiết kiệm ban đầu Windows

KẾT QUẢ:

Phân tích độc lập?

Nắm đầy đủ thực tế

Kết quả phụ thuộc lớn vào mô hình

Một thế giới thay đổi ...

NHỮNG "E NGẠI" CHƯA ĐƯỢC NÓI ...

- ☐ Xét từ phía CIO: Cắt giảm ngân sách tức là làm giảm ảnh hưởng của tôi trong công ty
- ☐ Xét từ phía người sử dụng: Phần mềm miễn phí là phần mềm kém chất lượng. Đó là tín hiệu cho rằng công việc của tôi là không mấy quan trọng

FLOSS TRONG QUÁ TRÌNH TIẾP NHẬN

- ☐ **44% các công ty cho rằng các giải pháp của FLOSS & có bản quyền có thể được tiếp cận như nhau**
- ☐ **62% quản trị IT cho rằng Linux đủ tin cậy để dùng cho các ứng dụng quan trọng**
- ☐ **59% các phòng IT cho rằng FLOSS TCO không bằng các phần mềm có bản quyền**
- ☐ **32% quản trị IT cho rằng lý do chính để chọn FLOSS là vì nó mềm dẻo**

MỤC TIÊU TRONG TƯƠNG LAI

- ☐ **2011 – Mã nguồn mở sẽ chiếm ưu thế trong kiến trúc phần mềm cho mạng lưới nhà cung cấp**
- ☐ **2012 - 90% doanh nghiệp sẽ sử dụng mã nguồn mở hoặc trực tiếp, hoặc gián tiếp**
- ☐ **2012 - SaaS sẽ “làm lu mờ” mã nguồn mở như một phương pháp cắt chi phí của các doanh nghiệp IT**
- ☐ **2013 - Nhân tố chính để triển khai Linux sẽ không phải là lợi thế về TCO so với các hệ điều hành khác**

Source: Gartner Group, « the State of Open Source 2008 »

TỔNG KẾT

- ☐ **Hiện tại FLOSS là xu thế chủ đạo**
- ☐ **Rào cản 1: mỗi quan tâm là tình trạng hiện có khi hỗ trợ**
- ☐ **Thiếu hỗ trợ: Các ứng dụng là cơ hội kinh doanh**
- ☐ **TCO và SCO**
- ☐ **Các phương pháp đánh giá phần mềm và chất lượng**
- ☐ **Hỗ trợ từ cấp quản lý và các chính sách FLOSS của doanh nghiệp**

GIỚI THIỆU CÁC TIÊU CHÍ LỰA CHỌN PHẦN MỀM NGUỒN MỞ

SOẠN BỞI: FRANÇOIS LETELLIER

CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT

PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 27 /2008/06/30

Đánh giá các dự án mã nguồn mở

CHÚNG TA SẼ TÌM HIỂU ...

- ☐ **Tiêu chí lựa chọn các dự án nguồn mở**
- ☐ **Tổng quan về một số phương pháp đánh giá**

SỰ KHÁC NHAU GIỮA NGUỒN MỞ VÀ SẢN PHẨM THƯƠNG MẠI

	Thương mại	Mã mở
Nhà cung cấp	Một công ty	Một cộng đồng
Phát triển sản phẩm	Định hướng bởi lợi ích kinh tế tập đoàn	Định hướng bởi tính năng sản phẩm
Lập trình viên	Hạn chế số lượng người biết về sản phẩm, tất cả giá trị thuộc về nhà cung cấp	Số lượng thay đổi từ nhỏ đến rất lớn, thường xuyên làm việc, bảo đảm, tình nguyện
Độ ổn định	Một phương hướng mới thiết lập nhanh chóng khi có động lực kinh tế	Những chương trình phát triển gắn bó chặt chẽ với sản phẩm khi điều đó mang lại lợi ích cho người dùng
Người sử dụng	Nhìn chung là không hợp tác, mọi người sử dụng thiết lập mối quan hệ với nhà cung cấp một cách độc lập	Người sử dụng tương tác với nhau trong môi trường giao tiếp ảo, thảo luận với nhau về sản phẩm và tương lai phát triển

☐ Cần thiết có các phương pháp đánh giá về nguồn mở

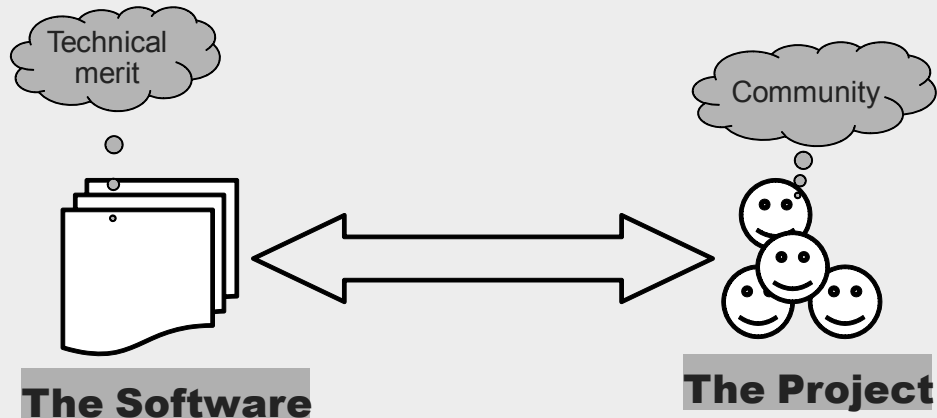
Source: « Open Source Maturity Model », Capgemini expert letter, 2003

TẠI SAO CẦN ĐÁNH GIÁ CÁC DỰ ÁN MÃ MỞ ?

- ☐ Để tìm ra các dự án mà có thể bạn đang cần
- ☐ Để so sánh chúng với các dự án độc quyền
- ☐ Để lựa chọn hoặc từ chối một số dự án khác
- ☐ Để cân bằng giữa mức độ tiếp thị, quảng cáo, báo cáo phân tích giả mạo với thông tin khách quan
- ☐ Để có thể dễ dàng đưa ra các quyết định và sẵn sàng đưa báo cáo đó vào quản lý

ĐÁNH GIÁ CÁC DỰ ÁN MÃ MỞ CÁI GÌ LÀ MẪU CHỐT ?

- ❑ **400,000 dự án mã mở- Dự án nào cho bạn ?**

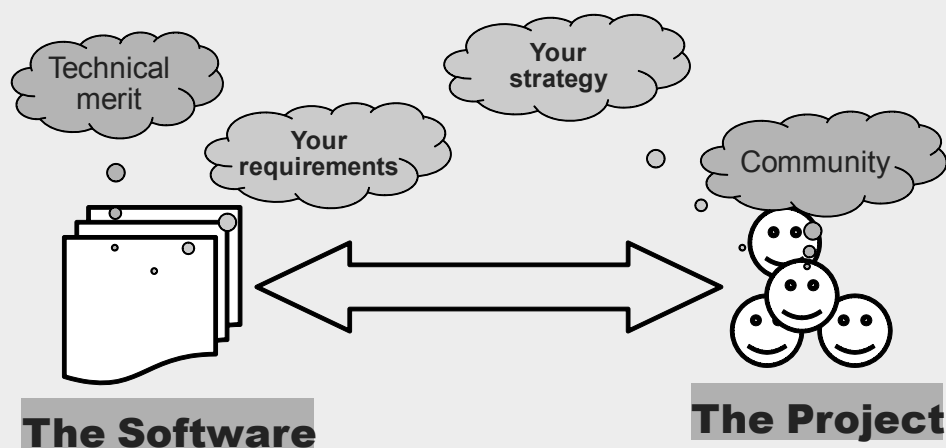


CHUẨN CHUNG

- ❑ **Thời gian dự án**
- ❑ **Số lượng lập trình viên chính**
- ❑ **Tốc độ phát hiện và sửa lỗi**
- ❑ **Diễn đàn và lưu lượng thư trao đổi**
- ❑ **Số lượng tải về**
- ❑ **Số lượng người dùng**
- ❑ **Chất lượng mã nguồn**
- ❑ **Chất lượng tài liệu đóng gói**

ĐÁNH GIÁ CÁC DỰ ÁN MÃ MỞ CÁI GÌ LÀ MẪU CHỐT ?

- ❑ 400,000 dự án mã mở – Dự án nào cho bạn ?



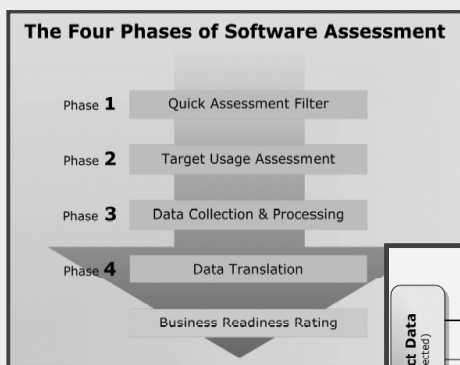
TỔNG QUAN VỀ MỘT QUÁ TRÌNH ĐÁNH GIÁ

- ❑ Định nghĩa các yêu cầu và tầm quan trọng
- ❑ Định vị và đưa ra danh sách đầu tiên
- ❑ Đánh giá
- ❑ Danh sách ngắn gọn cuối cùng sau đánh giá
- ❑ Lựa chọn dựa trên điểm số cuối cùng
- ❑ Kết quả thu được
- ❑ Khả năng thích ứng với mô hình sử dụng

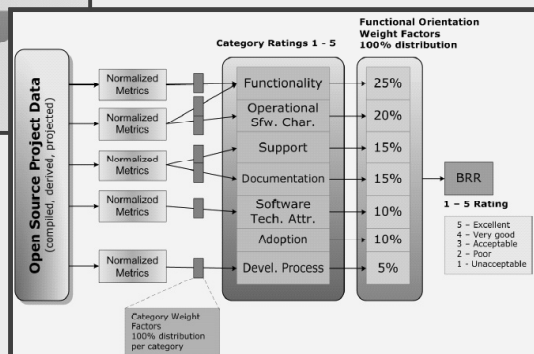
MỘT SỐ PHƯƠNG PHÁP ĐÁNH GIÁ

- ☐ Open BRR
- ☐ OSMM (Navica)
- ☐ OSMM (Capgemini)
- ☐ QSOS

OPENBRR



Source: « the Business Readiness rating: a framework for Evaluating Open Source », A. Wasserman, M. Pal & C. Cham, 2006



OPEN BRR

❑ Các lĩnh vực đánh giá:

- ❖ Tính năng
- ❖ Đặc điểm hoạt động
- ❖ Hỗ trợ và dịch vụ
- ❖ Tài liệu
- ❖ Thuộc tính công nghệ
- ❖ Thích nghi
- ❖ Tiến trình phát triển

Có thể sử dụng để so sánh
với phần mềm đóng

Chi tiết cho dự án mã mở

OPEN BUSINESS READINESS RATI

- ❑ Khởi xướng bởi Đại học Carnegie Mellon West
- ❑ Tiếp cận một cách khá khoa học
- ❑ Tham khảo tiến trình đánh giá chuẩn
(ISO/IEC 9126, ISO/IEC 25000)
- ❑ Dự án có vẻ đã ngưng trệ từ đó đến nay

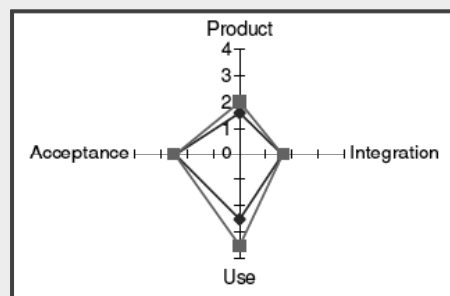
OSMM (CAPGEMINI) (1/2)

❑ **OSMM là một phần thêm vào tiến trình giám định và bán hàng của Capgemini**

- ❖ Xác định chu kỳ sống của “sản phẩm” FLOSS
- ❖ Phù hợp với các yêu cầu kinh doanh
- ❖ So sánh nguồn mở với các sản phẩm đóng khác
- ❖ Chỉ ra tầm quan trọng của “đối tác nguồn mở”

OSMM (CAPGEMINI) (2/2)

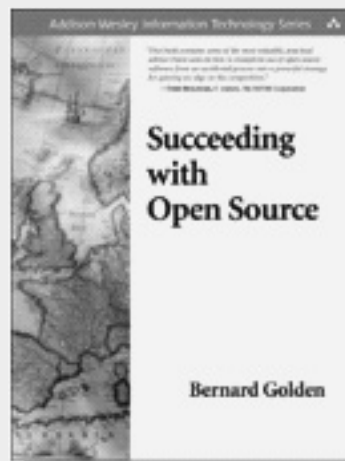
Product indicators		
Indicator	Product A	Product B
Product		
Age	1	3
Licensing	2	2
Human hierarchies	1	2
Selling points	2	1
Developer community	2	2
Integration		
Modularity	1	1
Collaboration with other products	2	2
Standards	2	2
Use		
Support	2	3
Ease of deployment	3	4
Acceptance		
User community	2	3
Market penetration	3	2
TOTAL	23	27



Source: « Open Source Maturity Model », Capgemini expert letter, 2003

OSMM (NAVICA) (1/2)

- ❑ **Xuất hiện đầu tiên trong cuốn “Succeeding with Open Source” (Addison-Wesley, 2004)**
- ❑ **Khái niệm “maturity” như một sự thay thế độc quyền “proprietary” SLA**
 - ❖ Phần mềm
 - ❖ Hỗ trợ
 - ❖ Tài liệu
 - ❖ Đào tạo
 - ❖ Tích hợp sản phẩm
 - ❖ Dịch vụ chuyên nghiệp



OSMM (NAVICA) (2/2)

- ❑ **OSMM là phương pháp đánh giá dựa trên điểm số**
 - ❖ Các yếu tố đánh giá sản phẩm
 - ❖ Giá trị các yếu tố áp dụng vào sản phẩm
 - ❖ Tính toán tổng điểm sản phẩm OSMM

Phần mềm	4
Hỗ trợ	2
Tài liệu	1
Đào tạo	1
Tích hợp	1
Dịch vụ chuyên nghiệp	1

Minimum / 100	Thích ứng sớm	Thực dụng
Sự thử nghiệm	25	40
Tìm hướng	40	60
Sản xuất	60	70

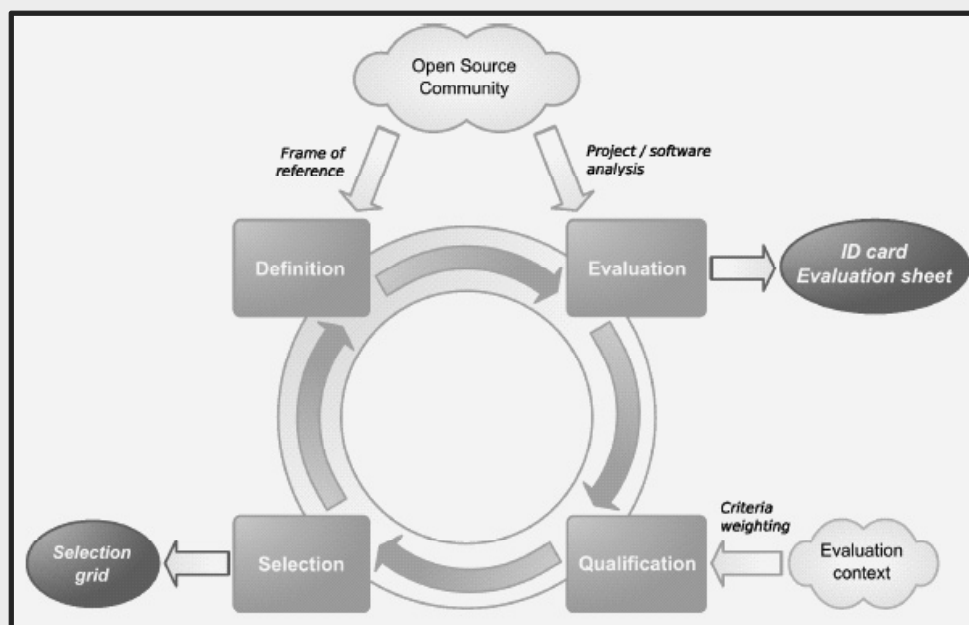
QSOS

- ❑ Khởi tạo bởi Atos origin, sau đó miễn phí (GPL, FDL)
- ❑ Tính tương thích, các công cụ nguồn mở
- ❑ Cộng đồng dựa trên đánh giá :
 - ❖ ID cards có thể đóng góp vào cộng đồng
 - ❖ Phẩm chất của mỗi cá nhân trong các điều kiện ngoại cảnh.
 - ❖ Các công cụ đánh giá trực tuyến (O3S) cho việc so sánh

