



# FLOSS: concepts and development

OSM2011  
Kaido Kikkas

# Traditional model

- Typical business model - everything seems clear:
  - Investors put the finances in, create a company
  - The company purchases the tools (hard- and software)
  - Employees are hired
  - Product(s) is/are developed
  - Advertising and marketing is arranged
  - The product(s) is/are sold by unit
  - The profit is lucrative due to near-zero duplication costs
- “If I give my work away, how will I survive?”
- “Freeware” is used as everywhere else – as free samples, a bonus, or to invade a new market

# Freeware?

- To address a popular misconception: we are not going to talk about freeware. Freeware is proprietary software (just like MS Windows or Adobe Photoshop) that just happens to have zero price at the moment. Shareware is also a different kind of animal
- Price is actually the visible tip of the iceberg
- Free and open-source software is actually about development and distribution mechanisms
- Community is the key



# Widespread misconceptions about FLOSS

- Must be free (as in no cost – gratis)
- No commercial use
- Cannot be sold
- Amateurish
- Insecure
- No warranty or support
- ALL THOSE ARE INCORRECT IN VARYING DEGREES

# In the beginning, there were hackers

- In the ancient times, every computer had its own software => it came with the computer just like a spare tyre with a car
- In the 50s and 60s, IT was regulated by the government in the US => no business. Rather, larger centres attracted 'angry young men' becoming the first-generation computer aces (see also “Hackers: the Heroes of the Computer Revolution” by Steven Levy)
- “Information wants to be free” - all creation was shared. At first, even passwords were shunned

# Factors

- “computer science” ~ “rocket science”
- Very limited technical resources (today's cell phones are more powerful than those machines were)
- Little to no compatibility, each computer had its own software
- No stimulus (or sometimes rights) for business
- Some reflections of the 'wild 60s'

# The original Hacker Ethic (Levy)

- Access to computers should be unlimited and total. Always yield to the Hands-On Imperative!
- All information should be free
- Mistrust Authority – Promote Decentralization
- Hackers should be judged by their hacking only
- You can create art and beauty on a computer
- Computers can change your life for the better

# The Business Age

- The 70s - software commercialisation (gradual)
- A main reason – increase of portability which ensured 'critical mass' for commoditisation
- Unix was closed up at the end of 70s (79/84), MacOS (83) and MS-DOS (81) were closed from the beginning. Proprietary operating systems were followed by similar application software
- About 1980-1995 – software is “a product like everything else” (books, records)

# Return of the hackers

- 1984 – Richard Stallman founds GNU
- 1991 – Linus Torvalds, Linux kernel
- 1992-3 – BSD386 and free BSD Unixes
- Ca 1995 – LAMP server technology
- 1998 – “The Halloween Documents”, Freeware Summit, OSI
- 2001 – OpenOffice.org
- 2002 – Creative Commons
- Open Access Initiative, Flickr, Wikipedia...

# Enter naysayers

- 1991 – "who needs THAT?"
- 1993 – "who cares to learn all that?"
- 1994 – "who does business with that stuff?"
- 1995 – "A server running Linux????"
- 1997 – "Linux – with graphical environment?"
- 1998 – "Oracle on Linux...?"
- 1999 - "Linux and Heroes of Might and Magic?"
- 2001 – "OpenOffice.org will never match MS Office!"
- 2002 – "South America has lost its mind!"
- 2003 – "Linux has no multimedia!"
- ...

# Two schools: Free Software

- Richard Stallman and FSF
- Not only technical but also philosophical and ethical category: “Helping your neighbour”
- Proprietary solutions are unethical
- Somewhat hard line but clear positions
- See <http://www.fsf.org>

# Two schools: Open Source

- Linus Torvalds, Bruce Perens, Eric S. Raymond
- “Shut up and show them the code!”
- Proprietary software is not unethical, but just a sub-optimal solution
- Pragmatical approach, sometimes blamed for compromise (e.g. the Novell-Microsoft deal)
- See <http://www.opensource.org>

# Uneasy allies

- Lots of common ground in practical projects
- Largely overlapping licensing, especially GPL:
  - The GNU General Public License (GPL) 1989/91/2007
  - Central point – authorship PLUS user's rights:
    - To use for whatever purpose (freedom 0)
    - To study and modify (freedom 1; implies access to source code)
    - To distribute copies (freedom 2)
    - To distribute the modified form (freedom 3; implies access to source code)
  - Switch from a product model to a mixed model of product/service – with emphasis on the latter
- The starting point for most phenomena of Open Culture – Open/Free Content, CC, Free Art etc

# A point

- “Free software doesn't mean no revenue, it means no barriers to revenue.” – Jonathan Schwartz, former CEO of Sun in November 2005
- Main idea:
  - Maximise distribution, remove barriers
  - Provide ways to generate additional value for both the original author and any interested party
- In principle, the single most important resource is needed: the human brain

# Current status

- Main business for Red Hat (also Canonical and a number of smaller firms)
- Visible business component in Oracle, Novell, HP, IBM...
- Microsoft? (Codeplex, MS-PL...)
- See also
  - <http://gigaom.com/2010/01/22/how-red-hat-routed-the-recession/>
  - <http://linux.slashdot.org/story/10/01/21/230201/75-of-Linux-Code-Now-Written-By-Paid-Developers>

# Homework

- Study the FSF and OSI websites and write a blog review to compare the views of the two schools of thought
- Play Wesnoth :) Actually, you need to be familiar with the game to be able to develop it
- Discuss and try to determine the role distribution at the project in your team (storyline, graphics, music, storyboarding/ special events etc). Do some related research on the game (from website)



---

</First\_Lecture>

---