Open Source Software Paradigms

Lecture - 01

Introduction to Open Source Software Engineering

Lecturer: Biniam Behailu
Addis Ababa Science and Technology University

Contents

Learning Objectives

By the end of this lecture, you should be able to:

- Define software engineering and its key paradigms.
- Explain the fundamental principles and characteristics of open-source software.
- Identify the historical figures and key events of the open-source movement.
- Summarize the benefits of open source for both users and developers.
- Recognize the various ways to contribute to open-source projects.
- Name several major open-source programs and projects.

Introduction

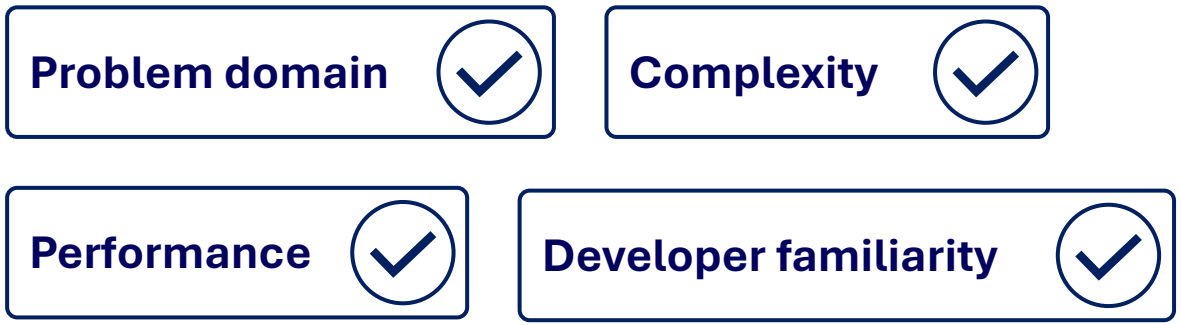
- Software engineering is an engineering based approach for the design, development, testing and maintenance of a software product.
- Its goal is to develop software applications for improving quality, budget, and time efficiency.
- It ensures that the software that has to be built should be consistent, correct also on budget, on time, and within the required requirements.

Introduction

- Software engineering paradigms define the fundamental principles, methodologies, and practices used to develop high-quality software systems.
- It defines how programmers view a problem, structure their solution, and organize the code.
- Different paradigms contributed to the growth and maturity of SWEng.
- Procedural, OOP, Functional, Agile, DevOps, SaaS (Features, Practical applications, Strength, Weakness)

Introduction

- The chosen paradigm significantly influences how developers design, write, and maintain software.
- It affects aspects like code readability, maintainability, and the overall structure of the program.
- Factors to choose the right paradigm



What is Open Source Software?

What is Open Source Software?

- Open source software (OSS) refers to software that is released with a **license** that **allows** anyone to view, modify, and distribute the source code.
- It promotes collaboration and transparency, enabling developers worldwide to contribute to projects.
- The open-source movement has revolutionized the software industry by encouraging innovation and reducing the cost of software development.

What is Open Source Software?

- Open Source software represents a new way of dealing with intellectual property.
- **Transparency**: Anyone can inspect the code.
- **Collaboration**: Community-driven development.
- **Licensing**: Governed by open-source licenses (e.g., MIT, GPL).

What is Open Source Software?

- At its core, open source is an **idea**, and it's a concept that revolves around access.
- You can access someone else's **source code** (or even a company's).
- This way, you can customize it for your **use case**, **find bugs**, and **fix them** too.
- Better still, you can **share** your "**version**" of the source code with others in addition to the original.

What is Open Source Software?

- In fact, open source is what powers the internet.
- For example, 60% of the world's websites run on **Apache** and **Nginx**, the most popular web servers.
- And yes, even **Fortune 500** companies use **Nginx** [1].

Open Source Paradigm

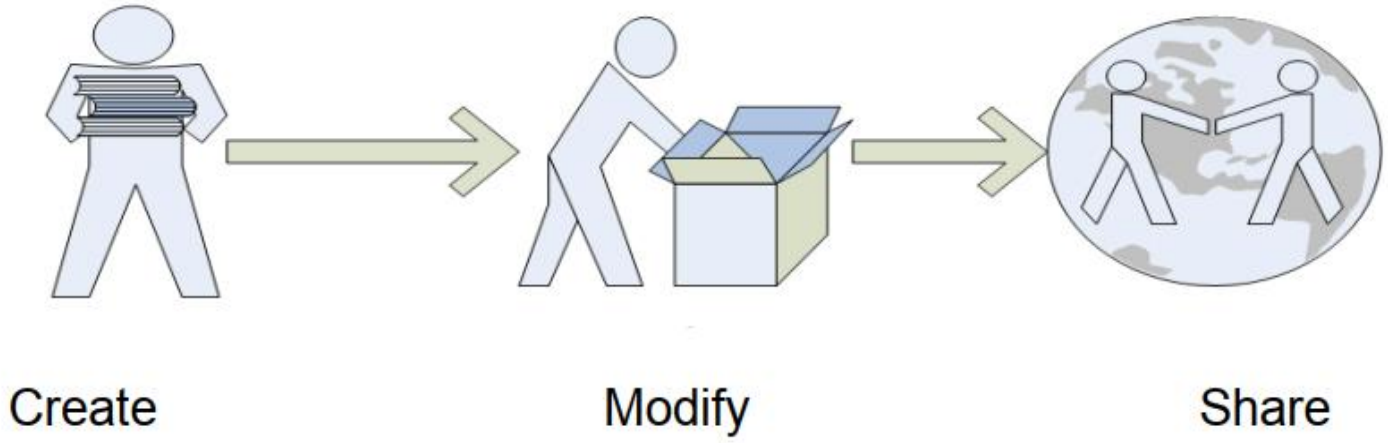
- The open-source software paradigm refers to the entire approach to software development where the source **code** is **open** and **collaboration** is encouraged.
- Source code must be **open** and **readable** for practical reasons
- It's a collaborative development paradigm or model
- Technically, OOS (Open Source Software) is defined in terms of distribution licenses not developmental methods