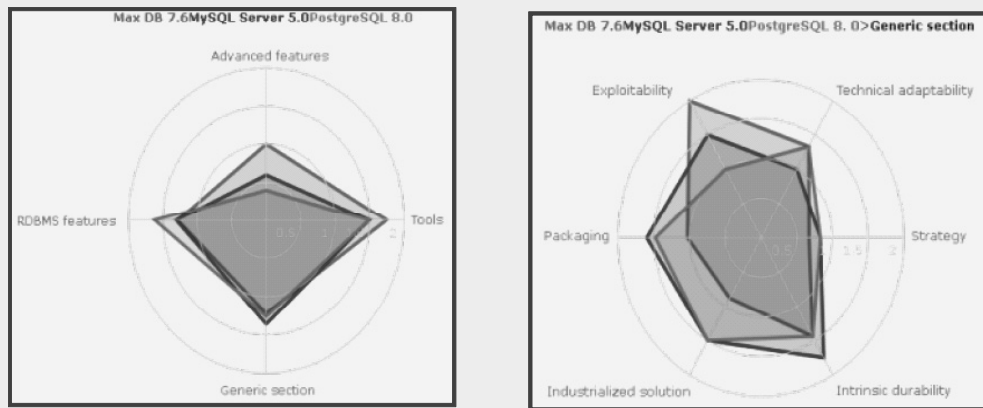


PHƯƠNG PHÁP LẬP... VÀ MỖI CÔNG VIỆC TRONG QUÁ TRÌNH

QSOS

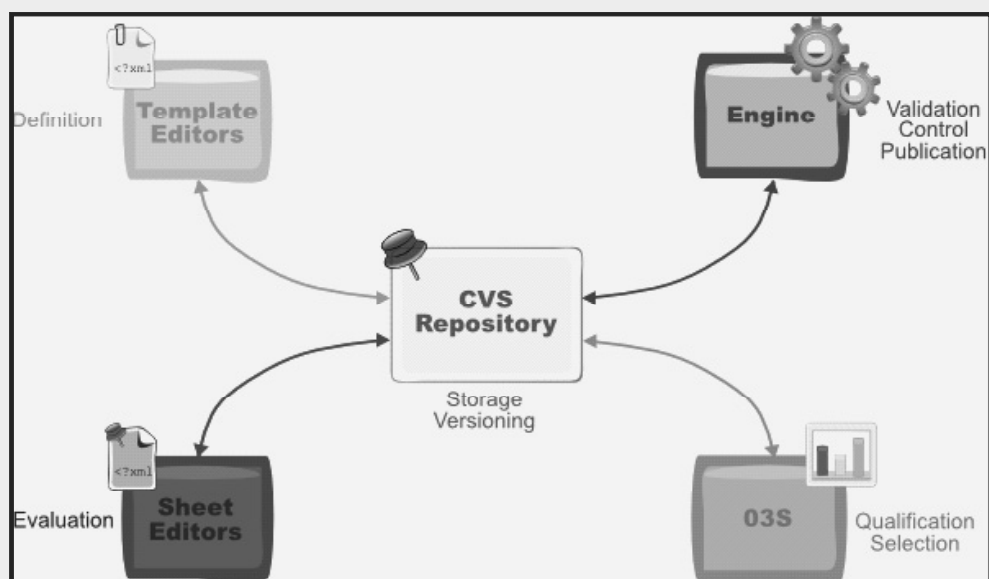


QSOS



Source: the Open Source Business Resource, May 2008

QSOS



“PHÁT HÀNH SỚM/THƯỜNG XUYỀN” KHI MÀ DỞ NGHĨA LÀ TỐT

- ☐ **Nhiều lỗi được phát hiện là dở**
 - ❖ ... hay có thể chứng minh nhiều người quan tâm
 - ❖ ... và là cơ hội để phát triển bởi cộng đồng
- ☐ **Lưu lượng ít là dở...**
 - ❖ ... hoặc nghĩa là dự án đang đóng cứng...
- ☐ **Số lượng tải về ít là dở ...**
 - ❖ ... trừ phi dự án là tiện ích nhỏ nhúng trong các dự án lớn

METRICS PITFALLS - CAM VÀ TÁO

- ☐ **Bạn có so sánh dữ liệu của cùng một ý nghĩa?**
- ☐ **Các thước đo bạn sử dụng có làm rõ ràng trong bối cảnh dự án đó ?**
- ☐ **Đánh giá một dự án đơn lẻ là tuyệt vời. Nhưng về tính tương hợp thì sao?**
- ☐ **Bạn có đưa các yêu cầu của mình vào đó?**

CÁC CHUẨN MỞ

- ☐ **Tuân thủ theo các chuẩn mở**
- ☐ **Chứng nhận việc tuân thủ**
- ☐ **Ai có thẩm quyền cấp chứng nhận?**
- ☐ **Những phiên bản nào được chứng nhận?**
- ☐ **Lộ trình?**

MỘT CẶP DẤU HIỆU CẢNH BÁO

- ☐ **Cấu trúc đơn lẻ**
- ☐ **Mã nguồn không chú giải & không tài liệu**
- ☐ **Có nhiều bản trình diễn và báo cáo hơn mã nguồn**
- ☐ **Một số ít đăng kí tham gia và theo dõi 24/7**

TƯƠNG THÍCH CÁC MÔ HÌNH SỬ DỤNG

- ☐ Chiến lược và chiến thuật
- ☐ Sự ổn định và cơ hội
- ☐ Sản xuất và đầu tư
- ☐ Thích ứng sớm và thực tiễn
- ☐ Sự quên lãng và nhiệm vụ cấp bách.
- ☐ Người dùng cuối và các gian hàng công nghệ
- ☐ Quyết định theo kỹ thuật vs quyết định từ trên xuống

MỘT VÀI ĐIỂM BẮT ĐẦU

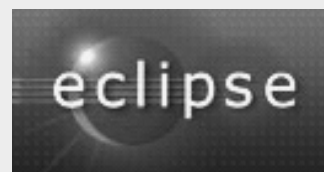
- | | |
|--------------------------------------|---|
| ☐ SourceForge | ☐ Ohloh |
| ☐ CodeHaus | ☐ SourceKibitzer |
| ☐ OW2 | ☐ Open Source Vulnerability Database |
| ☐ Savannah | ☐ FLOSS Mole |
| ☐ Gna | |
| ☐ BerliOS | |
| ☐ FreshMeat | |
| ☐ Tigris | |
| ☐ Java.net | |
| ☐ Koders | |
| ☐ ... Google! | |

OBJECTWEB



- ☐ **Phi lợi nhuận , EU khởi xướng , bắt đầu từ năm 2002 , phát triển thành OW2 năm 2007**
- ☐ **Trọng tâm: middleware, tuân theo chuẩn mở**
- ☐ **Thành viên là các công ty và cá nhân**
- ☐ **Mong muốn đưa chính phủ gần hơn với các công ty**
- ☐ **73 công ty thành viên ,hơn 120 dự án vào cuối năm 2006.**
- ☐ **Đặt trong phân tích các báo cáo (Gartner MQs) cùng với ISVs**

CƠ SỞ DỮ LIỆU ECLIPSE



- ☐ **Phi lợi nhuận, EU, bắt đầu năm 2003 với IBM**
- ☐ **Trọng tâm vào phát triển môi trường lập trình eclipse (Java)**
- ☐ **Các công ty thành viên, thành viên cao cấp, đề xuất các giá trị khác nhau cho lập trình viên và người dùng**
- ☐ **Quản lý theo hình thức hành chính tới quản trị dự án**

LỢI ÍCH CỦA TỔ CHỨC CUNG CẤP THỨ 3 VÀ NGƯỜI DÙNG

❑ **3GOS: Lợi nhuận cho nhà cung cấp**

- ❖ Giảm thiểu rủi ro
- ❖ Kinh tế qui mô lớn
- ❖ Trao đổi về công nghệ
- ❖ Hợp tác hoặc ngăn chặn cạnh tranh
- ❖ Tạo điều kiện thuận lợi cho sự phát triển quốc tế
- ❖ Hợp tác ngang hàng

❑ **Giảm thiểu rủi ro của các nhà cung cấp khiến khách hàng thoải mái hơn**

Source: Contractor, F.J., Lorange, P. (1988), « Why should firms cooperate? The strategy and economics basis for cooperative ventures »

KIẾN TRÚC DỰA TRÊN NGUỒN MỞ CÓ KHẢ NĂNG LIÊN THÔNG

SOẠN BỞI: FRANÇOIS LETELLIER

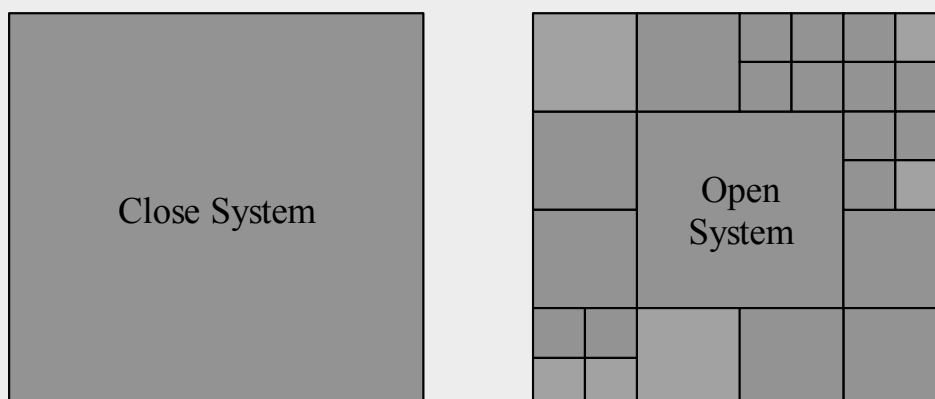
CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT

PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 28 /2008/06/30

CHÚNG TA SẼ HỌC GÌ...

- ☐ Thế nào là hệ thống đóng và mở
- ☐ Các nguyên tắc tương hợp
- ☐ Các chuẩn kỹ thuật cần thiết
- ☐ Quy trình chứng nhận
- ☐ Vai trò của FLOSS trong sự tương hợp

SỰ CẠNH TRANH GIỮA HỆ THỐNG ĐÓNG VÀ MỞ



Nhìn từ kiến trúc stand-point, 1 hệ thống mà không bị giới hạn ranh giới theo khuôn mẫu nào gọi là hệ thống mở, vấn đề nằm ở việc xem xét thiết kế.

TÍNH TƯƠNG HỢP

❑ Theo wikipedia

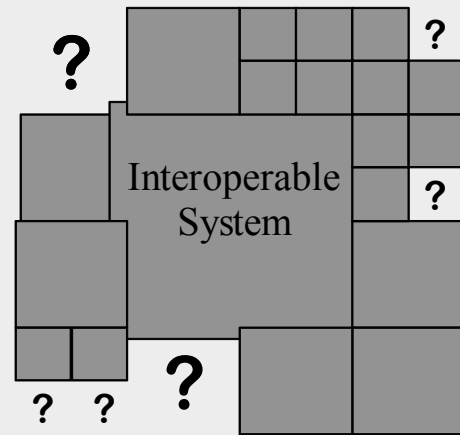
- ❖ “Tính tương hợp là khả năng cho phép các hệ thống và các tổ chức khác nhau làm việc cùng nhau (tác động lẫn nhau). Thuật ngữ này thường được dùng trong các hệ thống kỹ thuật.”

❑ Theo ISO/IEC 2382-01

- ❖ “Tính tương hợp được xác định là khả năng giao tiếp, thực thi các chương trình, hoặc truyền các dữ liệu giữa hàng loạt các bộ phận chức năng theo cách mà nó không đòi hỏi sự hiểu biết của người sử dụng đối với các chức năng duy nhất của các bộ phận này”

HỆ THỐNG TƯƠNG HỢP

- ❑ Hệ thống tương hợp là hệ thống mở trong đó chỉ rõ sự tương tác qua lại giữa các hệ thống con có sẵn đối với các nhà thiết kế phần mềm thứ 3 (third-party)
- ❑ Các Framework đặc tả kỹ thuật thường bao gồm: APIs, SPIs, Protocols,
- ❑ Formats, Grmmars, ...



CHUẨN KỸ THUẬT LÀ GÌ?

- ❑ Một khung các đặc tả kỹ thuật
 - ❖ Có thể được cấp phép bởi 1 tổ chức có thẩm quyền (de jure)
 - ❖ Hoặc có thể được cộng đồng tự chấp nhận và sử dụng rộng rãi (de facto)
- ❑ Chuẩn khác với phần mềm (không bao hàm code)
 - ❖ Chúng có thể là các định dạng file, các giao thức
 - ❖ truyền thông, giao diện lập trình ứng dụng, ...
 - ❖ Ví dụ: TCP/IP (IETF); HTTP (W3C, ISO); POSIX (Open Group, IEEE, ISO)

VẬY TẠI SAO CÁC CHUẨN LẠI QUAN TRỌNG?

- ☐ **Giải quyết vấn đề một lần, không phải mọi lúc**
- ☐ **Tạo sự dễ dàng khi kết hợp các thành phần từ nhà cung cấp khác nhau**
- ☐ **Thông báo cho khách hàng biết các tính năng của sản phẩm trước khi mua**
- ☐ **Mở ra một thị trường đông đảo khách hàng**
- ☐ **Đảm bảo ổn định theo thời gian**

CÁC CHUẨN MỞ, VẤN ĐỀ LÀ GÌ?

- ☐ **Công nghệ mở khác với Kinh doanh mở**
- ☐ **=> Một khi bạn tham gia vào bản hợp đồng, vấn đề là phân phối hay bán phần mềm tương hợp**
- ☐ **Các nguyên tắc kinh doanh chung:**
 - ❖ Bán lưỡi dao đắt, còn cái dùng để cạo thì miễn phí
 - ❖ Buộc người dùng phải nâng cấp nếu muốn duy trì truy cập dữ liệu
 - ❖ Buộc các đối tác phải điều chỉnh chiến lược kinh doanh
 - ❖ Tạo sự khan hiếm các hệ thống tương hợp
 - ❖ Điều khiển truyền thông các thông số kỹ thuật

ĐỊNH NGHĨA CHUẨN MỞ THỨ #1

❑ Theo định nghĩa của Bruce Perens

- ❖ Tính sẵn có
- ❖ Tối đa hóa lựa chọn cho người dùng cuối
- ❖ Không đặc quyền
- ❖ Không có sự phân biệt
- ❖ Không chứng thực các tập hợp con
- ❖ Bảo vệ chống lại sự ăn cắp (hợp tác và mở rộng)

ĐỊNH NGHĨA CHUẨN MỞ THỨ #2

❑ Theo định nghĩa của EU IDABC

- ❖ Chuẩn mở được thông qua và duy trì bởi:
 - 1 tổ chức phi lợi nhuận (S.O)
 - Với 1 quá trình ra quyết định mở cho các bên quan tâm
- ❖ Tài liệu kỹ thuật xuất bản có sẵn với chi phí thấp hoặc miễn phí
- ❖ Bằng sáng chế không bị thay đổi là có sẵn trên nền tảng tự do bản quyền
- ❖ Không có sự ràng buộc gì về việc tái sử dụng của các tiêu chuẩn

TUÂN THỦ CÁC CHUẨN: QUY TRÌNH CẤP GIẤY CHỨNG NHẬN

❑ 3 Tổ chức riêng biệt

- ❖ Standard O. ≠ Certification O. ≠ Testing O

❑ Nhãn hiệu chứng nhận

- ❖ Thuộc sở hữu của tổ chức chứng nhận

❑ Thỏa thuận chứng nhận

- ❖ Những quy định giữa nhà cung cấp phần mềm và tổ chức chứng nhận dựa trên nhãn chứng nhận

❑ Ví dụ đôi với Java EE

- ❖ Tổ chức cấp chuẩn: JCP; Tổ chức chứng thực: Sun; Tổ chức kiểm thử: ISV

TỪ MỞ TỚI ĐỘC QUYỀN SỞ HỮU: CHIẾN LƯỢC CỦA IEEE

❑ Cơ bản

- ❖ Nhà cung cấp chủ lực triển khai một chuẩn mở
- ❖ Nhà cung cấp đưa ra thị trường các sản phẩm tương thích hoàn toàn

❑ Mở rộng

- ❖ Nhà cung cấp thêm các thành phần độc quyền vào chuẩn
- ❖ Sản phẩm từng bước trôi khỏi chuẩn mở

❑ Hủy bỏ

- ❖ Các thành phần độc quyền trở thành chuẩn de facto
- ❖ Chuẩn mở bị hạn chế để đẩy kỹ thuật độc quyền

FLOSS VÀ CÁC CHUẨN MỞ

- ❑ **Tồn tại trong việc triển khai FLOSS có lẽ là sự ngăn cản từ IEEE**
- ❑ **Các dự án của FLOSS thích hợp để phát triển các phần mềm mà có thể tương hợp với các hệ thống đóng và tuân thủ pháp luật (ở nhiều quốc gia) thông qua công nghệ đảo**

Một số hạn chế cho DMCA / EUCD: nó được cho là quy trình mà giúp đỡ sự giả mạo trong việc thực hiện các biện pháp bảo vệ chống lại sao chép hoặc sử dụng phần mềm bất hợp pháp.

