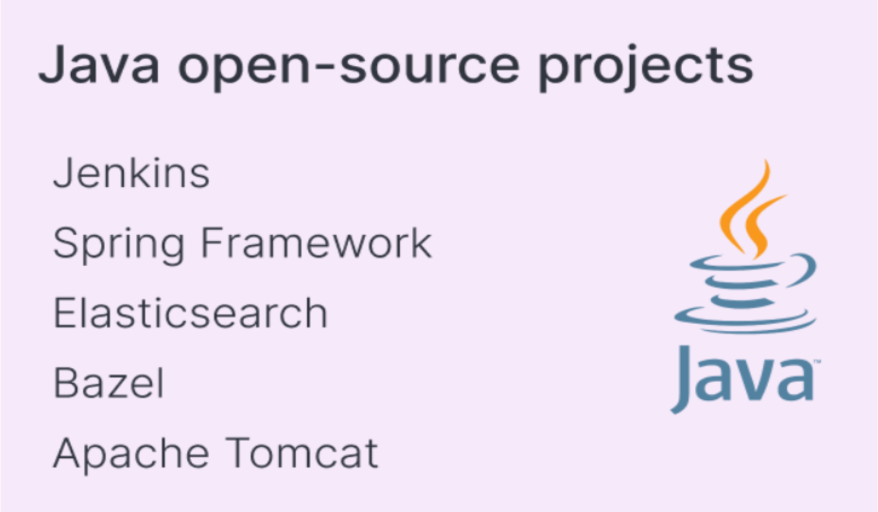
Open Source Software Engineering

SE ZG587

BITS Pilani
Pilani Campus

Kumar Manish
9th Sept, 2023

Open source projects



Java open-source projects

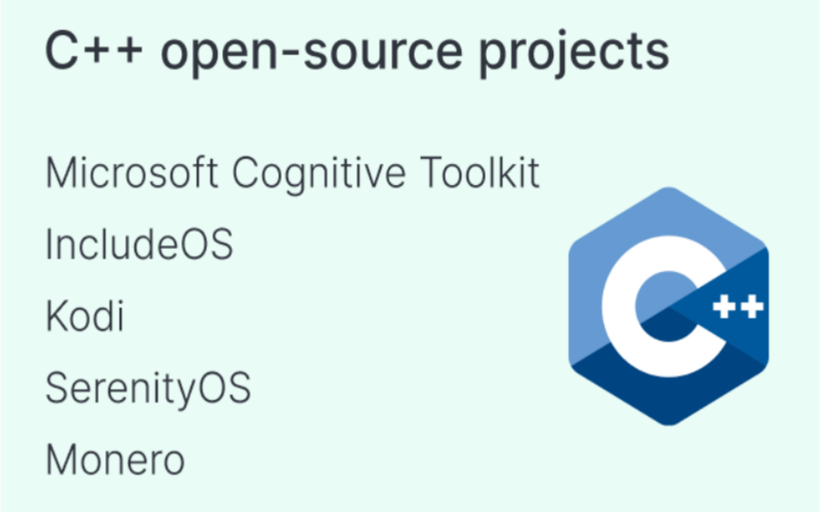
Jenkins

Spring Framework

Elasticsearch

Bazel

Apache Tomcat



C++ open-source projects

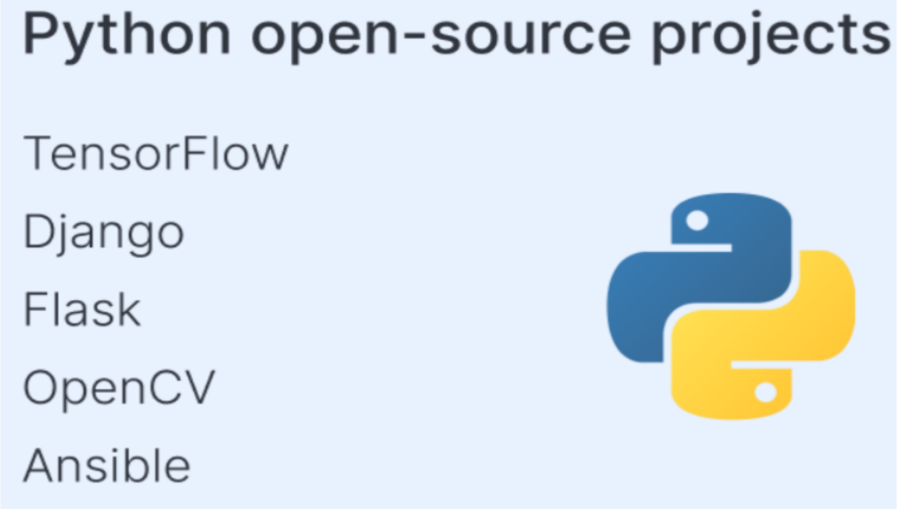
Microsoft Cognitive Toolkit

IncludeOS

Kodi

SerenityOS

Monero



Python open-source projects

TensorFlow

Django

Flask

OpenCV

Ansible

https://opensource.google/
https://opensource.fb.com/projects/
https://www.gimp.org/ : Alternate of Photoshop
https://www.openoffice.org/ alternate of Microsoft office
https://www.alfresco.com/ alternate of Content management

Agenda : Recap

Lifecycle and methodologies in Open Source Software

- Open Collaboration Model
- Community Driven Development Model

Open Collaboration Model

- **Open collaboration** is “a system of innovation or production or development that relies on **goal oriented**,
- yet loosely coordinated,
- participants who interact to create a product (or service) of economic value,
