Introduction to OSS Business Models

Peter C Rigby

Business Models

- Standard
    - Make it Proprietary
    - Functional Encapsulation
    - Support Services and Training
- Dual-license Model
    - Proprietary Components
    - Freemum
    - Delayed Open Sourcing

Business Models

- Tricky
  - Software as a Service (SaaS)
  - Advertising-Supported Software
- Other factors
  - Branding
  - Donations
- Infrastructure
  - Paid Developers
  - Proprietary Modules

Make it Proprietary

- Use a permissive (MIT, BSD) license
- Modify and release under a proprietary license
- Apple incorporated combined code from Mach and BSD to create OS X
- Unix history

Functional Encapsulation

- OSS software install separately from proprietary software
  - eg Download Linux distribution and then download proprietary software
- Shipped separately from OSS software even though it uses it

Functional Encapsulation

- Similar to commercial products
- e.g., software that runs on Windows isn't shipped with windows
    - e.g., Adobe Reader, Firefox

Linux revenue?

$2.4 Billion in first quarter of 2012
For whom?

http://www.internetnews.com/blog/skerner/linux-server-revenues-growing-faster-than-windows-hit-2.4-billion-in-1q12.html

Support Services

- For example, charge for
    - annual support fee
    - per-student/employee training fee
    - per-project consulting fee

Support Services

- How hard is it to configure your software?
- Provide the source but don’t provide the binaries
- Provide a compilation and packaging software service

Support Services

- Successful examples:
- Red Hat
  - 1.13 billion in 2012
  - Subscription based customer support
  - Quality assurances

Dual-license Model

- Released under an OSS license
- As well as a commercial license
- Wikipedia, dual-licensing

Dual-license Model

- Need reciprocal style license (eg GPL)
- Need to own all the copyright
- Release software under a proprietary license
  - Companies wanting to sell modifications to product, must buy the proprietary license

Dual-license Model

- Company must own entire copyright for the system
- Must require contributors to assign copyright to company
    - What happens when contributors start another fork?

Dual-license Model

- Forks threaten a company's control of the project
- Fork doesn't have to assign copyright to company, so
- Company can't use code in fork
- But project can use anything company code the company releases

Dual-license Model

- Forking is a huge risk
- Company can lose control of the project
  - Don’t own copyright on fork
  - MariaDB a fork of MySQL
- Must keep the community happy

Proprietary Components

- License part of the system with OSS
- License another part as proprietary
  - Proprietary hardware
  - Proprietary artwork for videogames
  - Proprietary data

Hardware

- Charge for the hardware
  - Includes only the running binaries
- Release the source code
  - Usually don’t release the binaries

Hardware

- Risk
  - If your proprietary hardware is similar to a competitor,
  - they may be able to use your source code, and
  - release a similar product

Videogames

- Release the engine as OSS (infrastructure)
- Keep the artwork, audio, graphics, etc as proprietary
- e.g, Kot-in-Action Creative Artel video game Steel Storm

Freemium

- Basic version is free
- Pay for premium features
  - Capacity limited
  - Seat limited
  - Customer class limited (educational user)

Freemium

- Why is this a more difficult model in OSS?

Freemium

- Example, seat limited
  - Find the place in the code where the number of instances is limited and change that value
- Must have inherent limitations or entire features missing

Delayed Open Sourcing

- Provide the latest version to paying customers under a proprietary license
- Release the latest patches under an OSS license after some time has passed

Delayed Open Sourcing

- Release as OSS on a regular and fixed timeframe
- Release after end-of-life
  - Avoid becoming abandonware

Release after end-of-life

- User community continues to maintain and evolve system
- Very common with videogames that have had a good return on investment
  - Want to focus energy elsewhere
  - Freely released commercial games

Release after end-of-life

- Other examples,
  - Netscape became Mozilla Firefox
  - Sun’s StarOffice became

Question

- If I modify Linux and use it privately, do I have to release the source code back to the community?

Question

- If I modify Linux and use it on internal servers to offer services to the public, do I have to release the source code?

Software as a Service (SaaS)

- Charge for
  - Services with a subscription
  - Have service and desktop software
    - software plus services

Software as a Service

- Stallman thinks this is "inherently bad"
- A loophole in the GPL
- AGPL fixes the hole
  - Not included in GPL V3
  - Most OSS licenses allow this service model

Android

- Why does Google support the development of Android?

Android

- Why does Google support the development of Android?
- Ad revenue, get people to use your products

Advertising-Supported Software

- Provide ads with your software
- Usually a SaaS running OSS in background
- Google, Mozilla, Ubuntu

Coca-cola brand?

Coca-cola brand?

worth $72 billion

Why buy OSS?

- Brand, people will know you
- Good will - Investing money into a community project
- Critical mass of users
- It’s the competition and will exist anyways
- Technical know how and infrastructure

Source code is freely available and distributable

- VMware bought SpringSource for $420 million
    - 20 times annual sales
- But, SpringSource code is freely available!
- Why!?

http://www.nytimes.com/2009/11/30/technology/business-computing/30open.html?pagewanted=2&_r=0

Firefox Brand

- Sells t-shirts etc
- Legal action against fake merchandise

Donations

- Micropayment systems (eg paypall)
  - Sourceforge has a donation link
- Firefox’s fundraising campaign with ad in New York Times

Infrastructure Project

- Not part of your core services
- A group of companies all need the same software
- Create a foundation and pay developers who work on the infrastructure

Paid Developers

- Many core developers on major systems are paid
    - Linux, KDE, Apache
- IBM donates Lotus Symphony to Open Office, as well as developer time

Proprietary Modules

- License must allow proprietary linking (LGPL)
- For example, EPL
  - According to article 1(b) of the EPL, additions to the original work may be licensed independently, including under a <u>commercial license</u>, provided such additions are "separate modules of software" and do not constitute a <u>derivative work</u>.

Proprietary Modules

- <u>Eclipse Ecosystem</u> has many proprietary plugins

Proprietary Modules

- Eclipse Ecosystem has many proprietary plugins
- Rational Software Architect for WebSphere Software

Proprietary Modules

- Eclipse Ecosystem has many proprietary plugins
- Rational Software Architect for WebSphere Software
  - $3,340 for a user license

Paid developers

- Project is important to a company
- Pay developers to ensure that software remains useful to company
  - Also have influence over decision making

Reference

- See links in document
- Wikipedia provides an good overview, their page is under revision as this is a evolving area