- ❑ **FLOSS và các tiêu chuẩn mở chia sẻ cùng một tầm nhìn về sự sẵn sàng, tính mở và sự tham gia.**

KIẾN TRÚC THU THẬP THÔNG TIN

Lab Activity

- ❖ Đọc tài liệu 1 cách cẩn thận (trên wiki)
- ❖ Tìm kiếm các quy trình và các chuẩn liên quan đến hệ thống
- ❖ Tìm kiếm các nền tảng mã nguồn mở và các thành phần mà có thể áp dụng cho những thiết kế
- ❖ Dựa vào các slide mẫu được cung cấp, viết 1 danh sách các chuẩn và các thành phần đã tìm được ở trên
- ❖ Dựa vào các slide mẫu được cung cấp, chọn 1 sơ đồ mẫu phù hợp nhất và điền vào đó các thành phần mã nguồn mở mà bạn đã lựa chọn
- ❖ Điền vào slide các nguyên tắc thiết kế và động cơ bạn lựa chọn như vậy
- ❖ Điền vào slide những điểm mạnh và yếu trong thiết kế của bạn

TỔNG KẾT

- ❑ Các thiết kế “hệ thống mở” là kiến trúc hiện đại dựa trên tiêu chuẩn về tính tương hợp
- ❑ Các chuẩn mở và các dự án FLOSS rất quan trọng với sự “cân bằng hệ sinh thái”

FOSS-BRIDGE-LAB28

SOẠN BỞI: FRANÇOIS LETELLIER
CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
PHIÊN BẢN / NGÀY : FOSS-BRIDGE-LAB28 /2008/06/30

FossToday, một cổng thông tin cho cộng đồng FLOSS Thế nào là hệ thống đóng và mở

❑ **Sơ bộ Kiến trúc Soạn bảo <bạn>**

TỔNG QUAN HỆ THỐNG

❑ **Cổng thông tin FossToday cho phép người dùng:**

- ❖ Đọc những bài viết bởi các thành viên FossToday
- ❖ Xem nội dung các site của các hãng thứ 3
- ❖ Đọc tin tức của Foss được xuất bản bởi các đối tác
- ❖ Đăng ký làm thành viên của Foss Today
- ❖ Gửi ảnh bằng tin nhắn nếu muốn thu thập thêm thông tin

CÁC CHUẨN LIÊN QUAN

❑ **Chuẩn mở**

- ❖ <Tổ chức>
 - <Tên các chuẩn>
 -
- ❖ Các chuẩn De-facto
 - <Tên và thông tin của các chuẩn>
 - ...

CÁC KỸ THUẬT LỰA CHỌN

❑ Các ngôn ngữ

❖ <Liệt kê danh sách các ngôn ngữ bạn chọn>

❑ Middleware

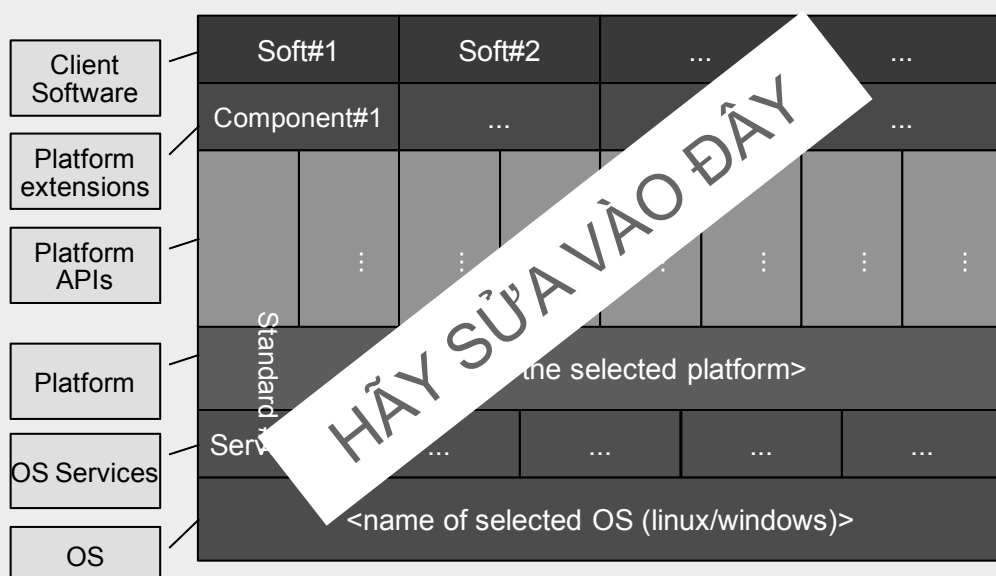
❖ <Liệt kê danh sách bạn chọn>

❑ Nền

❖ <Các nền (platform) bạn chọn>

❖ <Các platform modules đã chọn>

SƠ ĐỒ CÁC THÀNH PHẦN MỨC CAO



KẾT LUẬN

- ❑ **Các điểm mạnh trong thiết kế**
 - ❖ <Liệt kê những điểm mạnh trong thiết kế của bạn>
- ❑ **Các điểm yếu trong thiết kế**
 - ❖ <Liệt kê các điểm yếu còn tồn tại>

KHÁM PHÁ TRUNG TÂM LƯU TRỮ (FORGES) FOSS TIỀM NĂNG

SOẠN BỞI: CHRISTOPHE NEY
CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 29 /2008/06/30

XEM DEMO

Lab Activity

❑ Mỗi nhóm 2 người:

- ❖ Vào wiki và tìm đoạn phim demo tại phần: "Digging into SourceForge.net"
- ❖ Xem các demo 1, 2, 3, 4
- ❖ Đăng ký trên trang SourceForge.net
 - Sử dụng địa chỉ lab@flossexpert.com trong profile của bạn để chúng tôi có thể xác nhận tài khoản cho bạn.
 - Nếu bạn đã có tài khoản, hãy gửi các thông tin: tên đầy đủ và tên đăng nhập của bạn tới địa chỉ email lab@flossexpert.com

❑ Thời gian cho mỗi nhóm là 30 phút

CHÚNG TA SẼ HỌC ...

❑ Open-Source Forges

- ❖ Là gì?
- ❖ Từ đâu đến?
- ❖ Có thể làm gì với chúng?
- ❖ Có thể tìm chúng ở đâu?
- ❖ Tìm đường để đến với chúng như thế nào?

FORGE LÀ GÌ?

“Forge được hình dung như 1 lò rèn mà ở đó người thợ rèn giữ lửa trong đó để đốt nóng kim loại”

❑ Trong môi trường FLOSS

- ❖ Một phần mềm (nguồn mở, giao diện web) quản trị dự án và cộng tác
- ❖ Một hệ thống quản lý sự phát triển những phần mềm cộng tác cho các dự án phần mềm tự do
- ❖ Kho lưu trữ mã nguồn và là nơi để những nhà phát triển phần mềm điều khiển và quản lý các phần mềm mã mở
- ❖ Là hệ thống quản lý sự phát triển phần mềm và kiểm soát những thay đổi của nó

LỊCH SỬ CỦA FORGES

“Forge được hình dung như 1 lò rèn mà ở đó người thợ rèn giữ lửa trong đó để đốt nóng kim loại”

- ❖ Forges ra đời từ nhu cầu lớn của cộng đồng mã nguồn mở để cung cấp các công cụ phát triển cho thành viên của mình (ví dụ: Savana / GNU)
- ❖ SourceForge.net ra đời năm 1999 bởi VA Linux và là kiến trúc trên nền web đầu tiên miễn phí cho tất cả mọi người. Phần mềm SourceForge giấy phép GPL ra đời cùng lúc.
- ❖ Sau SourceForge.net, nhiều dự án mã nguồn mở khác đã ra đời như: Phát hành các Platform quản lý sự phát triển phần mềm nguồn mở- các Hosting miễn phí cho các dự án nguồn mở

CÁC NGUYÊN TẮC CỦA FORGES

❑ **Tuân thủ theo mô hình cộng tác “Bazaar⁽¹⁾” của Eric S. Raymond, các nguyên tắc chính là:**

- ❖ Phát hành sớm, phát hành thường xuyên
- ❖ Mọi người thường ưa thích giải quyết các vấn đề họ quan tâm hơn
- ❖ Theo dõi đầy đủ thì sẽ hạn chế phát sinh lỗi
- ❖ Nhìn chung, các dự án đang sống có giá trị nhiều hơn các dự án không hoạt động, dự án chết

(1): Cathedral và Bazaar (viết tắt là CatB) là một tiểu luận viết bởi Eric S. Raymond về phương pháp kỹ thuật phần mềm

GÓC NHÌN TRUNG TÂM DỰ ÁN

Forges là những hệ thống phát triển trên nền web để quản lý các dự án độc lập (1 tập hợp các dự án mã mở)

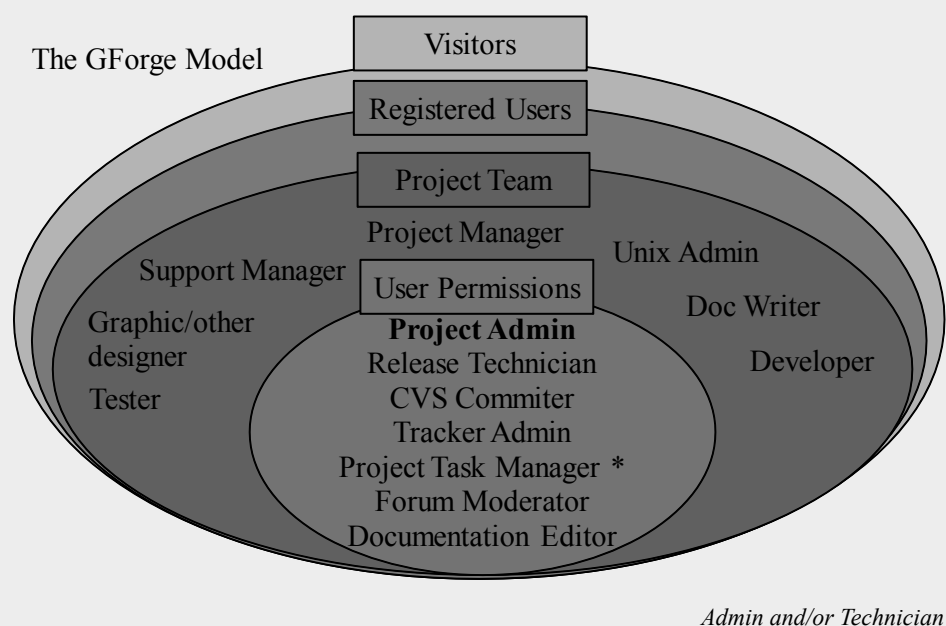
❑ **Một dự án:**

- ❖ Được quản lý bởi một đội (gồm 1 hoặc nhiều người và mỗi người có vai trò riêng)
- ❖ Người lo phần mã public với giấy phép được đưa ra
- ❖ Người lo phần phát hành tài liệu và các gói phần mềm
- ❖ Người đảm trách phần giao tiếp với thế giới bên ngoài (như các site, forum, mailing lists...)

CÁC THÀNH PHẦN LIÊN QUAN TỚI FORGE (VÍ DỤ: GFORGE)

- ❑ **Trang chủ của Forge**
 - ❖ Bao gồm các phần: Edito, News, Stats, Registration/Login
- ❑ **Trang dành cho người dùng**
 - ❖ Bao gồm các items, Diary, Account, Shortcuts, Bookmarks
- ❑ **Thư mục dành cho người dùng**
 - ❖ Bao gồm: Profile, Web Message, People Search, Advocado
- ❑ **Bản đồ liên kết dự án**
 - ❖ Bao gồm: Project List/Tree, Software Search
- ❑ **Forge Project Openings**
 - ❖ Bao Job Posting, Skill Search các dự án không hoạt động, dự án chết

VAI TRÒ TIỀM NĂNG VÀ HIỆN TẠI CỦA FORGES



CÁC THÀNH PHẦN ĐIỂN HÌNH TRONG MỖI DỰ ÁN CỦA FORGE

- ❖ **Summary** (Tổng quan của dự án)
- ❖ **Admin** (Quản trị các trang trong dự án)
- ❖ **Forums** (Nơi diễn ra các thảo luận)
- ❖ **Tracker** (Theo dõi các vấn đề của dự án)
- ❖ **Lists** (Project Mailing Lists)
- ❖ **Task** (Quản lý Gantt trên nền web đơn giản)
- ❖ **Docs** (Các tài liệu cơ sở)
- ❖ **Survey** (Khảo sát dựa trên web đơn giản)
- ❖ **News** (RSS Mashable Site News)
- ❖ **CVS, SVN** (Sử dụng để quản lý mã nguồn)
- ❖ **Files** (Phân phối các gói, phát hành các phiên bản và các file)
- ❖ **Wiki**

DANH SÁCH HOÀN CHỈNH CÁC FORGE ENGINE

Quản lý các dự án nguồn mở và các phần mềm cộng tác trên nền web

- ❖ **Savane** (Dự án GNU Savannah)
- ❖ **SourceForge** (Mở đầu cho dự án SourceForge.org)
- ❖ **GForge** (Nhánh SourceForge thứ 1)
- ❖ **Track** (Hệ thống Forge dựa trên wiki)
- ❖ **Codex/Codendi** (Nhánh SourceForge khác)
- ❖ **Origo** (Forge dựa trên P2P)
- ❖ **Libre Source** (Forge dựa trên JEE)
- ❖ **NovaForge** (Forge dựa trên Java Portel)
- ❖ (như các site, forum, mailing lists...)

DANH SÁCH HOÀN CHỈNH CÁC CƠ SỞ CỦA FORGE

- ❖ SourceForge.net (170 000+ project, 1,8M users)
- ❖ GNU Savannah
- ❖ Apache.org
- ❖ Google Code
- ❖ Novell Forge
- ❖ OW2 ObjectWeb Forge
- ❖ CodePlex (Microsoft)
- ❖ Gna!
- ❖ Java.net
- ❖ Tigris

TÌM KIẾM DỰ ÁN TRONG KHO LƯU TRỮ CỦA SOURCEFORGE

- ❑ **Một nhóm 3 người:**
 - ❖ **1/ Chọn 1 phần mềm quen thuộc trong danh sách**
 - ❖ **2/ Tìm 3 dự án "credible" trên SF.net**
 - ❖ **3/ Viết các dự án bạn tìm với link tương ứng lên wiki**
 - ❖ **4/ Chia sẻ với mọi người**
 - ❖ **Viết nhận xét về chúng**
 - ❖ **Giải thích tại sao chúng đáng tin cậy**
- ❑ **Thời gian**
 - ❖ **Cho mỗi Group : 30 phút**
 - ❖ **Trao đổi giữa các nhóm : 15 phút**

TÓM TẮT

Quản lý các dự án nguồn mở và các phần mềm cộng tác trên nền web

- ☐ **Forges là kiến trúc phát triển trên nền web cho các dự án mã nguồn mở**
- ☐ **Forges mang lại khả năng phát triển phần mềm dựa vào cộng đồng**
- ☐ **Có nhiều tùy chọn sẵn có cho bạn để host các dự án mã nguồn mở**
- ☐ **Forges là “mỏ vàng” của các phần mềm mã mở và mã nguồn**

MỐI LIÊN HỆ VÀ HIỆU QUẢ SỬ DỤNG CỦA FLOSS FORGES

SOẠN BỞI: CHRISTOPHE NEY
CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
PHIÊN BẢN / NGÀY : FOSS-BRIDGE KHỐI 30 / 2008/08/08

CHÚNG TA SẼ TÌM HIỂU ...

❑ **Open-Source Forges**

- ❖ Làm thế nào để tạo ra các dự án của riêng bạn trên đó
 - Làm thế nào để cấu hình lại dự án của bạn
- ❖ Làm thế nào các dự án được quản lý bởi quản trị viên Forge
- ❖ Làm thế nào để tạo ra hiệu quả sử dụng của các tính năng Forge
 - Giao tiếp
 - Chia sẻ kết quả
 - Tổ chức công việc
 - Đo sự tiến bộ

TẠO DỰ ÁN SANDBOX CỦA BẠN

Lab Activity

❑ Nhóm gồm 2 người:

- ❖ 1/ Vào wiki và tìm flash trình diễn trên trang "Tạo dự án của riêng bạn trên GForge".
- ❖ 2/ Xem trình diễn trang 5,6,7,8
- ❖ 3/ Wiki làm theo hướng dẫn và đăng ký dự án "Sandbox" của riêng bạn trên NetNam FossBridge Forge

❑ Thời gian

- ❖ 30 phút

TẠO MỘT DỰ ÁN TRÊN SOURCEFORGE.NET

❑ Cài đặt một dự án mới trên SF.net là một quá trình dễ dàng

- ❖ Đăng ký người sử dụng
- ❖ Điền một vài dạng "dự án sáng tạo"

❑ Nhưng ...

- ❖ Không dễ để giải các câu trả lời cho việc tạo dự án, vì vậy dự án có thể được chấp nhận hoặc không

WARNING !