- and make it available to contributors and non contributors alike.”

Open Collaboration Model : principles

- **Equalitarianism :**
  - To specify that everyone can contribute.
  - social equality, prioritizing it for all people.
- **Meritocracy :**
  - To judge contributions, in a transparent manner, based on the merit or the quality of the contribution.
- **Self –organization :**
  - communities organise themselves without any external control or influence from a person or process.

Community Driven Development Model

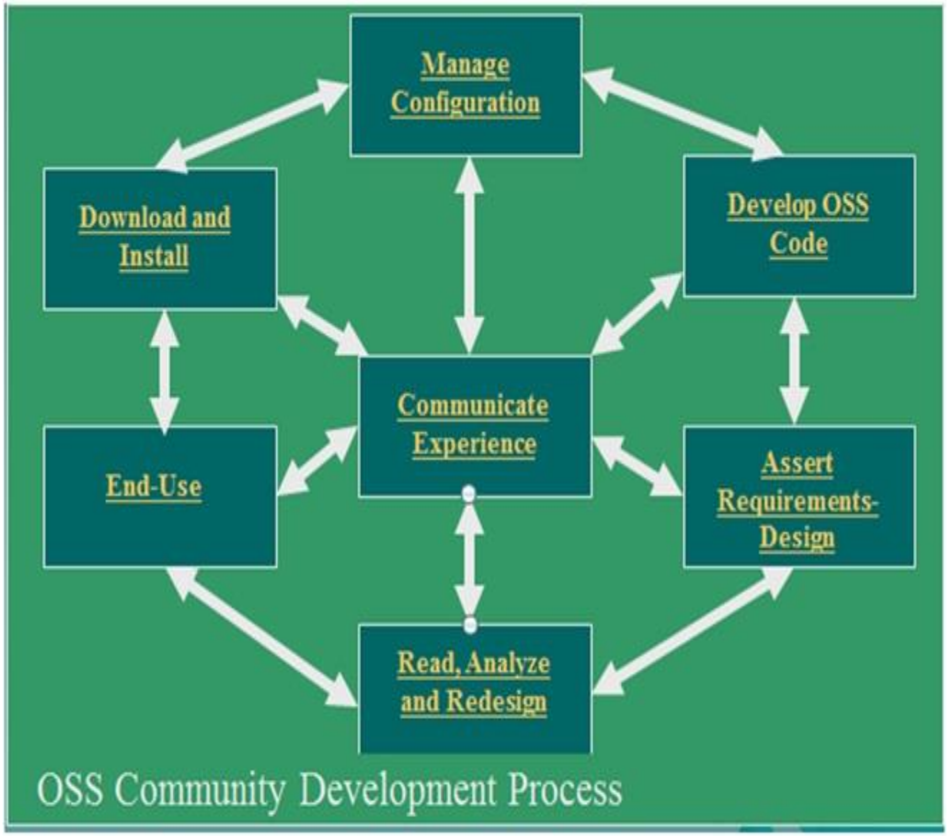
- A group of contributors or developers align their activities together and engage toward the development of open source software
- These group of people are often termed as “Community“
- A community can also be defined as a group of people
  - Which are diverse in nature
  - And engage in sharing ideas , work and experiences
  - Through a common platform for a common cause
- Community comprises of various jobs , roles and multiple sub teams : **Developer’s group, Builders group , Testers group Release management group**

