Open Source Software

Dr Vinit Kumar
Assistant Professor
Dr. Ranganathan Institute of Library and Information Science
Bundelkhand University, Jhansi

Premise

- Open source software(OSS) is very often developed in a public, collaborative manner.
- OSS is the most prominent example of open source development

OSS : Definitions

OSS can be defined as computer software for which the human-readable source code is made available under a copyright license that meets the **Open Source Definition.**

The Open Source Initiative wrote a document called The **Open Source Definition** and uses it to determine whether it considers a software license

Open Source Definition(OSI)

1. Free distribution
2. Source code
3. Derived work
4. Integrity of the author's source code
5. No discrimination among person or groups
6. No discrimination among fields of endeavour
7. Distribution of license
8. License must not be specific to a product
9. License must not restrict other software
10. License must be Technology-neutral

Milestones

1983

Richard Stallman formed GNU project.

1985

Creation of **Free Software Foundation**.

1991

Development of Linux kernel by **Linus Torvalds**.

1998

Open Source Initiative (OSI) formed by **Eric Raymond**.

Similar Movements

The philosophy of OS inspired several other movements:

- Open Content Movement
  - CC
  - Copyleft
- Open Source Hardware
- Open Standards
- Open Access

Prime Examples of OSS

- Linux(GNU)
- Android
- Apache HTTP server
- WordPress, Joomla, Drupal
- Mediawiki
- Mozilla Firefox
- Google Chrome(Chromium)
- MySQL(Community edition)/PostgreSQL

Reasons behind boost in OSS

- Better communication system
- Decreasing cost of computing
- Community collaborative effort
- Availability of source code
- Free access and good level of functionality

Open Source Licenses

An open source license is a copyright license for computer software that makes the source code available under terms that allow for modification and redistribution without having paying the original author.

- But there are some restrictions
  - a requirement to preserve the name of the authors
  - and the copyright statement within the code

Licensing Options in OSS

- GNU General Public License (GPL) – most popular
- BSD license,
- GNU Lesser General Public License,
- MIT License,
- Mozilla Public License and
- Apache License

Typical Features of OSS

- Users are treated as Co-developers
- Users are encouraged to submit additions to the software, code fixes for the software, bug reports, documentation, language translations, local support, etc.
- This increases the rate of evolution of software.
- Several testers on several machines, in several settings

Typical Features of OSS

- Early release
  - The first version of the software is released as early as possible to increase the chances of finding co-developers.
- Frequent Integration
  - The code changes(patches) are integrated as often as possible to avoid the overhead of fixing large number of bugs.
  - Once a significant number of features are added the new version of software is released.

Typical Features of OSS

- Often two releases
  - Release client
    - Buggier version with advanced features
    - Released as Nightly build or Development version
    - For testing purpose
    - is for those users who want the immediate use of the latest features, and are willing to accept the risks(bugs)
  - Stable release
    - Less buggier with less features
    - Stable and supported
    - For those users who want to avoid risks.

Typical Features of OSS

- **Modularisation**: usually modular structure of software is chosen for allowing parallel developments on several components
- **Version management** is done by dedicated platforms like, github, Tracks
  - As several parallel development is done.
  - Project lead developers are chosen who decide to commit or reject the packages
- **Software code hosting** on
  - Dedicated servers of individual projects
  - As well as on repositories like
    - Github
    - Sourceforge.net
    - code.google.com
    - Linux package repositories

Typical Features of OSS

- **Support** :
  - Through Web-forums
    - Project's hosted or
    - External (Stackexchange, etc)
  - Mailing lists
    - Users mailing list
    - Developers mailing lists
    - Technical mailing lists
  - Documentation
    - Wiki
    - Video tutorials
    - Training modules
    - Workshops/training sessions
    - Special Webcasting sessions

Disadvantages

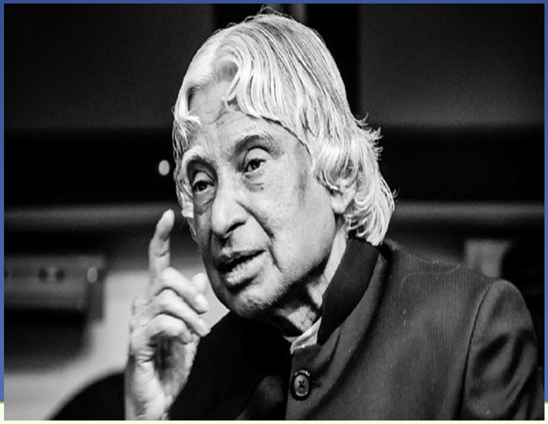
- No hurry to release updates. So the developments goes as per the urgency of demand of users.
- Rely more on community: sometimes no developer is available to provide solution to a particular problem
- No flashy interfaces (less user friendly).
- Although the software is free, but the service costs.
- Training also costs.
- Because the source is open the malicious users can potentially view it and exploit any vulnerabilities.
- Once the software reaches a target user base, **often** it converts to a paid version.

Choosing the "Right" Open Source Software

- Size and activity level of the open-source community
- Open-source license (not just free)
- Active deployments
- Available technical know-how

Explore

- Think long-term.
- Avoid vendor lock-in.
- Design your software for worst-case success.
- Consider the hidden costs.
- Explore components that are not industry-specific.
- Flexibility to leverage the newest capabilities.
- Staff time.



"Open source codes can easily introduce the users to build security algorithms in the system without the dependence of proprietary platforms. We should take maximum care to ensure that our solution is unique to protect our own defence security solutions implemented on open platforms,”

-- Dr APJ Abdul Kalam

Dr. Kalam had consistently advocated the use of open source software over proprietary software whose building block is rarely disclosed by corporates who own and sell them.

Thanks