- Trong hầu hết các cộng đồng nguồn mở, quyết định lưu trữ do một trong các đơn vị quản lý. Tiến trình đăng ký thường là phức tạp và đòi hỏi đàm phán cũng như hợp đồng.
- ❖ Ngay cả trên SF.net, thời gian để quyết định cũng kéo dài vài tuần

QUẢN TRỊ GFORGE

Live Demo

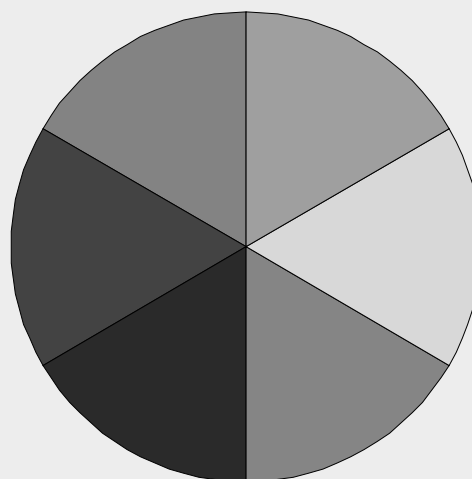
- ❖ Giới thiệu nhanh các công cụ quản trị GForge
- ❖ Các dự án Quản trị trang và Quản trị Tin tức
- ❖ Tiến trình việc phê duyệt/từ chối các dự án mới
- ❖ Các công cụ giám sát



QUẢN LÝ DỰ ÁN CỦA BẠN "CAREFULLY SHARE THE PIE"

Description
Trove Map
Roles
Forums
Tracker
Lists
Tasks
Docs
Survey
News
SCM
Releases
Wiki
Web Site
Screenshots

Project Facilities



- Leadership
- Animation
- Information
- Production
- Strategy
- Communication

HIỂU VỀ PROJECT ADMIN

Project Admin là nơi để tổ chức những gì người khác có thể thấy và có thể làm

- ❑ **Hầu hết các hành động (tái) cấu hình có thể thực hiện từ menu Admin**
- ❑ **Thường dùng để:**
 - ❖ Đăng ký và cấp quyền cho thành viên
 - ❖ Quản lý các diễn đàn, danh sách gửi thư, trackers
 - ❖ Kích hoạt/Vô hiệu hoá dịch vụ bổ sung
 - ❖ Tìm hiểu thêm thông tin (số liệu thống kê, giám sát)

TẬP TRUNG VÀO MAILING LIST

- ❑ **Mailing list có ích cho các thành viên tham gia, cũng như những người ghé thăm thường xuyên**
- ❑ **Ví dụ**
 - ❖ myproject-announce@forge.net : cho các thông báo và tin tức
 - ❖ myproject-user@forge.net : cho các thảo luận dự án liên quan đến người sử dụng
 - ❖ myproject-team@forge.net : cho thảo luận liên quan đến các thành viên trong nhóm
 - ❖ myproject-user-vn@forge.net : cho các thảo luận trong việt nam dự án liên quan đến người sử dụng

TẬP TRUNG VÀO CÁC DIỄN ĐÀN THẢO LUẬN

- ❑ **Diễn đàn có ích cho khách không thường xuyên, cho các chủ đề thường xuyên hoặc phát sinh**
- ❑ **Ví dụ:**
 - ❖ Getting Started: Giúp người dùng bắt đầu
 - ❖ FAQs: Những câu hỏi thường gặp
 - ❖ Announcements: thông báo của dự án
 - ❖ Cookbooks: sách vở hướng dẫn
 - ❖ Demos: Danh sách các trang web Demo
 - ❖ Vendors: Danh sách các lời chào mời thương mại liên quan đến dự án

TẬP TRUNG VÀO DANH SÁCH THƯ

- ❑ **Cẩn thận về vị trí: Đặt dự án của bạn vào đúng phần tương ứng trong Trove Map, viết mô tả thật hấp dẫn, tải lên các ảnh chụp màn hình, liên kết với một trang web “thương mại” (không phải trên SF)**
- ❑ **Tăng sự chú ý: Cung cấp diễn đàn cho những người mới đến, và chuyên môn danh sách gửi thư cho những người say mê**

TẬP TRUNG VÀO PHẦN MỀM

Các nhóm dự án nguồn mở có nghĩa vụ công bố mã nguồn khi phân phối phần mềm ...và được khuyến khích đưa ra các phần mềm được đóng gói sẵn

❑ **Ví dụ sử dụng công cụ :**

- ❖ Mã nguồn chia sẻ trên SVN (trunk/tags/branch)
 - Cách xây dựng cần đơn giản và tránh lỗi
- ❖ Các gói để người dùng tải về
 - Các gói cho Linux/Windows/Mac/Debian -hoặc- Java5, Java6, Tomcat ...
 - Đặt tên phiên bản (0.1, 0.2, 1.0, 1.1)
 - Gắn cờ (alpha1, beta, RC1, RC2, Final, Maintenance)
- ❖ Tài liệu riêng
 - Tài liệu thiết kế, các thay đổi theo phiên bản, lộ trình

TẬP TRUNG VÀO TRACKER

Trackers là cách hiệu quả để quản lý số lượng tác vụ nhỏ quan trọng

❑ **Ví dụ**

- ❖ Lỗi, miếng vá, các yêu cầu về đặc tính

❑ **Một số nguyên tắc**

- ❖ Gán (thảo luận) vai trò cho thành viên trong nhóm trước khi tạo trackers
- ❖ Yêu cầu đăng nhập trước khi gửi bài đến trackers, nhưng làm mẫu đơn giản và để cho các cộng đồng cung cấp các bản vá lỗi và báo lỗi
- ❖ Dành đủ thời gian để xử lý tracker
- ❖ Không sử dụng sơ đồ Gantt, trừ khi bạn có một dự án phức tạp

DỌN DẸP

- ☐ **Đăng ký dự án của bạn trong trove Map**
- ☐ **Hãy chắc chắn dự án của bạn có một website**
 - ❖ Dùng forge summary/wiki còn hơn không có gì
- ☐ **Khảo sát sự hài lòng, nếu muốn**
- ☐ **Tải lên ảnh chụp màn hình, nếu có thể**
- ☐ **Vô hiệu hóa các tính năng không dùng đến**
- ☐ **Thoát ra và kiểm tra kết quả cho khách truy cập**

Gửi cho thế giới một Tin

THIẾT LẬP DỰ ÁN GFORGE SANDBOX CỦA BẠN

Lab Activity

- ☐ **Nhóm gồm 2 người:**
 - ❖ Trở lại dự án GForge của bạn
 - ❖ Đăng ký các thành viên trong nhóm và chỉ định vai trò
 - ❖ Tải tài liệu từ Wiki (tài liệu, nguồn, phần mềm) lên dự án của bạn
 - ❖ Tổ chức các dự án của bạn để nó là hấp dẫn đối với những người khác
- ☐ **Thời gian:**
 - ❖ Làm lab : 30 phút

HOW OPEN SOURCE PROJECTS SURVIVE POISONOUS PEOPLE

- ❑ **Xem phần đầu tiên của Google Talks về “How Open Source Projects Survive Poisonous People” bằng cách lịch sử của YouTube.com**
- ❑ **Chia sẻ suy nghĩ của bạn**
 - ❖ kinh nghiệm của riêng bạn
 - ❖ bạn đồng ý / không đồng ý?
- ❑ **Thời gian:**
 - ❖ 30 phút

ĐÓNG GÓP VÀO DỰ ÁN “HELLO WORLD DEMO”

Lab Activity

- ❑ **Nhóm gồm 2 người:**
 - ❖ Tới dự án Hello World Demo trên SF.net tại <http://sourceforge.net/projects/demo>
 - ❖ Kiểm tra xem nếu bạn là một trong các thành viên
 - Nếu có, hoàn thành những nhiệm vụ đã được chỉ định cho bạn
 - Nếu không, liên hệ với nhóm, và cố gắng để có được tham gia vào việc thảo luận và sản xuất
 - ❖ Chia sẻ kinh nghiệm của bạn với người khác
- ❑ **Thời gian**
 - ❖ Lab : 15 phút
 - ❖ Trao đổi nhóm : 15 phút

TÓM TẮT

- ❑ **Forges là công cụ mạnh, nhưng phức tạp cho các dự án mã nguồn mở**
- ❑ **Forges đưa ra giao diện khác nhau tùy thuộc vào vai trò (Admin, Project Admin, thành viên, khách ghé thăm)**
- ❑ **Các công cụ Forges có thể thiết lập và tổ chức để**
 - ❖ liên lạc hiệu quả
 - ❖ chia sẻ kết quả
 - ❖ tổ chức công việc
 - ❖ giám sát tiến trình

TẠI SAO VÀ LÀM THẾ NÀO ĐỂ XÂY DỰNG TỪ MÃ NGUỒN

SOẠN BỞI: CHRISTOPHE NEY
CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 31 /2008/06/30

NỘI DUNG :

- Tại sao phải chú ý về mã nguồn phần mềm?
- Các phương thức chung để lấy mã nguồn và tạo ra gói sản phẩm phần mềm FOSS.
- Những đặc trưng của các Hệ thống xây dựng mã nguồn mở như: Make, ANT, MAVEN, Eclipse và Continuum.

CÁC VẤN ĐỀ SẼ TÌM HIỂU

- ❑ **Tại sao phải chú ý về mã nguồn phần mềm?**
- ❑ **Các phương thức chung để lấy mã nguồn và tạo ra gói sản phẩm phần mềm FOSS**
- ❑ **Những đặc trưng của các Hệ thống xây dựng mã nguồn mở như: Make, ANT, MAVEN, Eclipse và Continuum.**

LÝ DO NÊN XÂY DỰNG TỪ MÃ NGUỒN

- ☐ Để làm một số công việc trước mắt hoặc để kiểm tra khả năng để làm công việc sau này
- ☐ Để đảm bảo phần mềm phân phối hoàn toàn là mã nguồn mở
- ☐ Để áp dụng sửa đổi trên phần mềm (sửa chữa lỗi, vá lỗi, nâng cấp)
- ☐ Để thêm mới các thành phần phần mềm (biểu tượng, dịch, tài liệu, chữ ký)
- ☐ Để thay đổi tùy chọn đóng gói (ví dụ: tar gz, zip, installer)

CÁC CÂU HỎI LỚN VỀ CÁC HỆ PHỤ THUỘC

- ❖ Phần cứng không đủ để chạy các mã không thực thi
- ❖ Hầu hết các phần mềm dựa trên phần mềm hãng thứ ba để chạy được (ví dụ: hệ điều hành, middleware) và được biên dịch (vd: trình biên dịch, IDE)
- ❖ Nhiều gói phần mềm nguồn sẽ gắn với thành phần thi hành được hãng thứ ba trong nó
- ☐ **Bạn phải chọn điểm dừng khi bạn lấy mã nguồn của một hệ thống**

CƠ SỞ ĐẲNG SAU SỰ TIẾP CẬN XÂY DỰNG MÃ NGUỒN MỞ

- ❑ **Hầu hết dự án nguồn mở được đưa ra với một cấu hình dựng sẵn, để tự động sinh ra các gói phần mềm**
 - ❖ 1/ Kiểu xây dựng này đảm bảo tính tích hợp của mã nguồn được viết bởi nhiều người, và việc chạy nhiều tiến trình kiểm tra
 - ❖ 2/ Một điều rất hấp dẫn cho lập trình viên là chỉ mất vài phút là có thể can thiệp để tạo sự thay đổi cho một dự án nguồn mở và cho nó chạy
 - ❖ 3/ Điều này giảm thiểu sự phụ thuộc vào các công cụ thương mại và nặng nề, vốn là sự cản trở sự can dự của các lập trình viên

TRƯỜNG HỢP KHÔNG CÓ GÌ ĐỂ “DỊCH” (VD: PHP, AJAX)

- ❑ **Dù không có gì để “dịch”, chuỗi xây dựng vẫn cần có:**
 - ❖ Đóng gói phần mềm
 - Tạo nhiều kiểu lưu trữ khác nhau
 - Lọc mã không có liên quan đến người dùng
 - Tải lên các gói
 - ❖ Áp dụng cấu hình vào máy cụ thể
 - ❖ Chạy thử nghiệm
 - ❖ Làm tài liệu hướng dẫn hoặc hình minh họa
 - ❖ Triển khai thử nghiệm

CÁC BƯỚC CHÍNH ĐỂ XÂY DỰNG DỰ ÁN MÃ NGUỒN MỞ

- ❑ **Các công cụ tự động xây dựng cho các dự án mã nguồn mở:**
 - ❖ Shell Scripts (bash, sh, csh, ksh ...)
 - ❖ Compatible Make files (Makefile)
 - ❖ Apache ANT Scripts (build.xml)
 - ❖ Ruby Rake Files (Rakefile)
 - ❖ Maven Assembly Descriptor (pom.xml)
 - ❖ Mono NANT Scripts (build.xml)

LÀM THẾ NÀO ĐỂ LẤY MÃ NGUỒN?

Hầu hết các dự án nguồn mở cung cấp quyền truy cập vào mã nguồn thông qua CVS hay SVN.

- ❑ **CVS (anonymous)**
 - ❖ `cvs -d:pserver:anonymous@demo.cvs.sourceforge.net:/cvsroot/ demo login`
 - ❖ `cvs -z3 -d:pserver:anonymous@demo.cvs.sourceforge.net:/cvsroot/demo co -P modulename`
- ❑ **SVN**
 - ❖ `svn co https://demo.svn.sourceforge.net/svnroot/demo`

XEM KỸ ANT

- ❖ Apache Ant là tương tự make, nhưng được viết bằng Java và sử dụng xml xây dựng tập tin cấu hình, nó phù hợp nhất với các dự án Java.

```
<?xml version="1.0"?>
<project name="Hello" default="compile">
  <target name="clean" description="remove intermediate files">
    <delete dir="classes"/>
    <delete dir="dist"/>
    <delete file="hello.jar"/>
  </target>
  <target name="compile" description="compile the Java source code">
    <mkdir dir="classes"/>
    <javac srcdir="src" destdir="classes"/>
  </target>
  <target name="jar" depends="compile" description="create a Jar file">
    <mkdir dir="dist"/>
    <jar destfile="dist/hello.jar">
      <fileset dir="classes" includes="classes/**/*.class"/>
      <manifest>
        <attribute name="Main-Class" value="HelloProgram"/>
      </manifest>
    </jar>
  </target>
</project>
```

MAVEN

- ❖ Maven là một phần mềm công cụ cho dự án Java cao cấp về quản lý và tự động xây dựng, dùng một XML Project Object Model để mô tả sự phụ thuộc vào mô-đun và thành phần mở rộng, và thứ tự xây dựng

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>Maven Quick Start Archetype</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

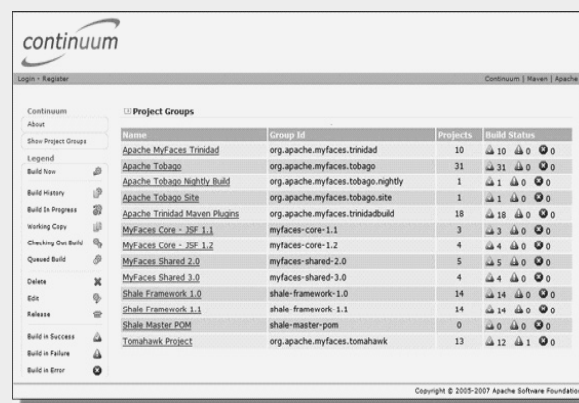
XÂY DỰNG TRONG ECLIPSE IDE

- ❑ Eclipse là một IDE (môi trường phát triển tích hợp) phổ biến cho các dự án phát triển nguồn mở
- ❑ Eclipse xây dựng tiến trình cho một dự án phụ thuộc vào project wizard (và plugins) sử dụng để tạo ra dự án.
- ❑ Biên dịch thường được thực hiện trong các nền tảng cho phép nâng cao hiệu lực của mã nguồn
- ❑ Eclipse là tương thích với ANT và MAVEN xây dựng hệ thống thông qua đầy đủ plugins.
- ❑ Thông tin dự án được lưu trữ trong một tập tin XML projectDescription file (.project)

XÂY DỰNG VỚI CONTINUUM

Máy chủ tích hợp liên tục để xây dựng các dự án Java-based

- ❖ Xây dựng theo yêu cầu và được lập kế hoạch thực hiện trên kho SVN/CVS.
- ❖ Hỗ trợ MAVEN, ANT and Shell Scripts
- ❖ Gửi thông báo qua điện thư, Jabber và Google Talk, MSN, IRC, ..



The screenshot shows the Continuum web interface with a table of project groups. The table has columns for Name, Group Id, Projects, and Build Status. The build status is represented by a series of icons: a green circle for success, a red circle with an 'X' for failure, and a yellow circle with a lightning bolt for in progress.

Name	Group Id	Projects	Build Status
Apache MyFaces Trinidad	org.apache.myfaces.trinidad	10	10 Success
Apache Tobago	org.apache.myfaces.tobago	31	31 Success
Apache Tobago Nightly Build	org.apache.myfaces.tobago.nightly	1	1 Success
Apache Tobago Site	org.apache.myfaces.tobago.site	1	1 Success
Apache Trinidad Maven Plugins	org.apache.myfaces.trinidadbuild	18	18 Success
MyFaces Core - JSP 1.1	myfaces-core-1.1	3	3 Success
MyFaces Core - JSP 1.2	myfaces-core-1.2	4	4 Success
MyFaces Shared 2.0	myfaces-shared-2.0	5	5 Success
MyFaces Shared 3.0	myfaces-shared-3.0	4	4 Success
Shale Framework 1.0	shale-framework-1.0	14	14 Success
Shale Framework 1.1	shale-framework-1.1	14	14 Success
Shale Master POM	shale-master-pom	0	0 Success
Tomahawk Project	org.apache.myfaces.tomahawk	13	12 Success, 1 In Progress

TÓM TẮT

- ☐ Các dự án mã nguồn mở đang sử dụng những phương pháp tiêu chuẩn để tự động hoá việc sản xuất phần mềm từ mã nguồn:
- ☐ Biên dịch, đóng gói, kiểm tra, tài liệu, triển khai...
- ☐ Cách tiếp cận này tạo thuận lợi cho việc sửa đổi, bổ sung của phần mềm và đồng nhất các phần phụ thuộc.
- ☐ Hầu hết các phần mềm hiện nay dựa vào phần mềm của bên thứ ba, lấy từ nguồn nào là vấn đề chiến lược và lựa chọn.