Community Driven Development Model

- **Self driven** OSS development communities , often provide flexibility to learn members to shuffle between job roles.
- These communities are **open to customers or end users , inviting them to join the community** and contribute to the project
- **Agile mindset, self motivated, self organised**

Community Driven Development Process

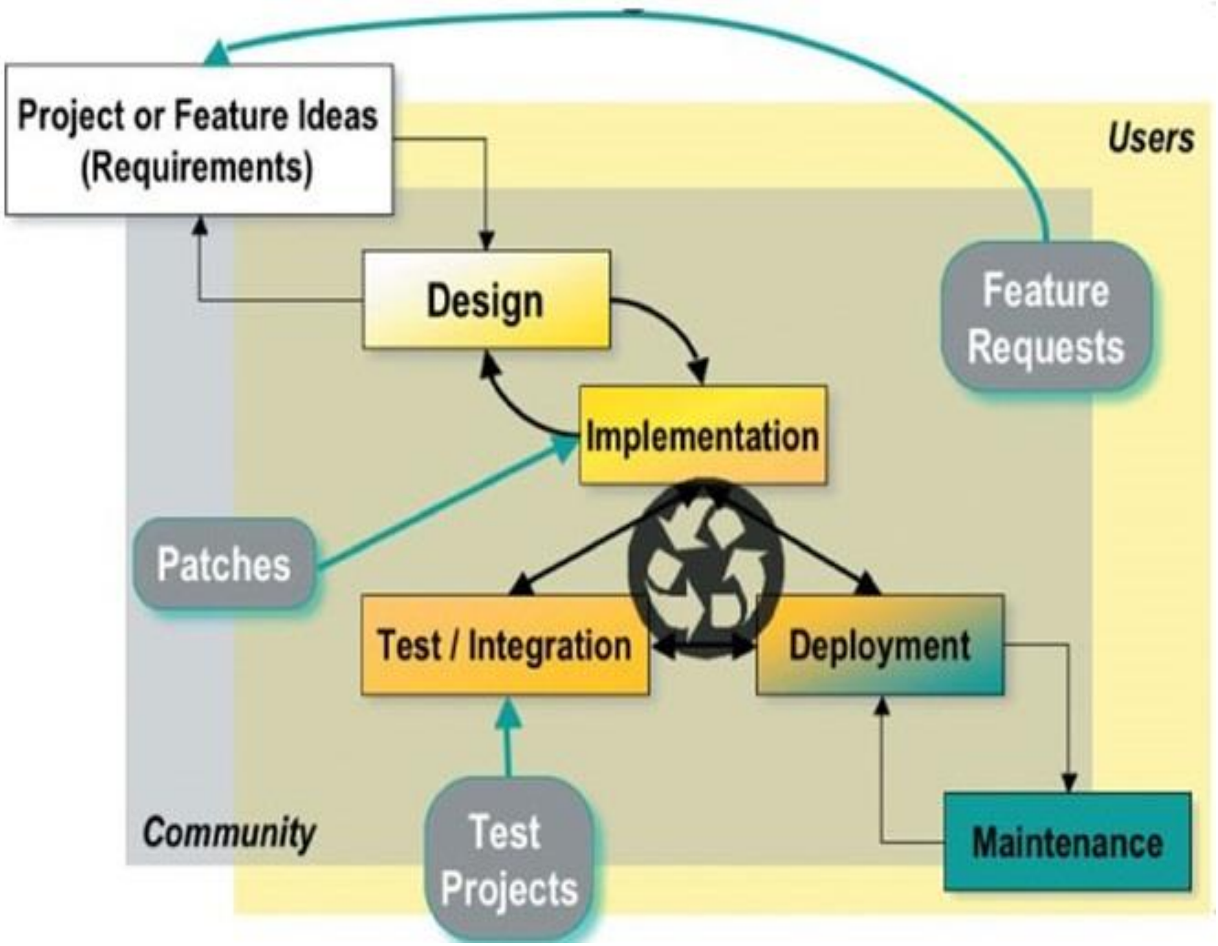
- Requirements & Design
- Develop Code
- Manage Configuration
- Download and Install
- End-Use
- Communicate Experience
- Read, Analyze and Redesign