KIỂM SOÁT PHIÊN BẢN MÃ NGUỒN MỞ VỚI SUBVERSION

SOẠN BỞI: CHRISTOPHE NEY
CHO "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
PHIÊN BẢN / NGÀY : FOSS-BRIDGE BLOCK 32 /2008/06/30

BAO GỒM :

- Quản lý mã nguồn mở cho những người lập trình là gì ?
- Làm thế nào để có thể quản lý một cách hiệu quả mã nguồn mở thuộc nhóm ba,
- Những công cụ gì tương thích với mã nguồn mở,
- Phiên bản Subversion có thể hỗ trợ như thế nào cho nhiệm vụ này.

Bây giờ chúng ta sẽ học ...

- ☐ **Thực tế của việc quản lý mã nguồn mở cho những người lập trình là gì ?**
- ☐ **Làm thế nào để quản lý một cách hiệu quả mã nguồn mở thuộc nhóm ba**
- ☐ **Những công cụ gì tương thích với mã nguồn mở**
- ☐ **Phiên bản subversion có thể hỗ trợ như thế nào cho nhiệm vụ này**

QUẢN LÝ MÃ NGUỒN MỞ TRONG THỰC TẾ CUỘC SỐNG

Một khi bạn đã đi theo hướng nguồn mở, bạn sẽ phải quản lý nhiều dự án khác nhau về kích thước mã và có rất nhiều mã có thể thay đổi nhanh chóng

❑ Như vậy thì bạn có thể muốn:

- ❖ Lấy tiêu chuẩn cụ thể để tìm URL từ các phần mềm, license sử dụng, các gói mà bạn đang dùng
- ❖ Dễ dàng cho quá trình thay đổi phần mềm thuộc nhóm ba, áp dụng lại các thay đổi của bạn cho phiên bản mới, chia sẻ các thay đổi của bạn cho cộng đồng
- ❖ Để tổ chức và kiểm tra tài liệu hướng dẫn xung quanh phần mềm của bên thứ ba (đó là tích hợp)

HIỆU QUẢ QUẢN LÝ MÃ THUỘC NHÓM BA

Làm việc trực tiếp trên một kho các thành phần mã nguồn của bên thứ ba không phải là một tùy chọn.

- ❑ Như một người tích hợp những vấn đề của bạn sẽ không giống với thành phần của nhà cung cấp (họ có thể là một douzains bất đồng bộ)
- ❑ Đó là khả năng nhận được gần một bản phát hành cuối cùng, bạn sẽ phải có để sửa lỗi và ngắt mã của bên thứ ba.

GIẢI PHÁP LÀ ĐỂ DUY TRÌ KHO MÃ NGUỒN CỦA BẠN

- ☐ Nơi “nhập” tất cả các mã nguồn mà bạn muốn dưới quyền kiểm soát của bạn
- ☐ Nơi mà bạn chứng minh bằng tài liệu về sự phụ thuộc
- ☐ Cung cấp các tiện ích để xây dựng lại thành phần của bên thứ ba và tập hợp lại phần mềm của bạn trong vài phút
- ☐ Vẫn tiếp tục theo dõi lịch sử của tất cả phần mềm của bạn và cung cấp nhiều phương tiện để quản lý các phiên bản của nó
- ☐ Đó là những chia sẻ giữa tất cả các nhóm

CÓ MỘT SỐ CÔNG CỤ SCM

- ☐ **Generation 1**
 - ❖ RCS
- ☐ **Generation 2**
 - ❖ CVS, Subversion
- ☐ **Generation 3**
 - ❖ GIT, Bazaar

SUBVERSION (SVN) - HIỆU QUẢ CỦA PHIÊN BẢN KIỂM SOÁT HỆ THỐNG

- ❑ Nơi lưu trữ nguồn có thể là file nội bộ hoặc từ xa. Có thể lấy được thông qua `http/https(webdav)` khi được kết nối tới `apache`.
- ❑ Có nhiều loại của Client có sẵn như: `shell`, `GUIs`, `web`, in `IDEs`
- ❑ Nơi lưu trữ nguồn có thể được lưu bởi `Webserver`
- ❑ Thay đổi là hạt nhân (tất cả hoặc không có gì)
- ❑ Subversion hỗ trợ: gắn thêm vào một số thuộc tính, gắn thẻ và phân nhánh, hợp nhất

LÀM VIỆC TRÊN SUBVERSION

- ❑ **Một nhóm hai người**
 - ❖ Ở trong a shell console, kiểm tra rằng `svn` đã được cài đặt
 - ❖ Tạo repository tại root of the machine
 - `svn admin create /repository`
 - ❖ in `$home`, checkout the repository
 - `svn co file:///repository && cd repository`
 - ❖ Bắt đầu chạy với subversion
 - Add a file, make change, commit change, rollback
 - If needed : a demo is available on the wiki
 - Quick Help : `svn help`
- ❑ **Thời gian : 15 phút**

SUBVERSION ĐỂ TRUY XUẤT PHIÊN BẢN TRƯỚC

❑ Mỗi một lần bạn cam kết repository tăng số revision

```
>svn commit -m "added a message in hello" Hello.txt
```

```
Sending      Hello.txt
```

```
Transmitting file data .
```

```
Committed revision 3.
```

❑ Nhưng bạn có thể tạo một bản sao của revision đặc biệt sử dụng câu lệnh `svn copy`

- ❖ Repository thường được phân chia thành 3 subdirectory (trunk, tags, branches), và có thể sao chép thông qua revision mà chúng ta muốn gán

```
>svn copy trunk tags/REV4
```

SUBVERSION ĐỂ LÀM VIỆC SONG SONG

❑ Khi thành viên trong nhóm làm việc trong thư mục riêng biệt, hoặc trên các tập tin khác nhau, subversion quan tâm đến tính hợp nhất.

❑ Branches là cần thiết.

- ❖ Khi bạn hay một số thành viên của nhóm phải làm nhiều công việc mà ảnh hưởng đến mã ở mọi nơi và dành thời gian để được thực hiện
- ❖ Branches được tạo ra như các thẻ, nhưng không giống như thẻ tiến triển theo thời gian và được hợp nhất lại cho trunk tại một số điểm

SUBVERSION TAGS AND BRANCHES

Lab Activity

❑ Một nhóm hai người

- ❖ Truy cập trang <http://svnbook.red-bean.com>
- ❖ Nghiên cứu chương 4. "Branching and Merging" của tài liệu subversion
- ❖ Trong a shell console, thực hành theo tài liệu yêu cầu trên SVN repository của bạn

❑ Thời gian : 30 phút

SUBVERSION ĐỂ QUẢN LÝ SỰ TIẾN TRIỂN PHẦN MỀM NHÓM BA

Một thách thức tích hợp chung là để quản lý các thay đổi mà đang diễn ra ở cả bên Vendor (black box) và trong kho của bạn (repository).

❑ Subverion có thể xử lý như sau:

- ❖ Mã liên quan của nhóm ba được nhập vào dưới một branch "vendor" đặc biệt
- ❖ Mã của phiên bản hiện tại mô phỏng mã sự tiến triển của vendor.
- ❖ Mã sửa đổi bản sao của vendor duy trì dưới trunk được kết hợp với sự thay đổi của vendor khi cần thiết

SUBVERSION VENDOR BRANCHES

Demo

- ❑ **Một nhóm hai người**
 - ❖ Đến trang wiki và xem ba bản demo trên mã quản lý nhóm ba với Eclipse and Subversion
- ❑ **Thời gian : 20 phút**

TÓM TẮT

- ❑ **Quản lý mã nguồn hệ điều hành là một khía cạnh quan trọng của sự phát triển cơ bản nguồn mở.**
- ❑ **Hệ thống SCM như là một Subversion nhưng phải có:**
 - ❖ Thực hiện mã tích hợp
 - ❖ Tiếp tục theo dõi lịch sử của phần mềm
 - ❖ Thực hiện phát triển đồng thời
 - ❖ Quản lý mã nhóm ba.
- ❑ **Nhưng kinh nghiệm và đào tạo cũng là then chốt trong sự tiến triển của mã hệ điều hành**

SOME IMPORTANT LESSONS SLIDES IN ENGLISH

ABOUT:

"FOSS Bridge EU-Vietnam - Joint Business through Free and Open Source Software" is an ASIA-Invest programme implemented by InWEnt - Capacity Building International of Germany, Institut National de Recherche en Informatique et en Automatique (INRIA) of France and the Institute of Information Technology of Vietnam (IOIT) with financial support by the European Commission and the German Federal Ministry of Economic Development and Cooperation (BMZ). For further information please visit the following websites: <http://www.foss-bridge.org>.



LICENSE:

Creative Commons Attribution-Share Alike 3.0 Germany Copyright: InWEnt About the Creative Commons Attribution-Share Alike 3.0 Germany License: The material is licensed under a Creative Commons Attribution-Share Alike 3.0 Germany License. You are free to copy, distribute, transmit and adapt the work. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/de/deed.en> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA. Please note that this license requires attribution of all authors in future versions, and should include the following attribution: "Based on training material created by Christophe NEY for the project "FOSS Bridge EU-Vietnam - Joint Business through Free and Open Source Software", InWEnt. Reference number of training material: [TRM1702400700]. For more information see <http://www.foss-bridge.org>. Under the license, the copyright holder (InWEnt) does not endorse any future versions of the material or the use of the work."

PLEASE ALSO NOTE THE FOLLOWING INDICATIONS AND DISCLAIMERS:

All trademarks mentioned in this document and potentially protected by third parties shall be subject to the unqualified provisions of the pertinent trademark law and property rights of the registered owner. - Under the license, the copyright holder (InWEnt) does not endorse any future versions of the material or the use of the work.- This document has been produced with the financial assistance of the European Commission and the German Federal Ministry of Economic Development and Cooperation (BMZ).. The contents of this document are the sole responsibility of the authors and can under no circumstances be regarded as reflecting the position of the European Commission and the German Federal Ministry of Economic Development and Cooperation (BMZ).

I - In addition to the attribution under the license, InWEnt would kindly request a brief notice in case of use of the material indicating context of use/ modification and number of people reached. Please give us a feedback at [ict\[at\]inwent.org](mailto:ict[at]inwent.org).

INTELLECTUAL PROPERTY AND LICENSES

CREATED BY: FRANÇOIS LETELLIER
FOR "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
VERSION / DATE : FOSS-BRIDGE BLOCK3 /2008/04/17

Created during the project "FOSS Bridge EU-Vietnam - Joint Business through Free and Open Source Software" (<http://www.foss-bridge.org>) a program of InWEnt - Capacity Building International of Germany, Institut National de Recherche en Informatique et en Automatique (INRIA) of France and the Institute of Information Technology of Vietnam (IOIT) with financial support by the European Commission and the German Federal Ministry of Economic Development and Cooperation (BMZ). For more information see <http://www.foss-bridge.org>.

Reference number of training material: [TRM1702400700].

Under the license, the copyright holder (InWEnt) does not endorse any future versions of the material or the use of the work. See also notes on last page of this document.

WHAT WE'LL GONNA LEARN NOW...

- ☐ **General regulatory framework of software protection**
- ☐ **International and national regulations**
- ☐ **« Intellectual property »patents, trademarks, copyright**
- ☐ **Reminder on proprietary licenses**

MASTER PLAN OF OSS IN VIET NAM

(2 March 2004)

❑ **Applying and Developing Open Source Software in Vietnam for the 2004-2008 period**

- ❖ Accelerating the application and development of OSS
- ❖ Enhancing copyrights protection
- ❖ Cutting costs of software purchase
- ❖ Promoting the development of Vietnam's information technology
- ❖ Forming a base of competent technical experts
- ❖ Creating some typical IT products

SOFTWARE PROTECTION:LEGAL FRAMEWORK

❑ **No such thing as an « International Law »**

- ❖ National laws
- ❖ Treaties between countries
- ❖ US Law ≠ « International » law

❑ **What's different from country to country?**

- ❖ Possibly everything!
- ❖ Language
- ❖ Mandatory elements (e.g. in contracts)
- ❖ Disclaimers of warranty, limitations of liability
- ❖ Author's rights -> Copyright (e.g. moral rights)
- ❖ Software patentability

SOFTWARE PROTECTION:

LEGAL FRAMEWORK - PATENTS

- ❑ **Ordinance on Protection of Industrial Property Rights (28 January 1989)**
- ❑ **ASEAN Framework Agreement on Intellectual Property Cooperation (1996) - WTO GATT/Trips treaty; section 5, art. 27: « patents shall be available for any inventions »**
- ❑ **Excluded from patentability (in VN):**
 - ❖ **Computer software**, integrated circuits, mathematical models, graphs for searching and the like
 - ❖ Scientific ideas, theories, and discoveries; methods for economic organizations and management, education, teaching, training; language and information systems, conventional signs, time schedule, rules and regulations, symbols; etc...
- ❑ **Software can be protected by patents in some countries (e.g. USA)**
 - ❖ Unlike copyright: independently created inventions still require a license from any valid patent before use

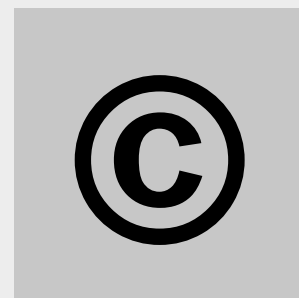
SOFTWARE PROTECTION: LEGAL FRAMEWORK - TRADEMARKS

- ❑ **Word, phrase, symbol, device, or any combination, used to distinguish the goods / services / membership of one person**
- ❑ **Registration (first to file principle) mandatory – except for well-known trademarks**
- ❑ **Right to prevent all persons not having the owner's consent from:**
 - ❖ Using identical or similar signs...
 - ❖ For identical, similar, or related goods/services where such use would result in a likelihood of confusion
- ❑ **Unlawful use is an infringement**



SOFTWARE PROTECTION:LEGAL FRAMEWORK

- ❑ **Computer software**
 - ❖ Considered part of « literary, artistic or scientific works »
 - ❖ Excluded from patentability
 - ❖ Protected by copyright
 - ❖ In Viet Nam: limited public awareness, low penalties, little judicial enforcement



Copyright symbol

SOFTWARE PROTECTION: LEGAL FRAMEWORK

❑ **Copyright / Author's rights: the Berne Convention**

- ❖ Protection of Literary and Artistic Works
- ❖ Vietnam signatory of the Berne Convention since 2004
- ❖ Principle of national treatment
- ❖ Protection over a work is automatic, not subject to registration
 - Though registration may be mandatory for litigation in some countries (e.g. USA)
 - Or helpful for proof of anteriority
- ❖ The work must be original and creative

SOFTWARE PROTECTION: LEGAL FRAMEWORK

❑ **Exclusive rights to the owner**

- ❖ Moral rights: attribution, integrity
- ❖ Personal rights: publish, copy, perform, display, distribute, make derivative works
- ❖ Property rights: royalties, benefits, awards

❑ **Rights are transferable, separately or together (limitations in some countries)**

❑ **Copyright infringements**

- ❖ substantial similarity and access to infringed work
- ❖ unlawful use

SOFTWARE PROTECTION: LEGAL FRAMEWORK

- ❑ **To use copyrighted work (e.g. Software):**
 - ❖ Must enter a contract (« license ») with the author or copyright owner
 - No right granted unless explicitly specified
 - No obligation of the licensee to accept
 - ❖ Must include provisions for:
 - How software is used
 - Scope and duration
 - Royalty fee (if any)
 - Liability of the parties in the event of a breach
- ❑ **Use of published copyrighted works for « non commercial purposes » not applicable to software (VN)**

SOFTWARE PROTECTION: THE PUBLIC DOMAIN

- ❑ **Status of software (or other work) over which no one retains any legal right**
 - ❖ By choice (« donation » to the public domain)
 - ❖ After the end of its protected term (e.g. ancient literary works)
 - ❖ By law (e.g. data produced by the US public administration)
- ❑ **Any use free and legal by anyone for any (legal) purpose**

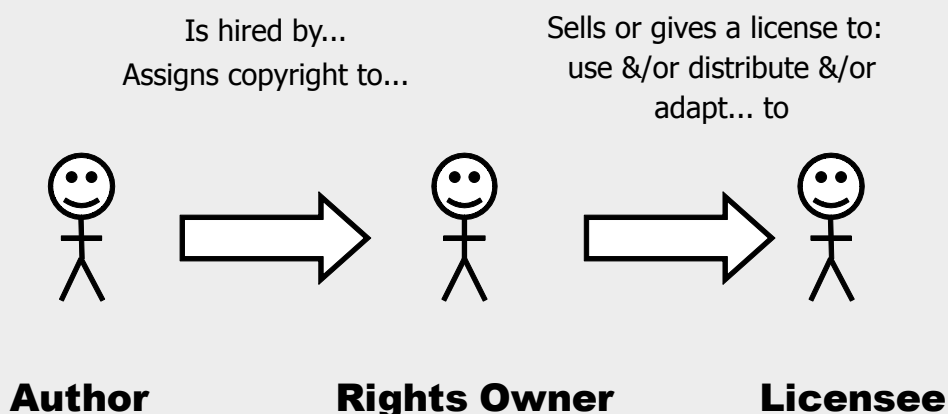
SOFTWARE PROTECTION: LEGAL FRAMEWORK

❑ Who's the « author »?

- ❖ The person directly creating the whole or part of a literary, artistic, scientific work
- ❖ A person who translates a work
- ❖ A person who adapts an existing work, re-writes or transforms a work
- ❖ A person who edits, annotates or selects works

❑ If more than one: co-authors

SOFTWARE LICENSES OVERALL SCHEME



❑ Examples:

Software engineer

Company (employer)

end user

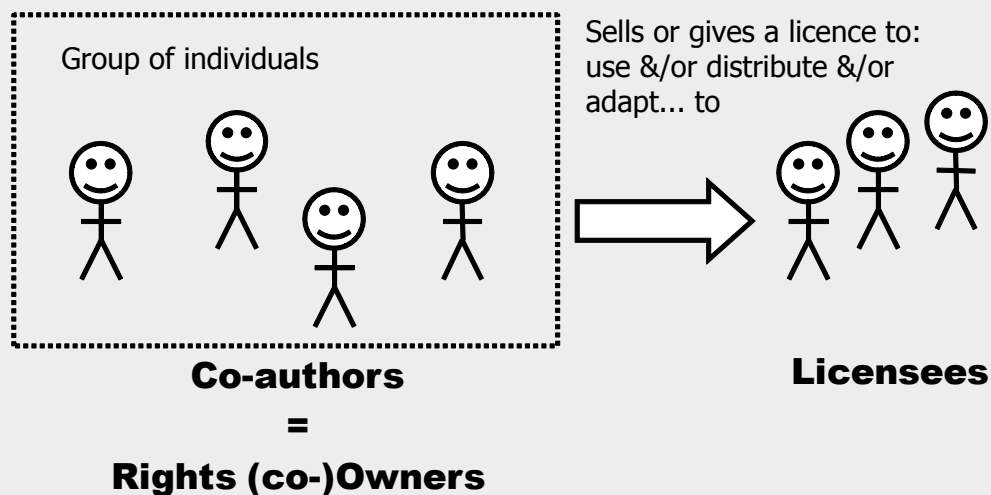
SOFTWARE PROTECTION: LEGAL FRAMEWORK

❑ Who's the legal « owner » of the rights?

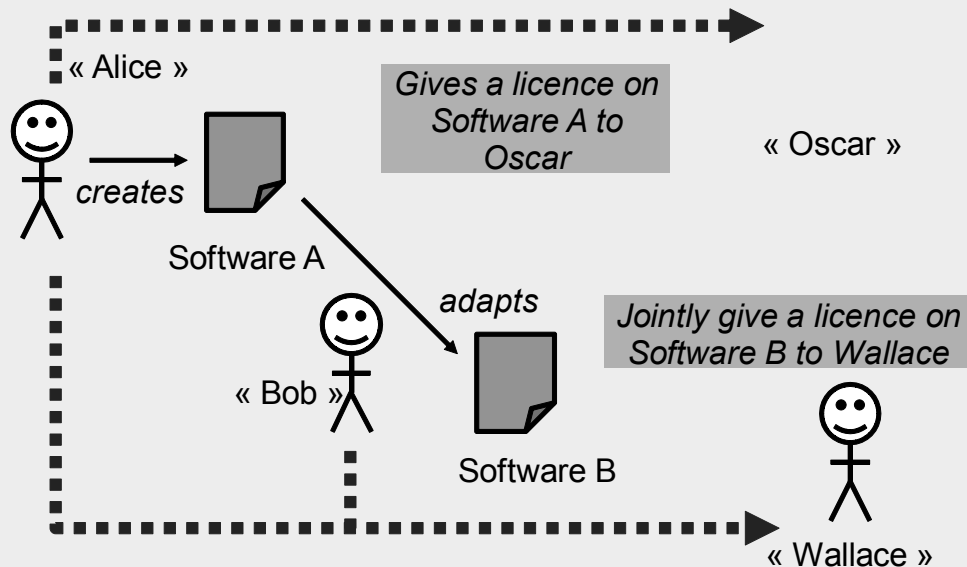
- ❖ By default: the author – or co-authors
- ❖ An authority or organization which delegates a duty to an author or enters into a contract for the creation of the work
- ❖ Possible exceptions (e.g. moral rights)
- ❖ A testamentary heir or an heir at law of the author
- ❖ The assignee who is assigned work from the owner of the work

SOFTWARE LICENSES

WHEN CO-AUTHORS = INDIVIDUALS



SOFTWARE LICENSES THE CHAIN OF TITLE



RECAP

- ☐ **Software is mainly protected (legal excludability) under copyright law**
- ☐ **There's nothing as an international law, so there's complexity in international contexts**
- ☐ **(Co)-authors ≠ owner of the rights**
- ☐ **Except for work in the public domain, no right is granted unless a license is given**

FLOSS HISTORY, BACKGROUND, CULTURE

CREATED BY: FRANÇOIS LETELLIER
FOR "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
VERSION / DATE : FOSS-BRIDGE BLOCK4 /2008/04/17

WHAT WE'LL LEARN NOW...

- ❑ **FLOSS history, background, culture**
- ❑ **Who's who - opinion leaders**

BILL GATES

(EVERY HERO NEEDS A VILAIN)

- ❑ Co-founded Microsoft in 1975
- ❑ Fierce proponent of Intellectual Property
- ❑ « Who can afford to do professional work for nothing? »
- ❑ 2007: richest man in the world (\$59B, Forbes)



RICHARD M. STALLMAN

- ❑ MIT 1971-1984
- ❑ « Free speech, not free beer »
- ❑ Founded the FSF (1984)
- ❑ Incepted the development of GNU (Emacs)
- ❑ Created the first copylefted licenses (GPL)



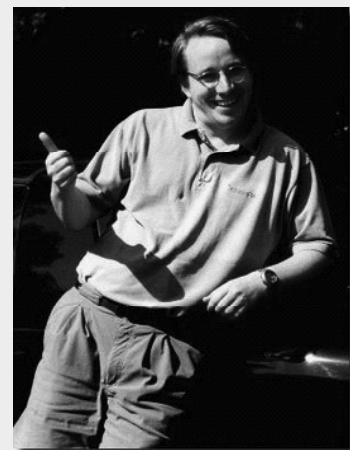
ERIC S. RAYMOND

- ❑ **Author of « The Cathedral and the Bazaar »**
- ❑ **« Shut up and show them the code »**
- ❑ **Founded the Open Source Initiative (1998)**



LINUS TORVALDS

- ❑ **Lunis -> « Linux »**
- ❑ **Famous for bringing the Linux kernel to GNU**
- ❑ **Hired by OSDL to lead the development of the Linux kernel**



BRUCE PERENS

- ❑ **Primary author of the « Open Source Definition »**
- ❑ **Open Source Evangelist**
- ❑ **Also works to promote open standards**



LAWRENCE LESSIG

- ❑ **Professor of Law at Stanford University**
- ❑ **Author of « Code and other Laws of the Cyberspace »**
- ❑ **Founded the Creative Commons**



RECAP

- ❑ **FLOSS history originates back to the early days of computers**
- ❑ **The « free software » movement appeared as a reaction to the emerging trend of commercial software**
- ❑ **Free vs open source: philosophy is an important aspect of the movement**
- ❑ **Opinionated and charismatic leaders**

FREE AND OPEN-SOURCE LICENSES

CREATED BY: FRANÇOIS LETELLIER
FOR "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
VERSION / DATE : FOSS-BRIDGE BLOCK5 /2008/04/17

WHAT WE'LL LEARN NOW...

- ☐ **What's « free software »**
- ☐ **What's « open source software »**
- ☐ **What's « copyleft » and what it's not**
- ☐ **What's neither « free » nor « open source » software**

GRATIS SOFTWARE

- ❑ **Freeware**
 - ❖ Gratis, close source
- ❑ **Shareware**
 - ❖ Payment is optional, close source
- ❑ **Demo version**
 - ❖ Gratis for some time, then user must pay
- ❑ **Free Software, Open Source Software**
 - ❖ Does not mean gratis – even though often gratis
 - ❖ Does not mean « public domain » - protection and licensing apply

FREE SOFTWARE - DEFINITION

- ❑ **Free as in « Free Speech » not « Free Beer »**
 - ❖ Four « freedoms » (source: Free Software Foundation)
 - ❖ 0 - to run the program, for any purpose
 - ❖ 1 - to study how the program works, and adapt it to your needs
 - ❖ 2 - to redistribute copies so you can help your neighbor
 - ❖ 3 - to improve the program, and release your improvements to the public, so that the whole community benefits
- ❑ **Access to the source code is a precondition for this freedoms 1 and 3**

OPEN SOURCE SOFTWARE – DEFINITION (1/3)

Definition of « open source software »

(source: Open Source Initiative)

- ❑ **1. Free Redistribution**
 - ❖ including as component of other software
- ❑ **2. Source Code**
 - ❖ included or available
- ❑ **3. Derived Works**
 - ❖ modifications can be licensed the same

OPEN SOURCE SOFTWARE – DEFINITION (2/3)

Definition of « open source software »

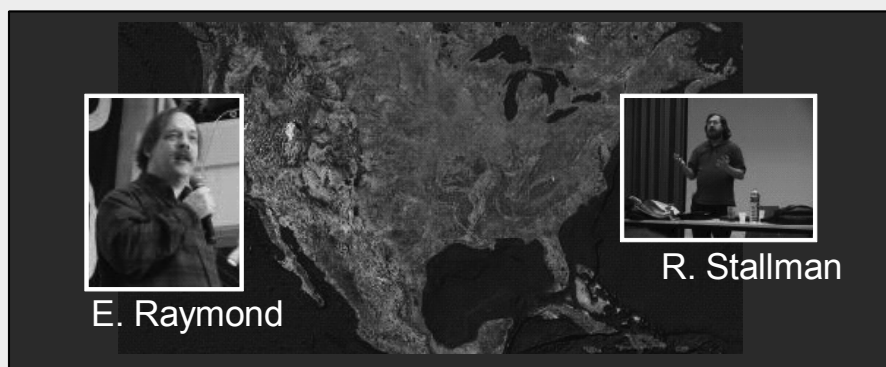
(source: Open Source Initiative)

- ❑ **4. Integrity of The Author's Source Code**
 - ❖ May distribute derivative works, but might need name change or use patch files
- ❑ **5. No Discrimination Against Persons or Groups**
 - ❖ nor can you give one company extra rights
- ❑ **6. No Discrimination Against Fields of Endeavor**
 - ❖ cannot forbid use in particular contexts

OPEN SOURCE SOFTWARE – DEFINITION (3/3)

- ❑ **7. Distribution of licence**
 - ❖ no other required license
- ❑ **8. License Must Not Be Specific to a Product**
 - ❖ same rights if extracted and used elsewhere
- ❑ **9. License Must Not Restrict Other Software**
 - ❖ if simply distributed on same medium
- ❑ **10. License Must Be Technology-Neutral**
 - ❖ cannot predicate technology or style of interface

CULTURE DIFFERENT: FREE VS OPEN SOURCE



- ❑ **Ethical (FSF) vs pragmatic / « business friendly » (OSI)**
- ❑ **Acronym FLOSS: Free / Libre / Open Source Software**
- ❑ **« close source » or « proprietary » for short = non-FLOSS**
- ❑ **FSF creates its own licenses, OSI validates submitted licenses**

SOUNDING LIKE FLOSS BUT NOT FLOSS

- ❑ **Gratis ≠ FLOSS**
- ❑ **Access to the source code does not mean FLOSS**
 - ❖ e.g.: bespoke software; « Shared Source » (Microsoft)
- ❑ **Freely redistributable does not mean FLOSS**
- ❑ **All FLOSS rights but restricted (e.g. to non commercial uses) does not mean FLOSS**
 - ❖ e.g.: Scilab

LICENSE « PROLIFERATION »

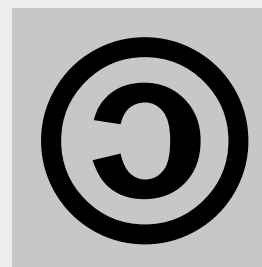
- ❑ **1000 (?) licenses compliant to the F/OSS definition (Saugatuck 07)**
 - ❖ Various philosophies / business models
 - ❖ International context
- ❑ **61 OSI-approved (APR 08)**
- ❑ **Drawbacks**
 - ❖ Complexity for the user
 - ❖ Possible incompatibilities
- ❑ **OSI answer: categorize**

POPULAR AND WIDELY USED OR WITH STRONG COMMUNITIES

- ☐ **Apache License, 2.0**
- ☐ **New BSD license**
- ☐ **GNU General Public License (GPL version 2)**
- ☐ **GNU Library or "Lesser" General Public License (LGPL version 2)**
- ☐ **MIT license**
- ☐ **Mozilla Public License 1.1 (MPL)**
- ☐ **Common Development and Distribution Licence (CDDL)**
- ☐ **Common Public Licence (CPL)**
- ☐ **Eclipse Public Licence (EPL)**

« COPYLEFT »

- ☐ **Based on copy-right**
- ☐ **The right owner grants rights to**
 - ❖ Use, adapt, redistribute
- ☐ **Under some « reciprocal » conditions:**
 - ❖ In case of distribution...
 - ❖ ...the same rights must be propagated
 - ❖ Depending on the licenses, this obligation also applies to contributions or not
- ☐ **Copylefted licenses exist for software, music, art, ...**



MISCONCEPTIONS ABOUT COPYLEFT

The following common beliefs are not true:

- ☐ **Obligation to distribute derivative works**
- ☐ **Obligation to make derivative works publicly accessible**
- ☐ **« Contamination » of the Information System**
 - ❖ FUD terms: « virality »
 - ❖ Confusion between aggregate and derivative work
 - ❖ Confusion between development and execution
- ☐ **Incompatibility with a commercial activity**

COPYLEFTED FLOSS LICENSES AKA

« RECIPROCAL » LICENSES

- ☐ **GPL – GNU Public licence**
 - ❖ v1 (1989), v2 (1991), v3 (2007)
 - ❖ E.g.: Linux kernel, PERL, MySQL
- ☐ **LGPL – Lesser GNU Public licence**
 - ❖ Designed for libraries
 - ❖ E.g.: glibc, JBoss AS
- ☐ **CPL – Common Public licence**
 - ❖ E.g.: Microsoft Windows Installer XML
- ☐ **EPL – Eclipse Public licence**
 - ❖ E.g.: Eclipse platform

NON-COPYLEFTED LICENSE AKA «ACADEMIC» LICENSES

- ☐ **Inspired of the « academic freedom »**
- ☐ **Historically the first: BSD (Berkeley, < 1980)**
- ☐ **≈ « The copyright owner of this software hereby licenses it to you for any purpose whatsoever »**
- ☐ **Usually, Warranty and Liability disclaimers**
- ☐ **(New) BSD, MIT, Apache 1.0, Artistic licenses**

RECAP

- ☐ **FLOSS ≠ gratis software, even though usually available for free**
- ☐ **Free software = FSF definition**
- ☐ **Open Source = OSI criteria**
- ☐ **Distribution of FLOSS is not an obligation**
- ☐ **« copyleft » is a reciprocal principle**
- ☐ **Some FLOSS licenses are not copylefted**

DETAILS OF FLOSS LICENSES

CREATED BY: FRANÇOIS LETELLIER
FOR "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
VERSION / DATE : FOSS-BRIDGE BLOCK6 /2008/04/17

WHAT WE'LL LEARN NOW...