Manage Configuration
Download and Install
Develop OSS Code
Communicate Experience
End-Use
Assert Requirements-Design
Read, Analyze and Redesign
OSS Community Development Process

Open Source Software Development Model

- **OSSD Model is different from the traditional waterfall or cascading model of software development.**
- **Open Source Development Model** Involves an interconnected OSS Community Development Process in which each stage or phase plays a vital role in building the ethics of the community, keeping contributions of each developer in mind, working with the latest technologies, keeping a track on the version control system and fixing the bugs in the software.
- The moment the software runs (partially , mostly), it is released as development release (even though it may contain known and unknown bugs) .
- In alignment of Philosophy : **“Release early, release often “**

Open Source Software Development Model



Release early, release often

- Idea for a new project or new feature or functionality
- Design for proposed solution
- Implementation by running (partially or fully)
- Development release (even bug)

Quick check

- Can open source software be used for commercial purposes ?

- Can I restrict how people use an open source licenses programs ?

- Can I call my programs : open source “ even if I don't use an approved licenses ?

Recap :

- **Open source :** Collaboration open and Source freely available
    - Freely : Share, Adapt, Modify and Collaborate
- **OSS :** Source code Released under a **License** in which the **copyright** holder grants users the rights to **use**, **study**, **change**, and **distribute** the software and its source code to anyone and for any purpose.
- **Key Principle of OSS :** Openness, Transparency, Collaboration, Expectations of Community
    - Release Early and Often : Rapid prototypes and iterative approach
- **Free software**" means software that respects users' freedom and community. It refers to free use of software and not price
- **OSI :** Definitions, Standard Requirements, Licenses
- **FSF :** To promote computer user freedom and defend the rights of all software users
    - Richard Stallman in 1983, by launching **GNU Project**

Recap :

Free software Four essential freedoms

- **A program is “free software” if its users have the following** :
  - Freedom 0 : The freedom to run the program as you wish, for an purpose
  - Freedom 1 : The freedom to study how the program works , and change it as per your requirements
  - Freedom 2 : The freedom to redistribute copies so you can help others
  - Freedom 3 : The freedom to distribute copies of your modified versions to others
    - **Pre-requisite “ Access of the source code”**
- **Legal considerations :**
  - The owner of the free software does not have power to withdraw or invalidate the license , or add additional restrictions to its terms and conditions,
- **License terms should be permanent and irrevocable**
- **License must not restrict other software**
- **License must be technology neutral**

Recap :

	Free software	Open source Software	Freeware	Public domain software
Definition	"FREE" is a matter of liberty , not price	"OPEN" does not just mean access to the source code; more about collaboration	Free refers to price , while freedom of the use is restricted by creator	PUBLIC domain belongs to the public as a whole
Philosophy	Social movement	Development methodology	Marketing goals	Copyright disclamation
Rules	Four freedoms	Open source Initiatives		Creative common Org
Free of charge	Not necessary	Not Necessary	YES	YES
Copyright law	YES	YES	YES	NO
Examples	Linux, Ubuntu , MySQL, Apache	Linux, Ubuntu , MySQL, Apache	Skype, Adobe acrobat	SQLite

Recap :

- Understanding Intellectual property Rights (**IPR**) to the software industry
  - **Patents** – used to protect functional features , like hardware configurations etc
  - **Copyrights** – used to protect works of authorship, like source code , diagram etc
  - **Trade secrets** – used to protect internal business secrets, like business and pricing models
  - **Trademarks** - used to protect brand recognition, through logo etc
- All the people involved in a development project, may claim for the ownership of the IPR for the various artefacts developed as a part of the project.