- ☐ **Study FLOSS licenses in details**
- ☐ **Consider issues about compatibility between licenses**
- ☐ **Discover free documentation licenses**

GNU GENERAL PUBLIC LICENSE 2

- ❑ **SReach: strong copyleft**
 - ❖ Teciprocality covers container/derivative works
- ❑ **Sublicense:**
 - ❖ No sublicense; license granted from original author upon each distribution
- ❑ **Patent Grant:**
 - ❖ No explicit patent grant, but license implies one
- ❑ **Retaliation:**
 - ❖ No distribution unless everyone can distribute royalty free

GNU LESSER GENERAL PUBLIC LICENSE 2.1

- ❑ **Reach: copyleft**
 - ❖ Covers containers/derivative works, only restricts (does not require reciprocity of) licenses
- ❑ **Sublicense:**
 - ❖ Same as GPL
- ❑ **Patent Grant:**
 - ❖ Same as GPL
- ❑ **Retaliation:**
 - ❖ Same as GPL

COMMON PUBLIC LICENSE 1.0

ECLIPSE PUBLIC LICENSE 1.0

❑ **Reach: copyleft**

- ❖ Reciprocity does not apply to modules distributed with the software which are not derivative works of the software

❑ **Sublicense:**

- ❖ May relicense object code if license includes notices and disclaimers

❑ **Patent Grant:**

- ❖ Explicit grant for contribution and combination

❑ **Retaliation: revocable non assertion**

- ❖ (in CPL only) patent litigation against a contributor with respect to a patent applicable to software - then any patent licenses granted by that contributor to the recipient shall terminate
- ❖ Patent litigation against any entity alleging that the program itself infringes the recipient's patent(s): then such Recipient's rights granted by the license shall terminate

APACHE LICENSE 2.0

- ❑ **Reach: academic**
 - ❖ Not reciprocal
- ❑ **Sublicense:**
 - ❖ May provide additional or different license terms and conditions of modifications
- ❑ **Patent Grant:**
 - ❖ Explicit grant for contribution and combination
- ❑ **Retaliation:**
 - ❖ Very similar to EPL

MOZILLA PUBLIC LICENSE 1.1

- ❑ **Reach:**
 - ❖ File-based reciprocity
- ❑ **Sublicense:**
 - ❖ Redistribute executable under own terms
- ❑ **Patent Grant:**
 - ❖ Explicit patent grant for “contributor version”
- ❑ **Retaliation:**
 - ❖ Lose contributor’s patent & copyright grants if assert claim against contributor for work
 - ❖ Lose contributor’s patent grants if assert patent claim against any “software, hardware, or device”

NEW BSD

- ☐ **Reach: academic**
 - ❖ Not reciprocal
- ☐ **Sublicense:**
 - ❖ No mention of sublicense, but often assumed
- ☐ **Patent Grant:**
 - ❖ No patent grant, just the word “use”
- ☐ **Retaliation:**
 - ❖ None

MIT

- ☐ **Reach: academic**
 - ❖ Same as BSD
- ☐ **Sublicense:**
 - ❖ Sublicensing is explicit, relicensing implicit
- ☐ **Patent Grant:**
 - ❖ Not explicit terms
- ☐ **Retaliation:**
 - ❖ Same as BSD

EUROPEAN UNION PUBLIC LICENSE

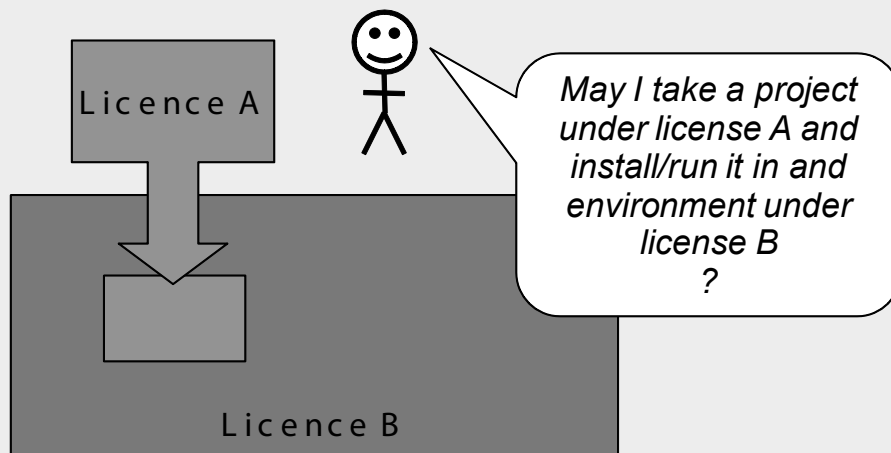
- ☐ **Proposed by IDABC, January 2007, approved by the European Commission**
- ☐ **Compliant with laws of all EU countries**
- ☐ **Versions in the 23 official languages of the EU (January 2008) of same legal value**
- ☐ **Copylefted, CPL-, EPL-, GPL-, OSL-, CeCILL- compatible**
- ☐ **Currently being reviewed by the OSI**

PATENT PROVISIONS IN FLOSS LICENSES

- ☐ **Reminder: software is patentable in some countries (USA)**
- ☐ **Main classes of patent provisions in FLOSS licenses**
 - ❖ None (e.g. New BSD, MIT)
 - ❖ Revocable non assertion (e.g. MPL, CPL, EPL, CDDL)
 - ❖ Perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (e.g. Apache Licence 2.0)
 - ❖ Restrictions on distribution in case of possible patent infringement (e.g. *GPL)
- ☐ **Remark: grant of patent licenses by the initial developer or contributors does not guarantee that users may not infringe on any patent claim from any third party in any country**

LICENCE (IN)COMPATIBILITIES (1/5)

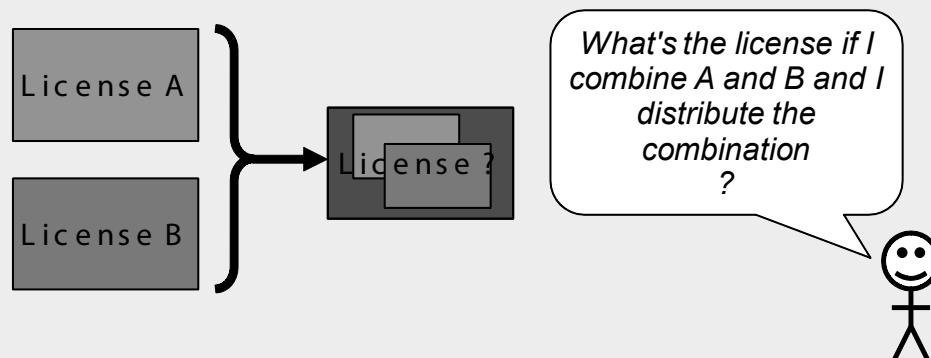
Inclusion of software



Matters for the end users of FLOSS

LICENCE (IN)COMPATIBILITIES (2/5)

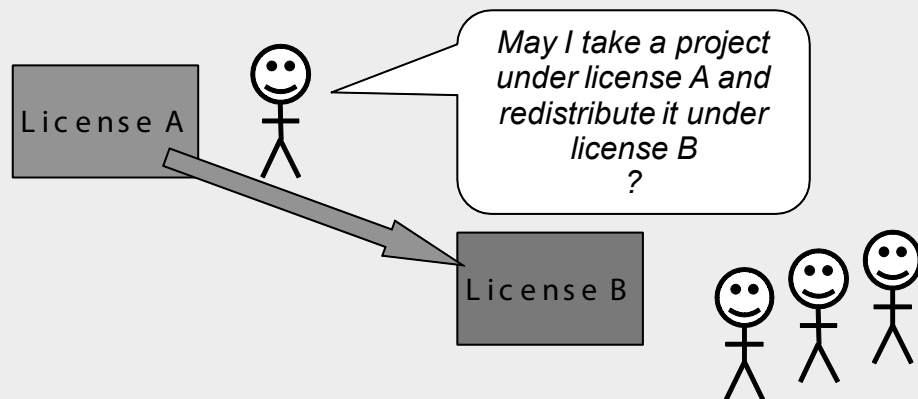
Combination of software under different licenses



Matters for the FLOSS development process

LICENCE (IN)COMPATIBILITIES (3/5)

Change of license for redistribution



Matters for setting up FLOSS based business models

LICENCE (IN)COMPATIBILITIES (4/5)

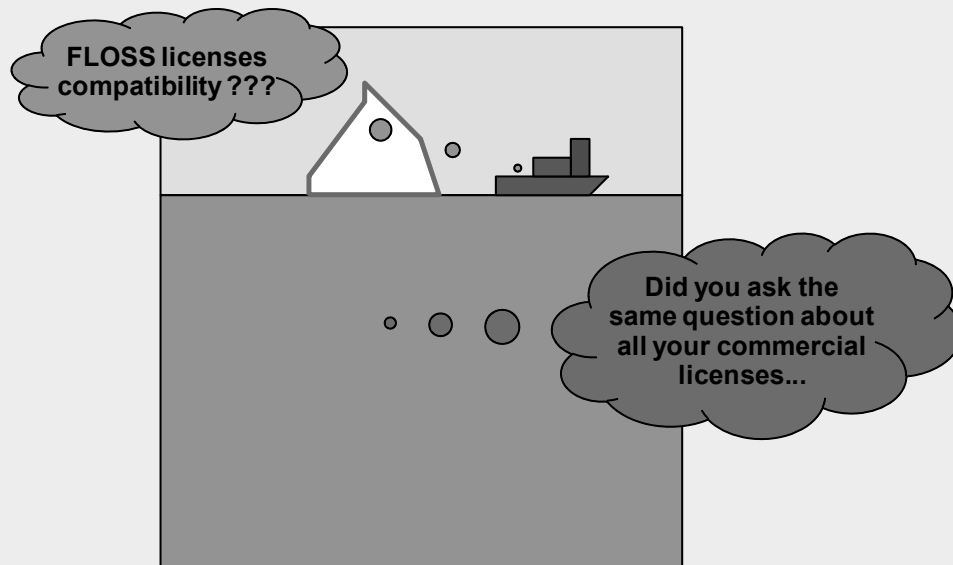
❑ No simple answer – case by case

- ❖ Detailed, legal analysis of licenses
- ❖ Countries, applicable law
- ❖ Parties involved (legal/physical, etc)

		DERIVATIVE WORK				
		GPL	MPL	CPL	OSL	Academic
CONTRIBUTION	BSD	yes	no ¹	no ²	yes	yes ³
	MIT	yes	no ¹	no ²	yes	yes ³
	Apache	yes ⁴	no ¹	no ²	yes	yes ³
	AFL	yes ⁴	no ¹	no ²	yes	yes ³

Source: « Open Source Licensing -Software Freedom and Intellectual Property Law » by Lawrence Rosen

LICENCE (IN)COMPATIBILITIES (5/5)



DOCUMENTATION LICENSES (1/3)

- ☐ **Documentation for conception is usually licensed with software**
- ☐ **Other kinds of documentation:**
 - ❖ Books, whitepapers, manuals, wikis, web sites, slideshows, artwork, multimedia content...
 - ❖ Protected under copyright law
 - ❖ Usually not licensed with software

DOCUMENTATION LICENSES (2/3)

❑ **GNU Free Documentation License (GFDL)**

- ❖ Created by the FSF
- ❖ Equivalent of a GPL for non-software works
- ❖ Copylefted
- ❖ Notion of « transparent copy » (= source code) and « opaque copy » (= executable code)

DOCUMENTATION LICENSES (3/3)

❑ **Creative Commons (CC)**

- ❖ Creator: Lawrence Lessig
- ❖ Roster of licenses crafted by lawyers to fit:
 - Various countries / languages
 - From « all rights reserved » to « some rights reserved »
 - Owner decides on the rights and language – and gets the matching legal text for the license

RECAP

- ❑ **Wide spectrum of license terms, from strong copyleft to very permissive « academic » licenses**
- ❑ **License compatibility issues appear when using, developing and making business on FLOSS - but also with commercial software**
- ❑ **FLOSS principles have been ported to documentation licenses too**

FLOSS COMMUNITIES

CREATED BY: FRANÇOIS LETELLIER
FOR "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
VERSION / DATE : FOSS-BRIDGE BLOCK9 /2008/04/17

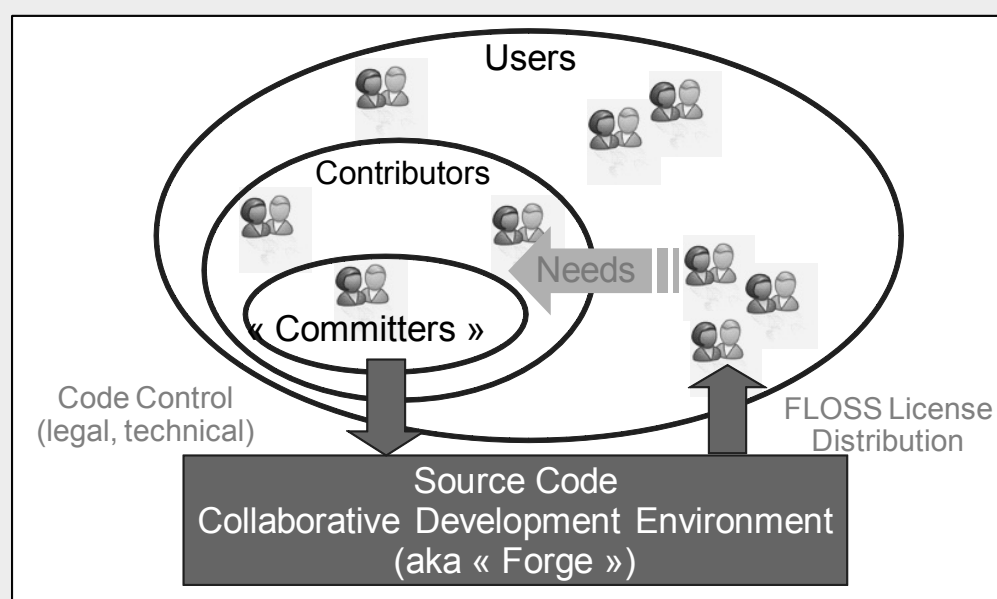
WHAT WE'LL LEARN NOW...

- ☐ **Typical FLOSS communities**
- ☐ **The hacker and FLOSS culture**
- ☐ **Profile of the community members**

THE FLOSS PROJECT

- ❑ **Project - the basic unit in FLOSS communities**
 - ❖ Collaborative development
 - ❖ Technology driven - mostly code production
- ❑ **Project leaders**
 - ❖ Often are at the origin of the project, oversee its major direction
 - ❖ Personality and charisma are key
- ❑ **Core members «(sometimes organised as PMC)**
 - ❖ Make significant contributions over time
 - ❖ Usually less than ≈ 15 p (over this limit, the project is likely to give birth to subprojects)
- ❑ **Active users (aka « contributors »)**
 - ❖ Report bugs, write documentation, occasionally patches

COLLABORATIVE ENGINEERINGCODE (1/3)



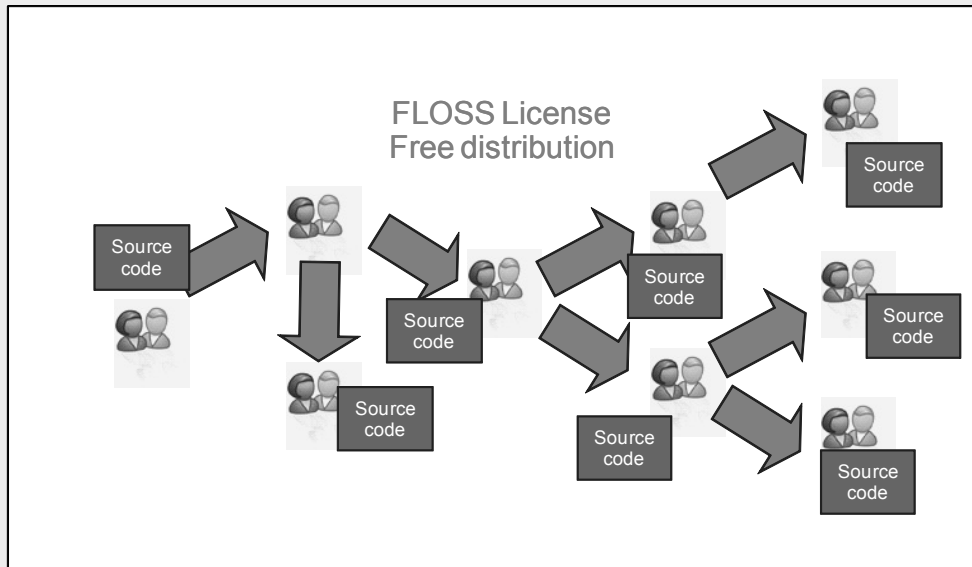
COLLABORATIVE ENGINEERINGCODE (2/3)

- ❑ **Mailing lists, forums - form a knowledge base**
- ❑ **Wikis - online documentation**
- ❑ **Source code management - so concurrent development can happen**
 - ❖ CVS, SVN
- ❑ **Bug tracker**
 - ❖ Bugzilla, TRAC
- ❑ **Administration tools, user management, stats, continuous integration, ...**

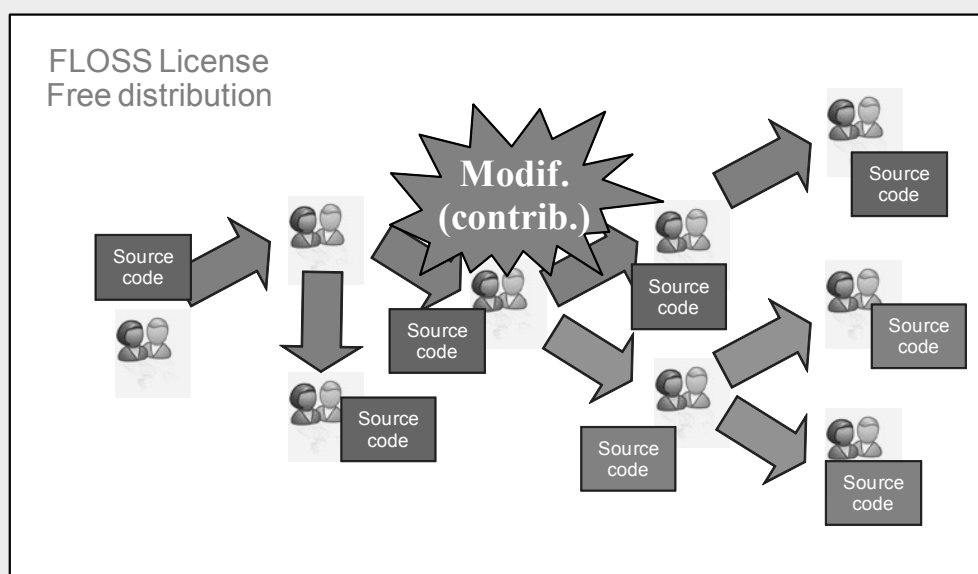
COLLABORATIVE ENGINEERINGCODE (3/3)

- ❑ **Warning: centralization of the code base is done for practical reasons**
- ❑ **The license does not require it (apart exceptions)**
- ❑ **As a consequence:**
 - ❖ Some projects' code base is not publicly accessible
 - ❖ Some projects have a distributed code base
 - ❖ Projects may fork...

PROJECT FORKING (1/3)



PROJECT FORKING (2/3)



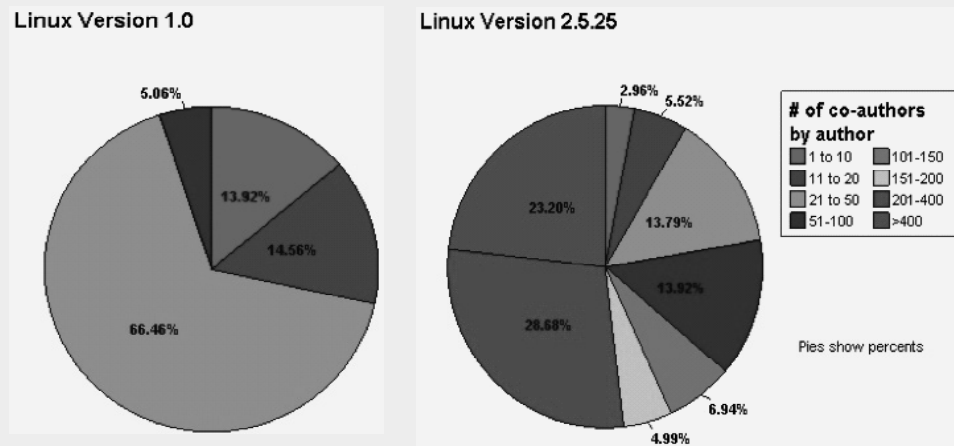
PROJECT FORKING (3/3)

- ❑ **When a project gives birth to two variants, which gradually diverge**
- ❑ **Various possible reasons**
 - ❖ Technical (eg: port on # platforms)
 - ❖ Human (eg: lead conflict)
 - ❖ Business (eg: starting a new commercial offer)
- ❑ **Forks are the best... and the worst thing**
- ❑ **« Survival of the fittest »**

FLOSS COMMUNITIES EVOLUTION

- ❑ **Projects evolve**
 - ❖ By adding new developers
 - ❖ Through migration of developers from other existing projects
- ❑ **The largest projects grow faster**
 - ❖ Hence the phenomenon of « critical mass »
- ❑ **New projects form around a core of connected developers**
- ❑ **Information flows along project dependencies**

COLLABORATION ON THE LINUX KERNEL



The nature and composition of the Linux kernel developer community: a dynamic analysis, MERIT/Institute of Informatics, 2003

❑ Co-authorship on single source code modules

FLOSS & THE HACKER CULTURE

❑ The hacker ethics (Steven Levy)

- ❖ Access to computers should be unlimited and total
- ❖ All information should be free
- ❖ Mistrust authority, promote decentralization
- ❖ Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position
- ❖ You can create art and beauty on a computer
- ❖ Computers can change your life for the better

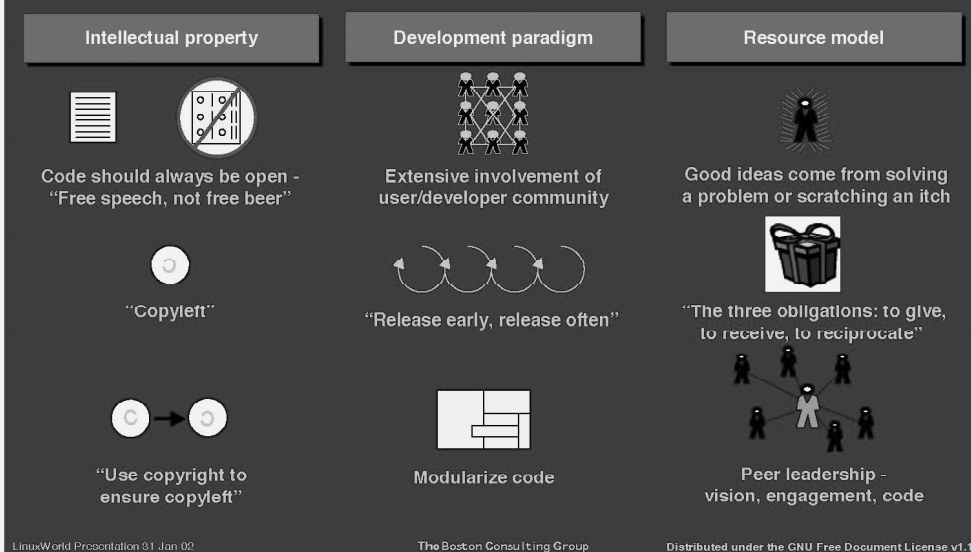
❑ Sense of responsibility as a counterpart to freedom

❑ The word « hacker » acquired a corrupted connotation in the 90's

PROJECT FORKING (3/3)

- ❑ **When a project gives birth to two variants, which gradually diverge**
- ❑ **Various possible reasons**
 - ❖ Technical (eg: port on # platforms)
 - ❖ Human (eg: lead conflict)
 - ❖ Business (eg: starting a new commercial offer)
- ❑ **Forks are the best... and the worst thing**
- ❑ **« Survival of the fittest »**

OPEN SOURCE PRINCIPLES



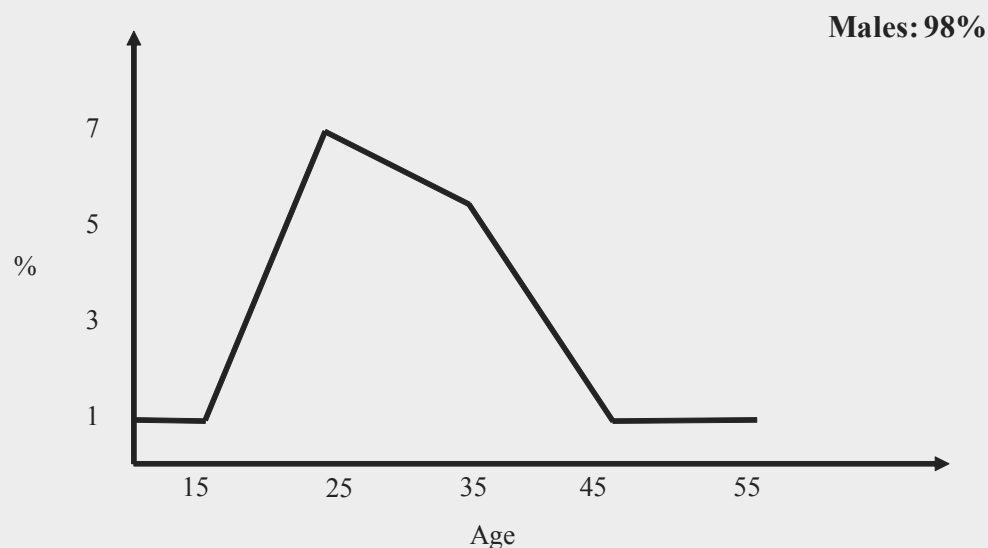
MOTIVATIONS AND PROFILE OF FLOSS DEVELOPERS

❑ Main motivations (BCG Hacker Survey 2002)

- ❖ Stimulated Learners - write code for both skill improvement and fun (29%)
- ❖ Hobbyists - are all stimulated by the opportunity to develop skills for non-work activities (27%)
- ❖ Professionals - involved in the OSS community to enhance their work skills, professionals (25%).
- ❖ Community Believers - believe strongly that source code should be open (19%)

❑ 65% do not contribute from work - or without their hierarchy being aware of it

THE TYPICAL FLOSSDEVELOPER, AS OF 2002



GENDER BIAS IN FLOSS COMMUNITIES

- ❑ **% females: <2% in FLOSS / 28% in proprietary software**
- ❑ **Women are actively (if unconsciously) excluded of FLOSS communities rather than passively disinterested**
 - ❖ The 'hacker' ethic, situates itself outside the 'mainstream' sociality, and equates women with that mainstream
 - ❖ FLOSS rewards the producing code rather than the producing software
 - ❖ Women tend to engage later in their lives with computers. To join they have a larger amount of catching up work to do
 - ❖ Flame-wars are accepted, with an off-putting effect, particularly pronounced in the case of women
 - ❖ Men are usually freer to participate than women in long hours of intensive computing

RECAP

- ❑ **The project is the basic unit in FLOSS communities**
- ❑ **Projects evolve, fork**
- ❑ **Often organized around a code base, members of the community have various levels of contribution**
- ❑ **Typical FLOSS developers follow personal motivations (not primarily making profit)**
- ❑ **Mostly men, 25-35, hacker culture**

BUSINESS MODELS

CREATED BY: FRANÇOIS LETELLIER
FOR "FOSS BRIDGE EU-VIETNAM - JOINT BUSINESS THROUGH FREE
AND OPEN SOURCE SOFTWARE", INWENT
VERSION / DATE : FOSS-BRIDGE BLOCK14 /2008/04/17

WHAT WE'LL LEARN NOW...

- ❑ **What a « business model » is**
- ❑ **How companies build a commercial offers**

WHAT IS A « BUSINESS MODEL »

Planning level: the Strategy

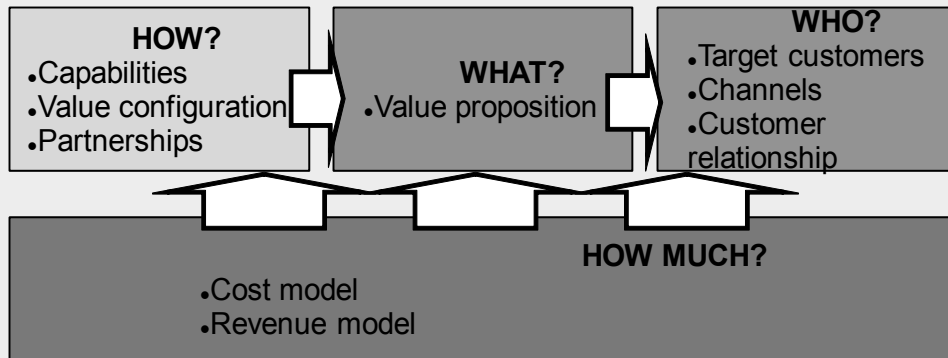
**Architectural level:
the Business Model**

Operations: the Business Processes

WHAT IS A « BUSINESS MODEL »

- ❑ **Not only a revenue model**
- ❑ **Four aspects:**
 - ❖ Infrastructure management
 - ❖ Product innovation
 - ❖ Customer relationships
 - ❖ Financials

WHAT IS A « BUSINESS MODEL »



-> **business logic of how a company generates profit in a sustainable way**

WHAT IS A « BUSINESS MODEL »?

Product Innovation	• Target Customer Segment • Value Proposition • Capabilities
Customer Relationship	• Information Strategy • Feel & Serve • Trust & Loyalty
Infrastructure Management	• Resources • Activity Configuration • Partner Network
Financials	• Revenue Model • Cost Structure • Profit/Loss

CLASSICAL BM IN SOFTWARE SOFTWARE COMPANY

- ❑ **In a nutshell - and typically:**
 - ❖ An software company (could be an « ISV ») develops software products...
 - ❖ on its own internal R&D;
 - ❖ These products are packaged as COTS (commercial off-the-shelf) software...
 - ❖ protected by copyright (& optionally patents, trademarks);
 - ❖ COTS are licensed to customers for a fee;
 - ❖ Additionally, the company may provide basic services (support, maintenance, etc).

CLASSICAL BM IN SOFTWARE SOFTWARE COMPANY

- ❑ **High profitability once a critical mass is reached**
- ❑ **But significant risk of negative return on internal R&D investment**
- ❑ **Importance of network effects to grow the market**
- ❑ **Needs enough internal R&D to remain innovative**
- ❑ **Lock-in strategy not uncommon**

CLASSICAL BM IN SOFTWARE PROFESSIONAL SERVICES

- ☐ **In a nutshell - and typically:**
 - ❖ A company providing professional services...
 - ❖ Delivers manpower intensive services such as bespoke software development / technical consulting / outsourcing...
 - ❖ To non-IT customers or as subcontractor of IT companies.
 - ❖ Each contract is based on the detailed requirements of the customer...
 - ❖ And priced based on daily rates.

CLASSICAL BM IN SOFTWARE PROFESSIONAL SERVICES

- ☐ **Limited initial investment**
- ☐ **Reduced margin but limited risk**
- ☐ **Internal R&D very limited, most technical manpower dedicated to serving customers**
- ☐ **Usually technology agnostic and aligned on customers preferred technologies**
- ☐ **Skills / expertise (and agility) necessary to remain competitive and keep customer loyalty**
- ☐ **Prone to offshore competition**

CLASSICAL BM IN SOFTWARE VALUE ADDED RESELLER

- ❑ **In a nutshell - and typically:**
 - ❖ A Value Added Reseller (VAR) offers turn-key, customized solutions...
 - ❖ To non IT specialist, end-user companies.
 - ❖ Solutions are built on COTS, customized according to requirements...
 - ❖ And offered with additional professional services, such as consulting, training, maintenance.
 - ❖ The VAR gets discounts on COTS licenses thanks to agreements with software companies...
 - ❖ And makes its margin both on license resale and services.

CLASSICAL BM IN SOFTWARE VALUE ADDED RESELLER

- ❑ **VARs do co-marketing with software companies, and act as distribution channels for them**
- ❑ **Strategic decisions must be taken as to which software company to partner with - when several competitors offer similar products**
- ❑ **In terms of risks and profitability, sort of tradeoff between software companies and service companies**

RECAP

- ❑ **Business models are not only revenue models**
- ❑ **Four aspects:**
 - ❖ Infrastructure management, product innovation, customer relationships, financials
- ❑ **Software company, professional services, VARs are typical business models in IT**

November 2008 – The FOSS-Bridge EU-Vietnam project “Joint Business through Free and Open Source Software” releases a series of advanced FOSS Training material on innovative business models and EU regulations.

“High Level FOSS Bridge Training Material” is a compilation of advanced courses covering key issues such as “Open Source Regulations, Licensing and Business Models, Open Source Development & European Best Practice as well as design of Training Activities”.

The Training material is now online at the following URL: <http://www.foss-bridge.org/trainingmaterial.html>. It is made available as open format (odt) and can be downloaded both in English and Vietnamese language. This is targeting trainers from Vietnamese Intermediaries as well as other interested parties. The main purpose of this training course is to enable Intermediaries of vietnamese IT sector to spread key knowledge on how to create business based of FOSS.

Thanks to the authors

We would like to thank the team of authors and trainers which has been active in building up and adapting the modules of the course. This training content has been built by Christophe Ney and François Letellier, two European experts in the field of open source technical regulations who were in charge of the Training program in cooperation with Vietnamese partners and participants.

The editing team

FOSS Bridge EU-Vietnam is an ASIA-Invest Programme implemented by InWEnt – Capacity Building International of Germany, Institut National de Recherche en Informatique et en Automatique (INRIA) of France, and the Institute of Information Technology (IOIT) of Vietnam with financial support by the European Commission and the German Federal Ministry for Economic Cooperation and Development (BMZ).

Contact

web: www.foss-bridge.org

e-mail: contact@foss-bridge.org

Tháng 11 năm 2008, Dự án FOSS-Bridge EU-Vietnam “Hợp tác kinh doanh bằng Phần mềm Tự do & Nguồn mở (FOSS)” đưa ra một loạt các tài liệu đào tạo nâng cao về các mô hình kinh doanh và quy định của EU về FOSS.

Tập “Tài liệu Đào tạo nâng cao FOSS Bridge” là tổng hợp của các khóa học về các chủ đề quan trọng như “Các quy định của Phần mềm Tự do & Nguồn mở, mô hình giấy phép và mô hình kinh doanh; Sự phát triển và các bài học từ Châu Âu; Thiết kế các hoạt động tập huấn”.

Các tài liệu này có thể truy cập trực tuyến tại địa chỉ: <http://www.foss-bridge.org/trainingmaterial.html>. Tài liệu được đặt định dạng mở (odt) và có thể tải về cả phiên bản tiếng Anh và tiếng Việt, dành cho các học viên từ các tổ chức xúc tiến, các nhân viên công ty Công nghệ thông tin, và những ai quan tâm đến FOSS. Mục đích chính của các khóa đào tạo là cho phép các tổ chức xúc tiến trong lĩnh vực Công nghệ thông tin của Việt Nam có thể phổ biến những kiến thức, kinh nghiệm về việc làm thế nào có thể kinh doanh được với Phần mềm Tự do & Nguồn mở.

Lời cảm ơn các tác giả

Chúng tôi chân thành cảm ơn nhóm tác giả đã dành nhiều công sức biên soạn tập tài liệu huấn luyện này. Tài liệu này được biên soạn bởi Christophe Ney và François Letellier, hai chuyên gia Phần mềm Tự do & Nguồn mở đến từ Châu Âu, những người chịu trách nhiệm về chương trình đào tạo của FOSS-Bridge cho các đối tác, học viên Việt Nam.

Ban biên tập

FOSS Bridge EU-Vietnam là một dự án ASIA-Invest được thực hiện bởi InWEnt - Tổ chức Nâng cao năng lực của CHLB Đức, INRIA - Viện nghiên cứu quốc gia về CNTT và Tự động hóa của CH Pháp, và IOIT - Viện Công nghệ thông tin của Việt Nam với sự hỗ trợ tài chính của Cộng đồng Châu Âu EC và Bộ Liên bang về Hợp tác kinh tế và Phát triển CHLB Đức (BMZ).

Liên hệ

web: www.foss-bridge.org

e-mail: contact@foss-bridge.org

Project funded by



Project implemented by

Networking partners



Part of the project