- **Understanding Software Licenses**
  - **Licensing Models in OSS:**
    - Copyright, Copyleft, Permissive, Creative Commons

Recap :

Copyleft or Protective Licensing Model :

- To make a work (or program) free to use, modify , adapt or extend.
- Aligned with the **four essential freedoms**
- No re-licensing allowed, No commercial usage allowed
- All modified and extended version of the work) or program) should be free as well - hence called **protective licenses**
- All derivative works should be attributed to the creator, open sourced and copyleft

Variants of Copyleft : **Strong and Weak copyleft :**

- The strength of the copyleft license is decided based on the extent its provision are imposed on the derived works
- Most commonly, weak copyleft licenses are used to create **Software libraries**
- Help software to link to the library and redistributed without requirement for the linking software to also be copyleft –licensed

Examples :

- Strong Copyleft :
  - GNU general public License
- Weak copyleft
  - GNU lesser General Public License –
  - Mozilla public license

Recap

Permissive Licenses

- Free software licenses with only minimal restrictions on how the software can be used , modified and redistributed (also called Berkeley software distribution : BSD - like or style licenses)
- Rules for usage - what ever user wants, but with few restrictions – derived works must be attributed to the creator
- Source code need not be open or made available in the public domain
- Re-licensing allowed - derivative works can be release under any other licenses or used as proprietary products
- Allows commercial usage
- Examples : GNU All permissive License, MIT License, BSD licenses, Apple Public source licenses. Apache license

Recap

Copy left (Protective Licenses)	Permissive Licenses
Publication of software code of all modified versions or derived works under the original copyleft licenses / protective licenses	Provides No guarantee that derived works of the software will remain free and publicly available ; generally requiring only that the original copyrights notice be remained
	- Hence derived works , or future versions , of permissively – licenses software can be released as proprietary software - Permissive licenses offer highly extensive license compatibility as compared to copyleft licenses - Wider adaptability in the open source community

Recap

Choosing Open Source Licensing Models : Based on Permissions, Distribution, Modification, Use.

- Option 1: Work with a community –
  - Contributors License Agreement (CLA)
    - Simple CLAs vs Detailed CLAs
    - Individual CLAs vs Corporate CLAs
- Option 2: Keep it simple and permissive - Choose the MIT License ,
- Option 3: Need to share improvements
- Option 4: Work without a license

Why to choose Open Source Licensing Models?

- To protect works contributed in the open source domain
- To protect contributors and users from any copyright infringements
- To Interested developers or contributors or business will not touch a project without a license protection.

Recap

Open Source Business Model

1. **By Intellectual Property :**

- Dual Licensing Model –

  – Community version – under GPL

  – Enterprise version – under commercial license

- Open Core Model : Core portion of the software, which provides the basic functionality is made available as open source

2. **By Services and other Business model**

  – **Open source SaaS model** – example WordPress, Sharetribe...

  – **Selling services** : such as consulting service , technical support service, or training

  – **Donation, Users advertisement, and Merchandise**

  – **Crowd Funding and Crowd Sourcing**

  – **Freemium Business Model** - Feature limited - basic Features vs. Extended features Time limited – fixed duration after that paid versions

References and further readings

- Free software foundations https://www.fsf.org/
- GNU https://www.gnu.org/
- Open Source Initiative https://opensource.org/
- Open Source Resources https://opensource.com/
- Open Source Guides https://opensource.guide/
- Creative commons https://creativecommons.org
- Choose a license https://choosealicense.com/
- Copyleft https://copyleft.org/

References and Recommended Reading:

- How Developers Can Make money with Open Source Projects

https://rubygarage.org/blog/how-make-money-with-open-source-projects

- What Motivates a Developer to Contribute to Open-Source Software?

https://clearcode.cc/blog/why-developers-contribute-open-source-software/

- How do Open Source Programmers make money

https://www.thewindowsclub.com/open-source-companies-programmers-make-money

- **https://en.wikipedia.org/wiki/Business_models_for_open-source_software**

- Bounty Source https://bountysource.com/ for Cloud Funding
- Gunio https://gun.io/