Open Source Paradigm

- Programmers who have access to a computer program's source code can improve that program by adding features to it or fixing parts that don't always work correctly.
- Users must have freedom (characteristics)
  - ✓ To run the program/software for any purpose
  - ✓ To study how the program works and modify it to suit their needs
  - ✓ To redistribute copies freely
  - ✓ To improve the program and release their improvements to the public

Open Source Paradigm

- Sharing and collaboration is a key of open source



Characteristics of Open Source

- **Transparency**: The open nature of the source code allows for greater scrutiny, which can lead to improved security and quality through peer review.
- **Licensing**: Open source software is released under licenses that comply with the Open Source Definition, such as the GNU General Public License (GPL) or the MIT License.
- **No Vendor Lock-in**: Users are not tied to a specific vendor, allowing for flexibility in usage and support.
- **Continuous Improvement**: The collaborative nature of open source encourages ongoing enhancements and updates from the community.

How Open Source Started?

How Open Source Started?

- Open source first started as a **movement**.

Yes, initially, open source was a “**rebellion**” **against closed source** (proprietary software whose source code is “owned” by a certain company or organization and therefore is not accessible).

How Open Source Started?

- The modern era of the movement is largely considered to have begun with **Richard Stallman**.
- He is an American programmer and activist who is best known for founding the **free software movement**.

- **The GNU Project**
- **GNU General Public License (GPL)**
- **The Free Software Movement**

Uses This, "Richard Stallman," 23-Jan-2010. [Online]. https://usesthis.com/interviews/richard.stallman/. [Accessed: 02-Sep-2025]

The Free Software Movement

- A social and political movement advocating for the freedom to use, study, modify, and distribute software.

Freedom 0
The freedom to run the software for any purpose.

Freedom 1
The freedom to study and change the software.

Freedom 2
The freedom to redistribute copies of the original software.

Freedom 3
The freedom to distribute copies of modified versions.

Uses This, "Richard Stallman," 23-Jan-2010. [Online]. https://usesthis.com/interviews/richard.stallman/. [Accessed: 02-Sep-2025]

Free software foundation

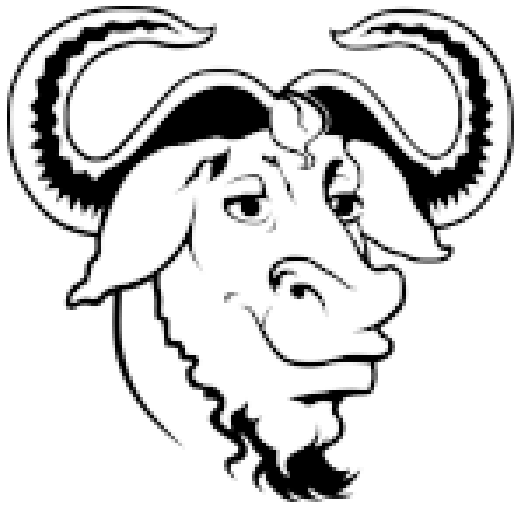
- Founded in 1985 by Richard Stallman to promote computer user freedom and defend the rights of all free software users.
- It is a nonprofit organization dedicated to promoting computer user freedom and advocating for the free software movement.
- To empower users to control their computing through free software.
- To advocate for the ethical and social implications of software freedom.

Image Source: *J. García, "The Free Software Foundation receives a donation of $1 million in bitcoins," Linux Adictos, [Online]. Available: https://en.linuxadictos.com/the-free-software-foundation-receives-a-donation-1-million-dollars-bitcoins.html. Accessed: Sep. 02, 2025*

Free software foundation

- Notable projects in the GNU project include:
  - ✓ **GNU Project**: A comprehensive set of free software tools and utilities.
  - ✓ **Licensing**: Development of the GNU General Public License (GPL), ensuring software freedom.

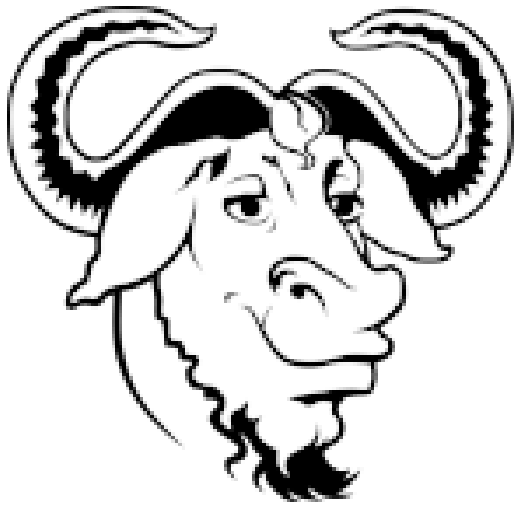
Image Source: *J. García, "The Free Software Foundation receives a donation of $1 million in bitcoins," Linux Adictos, [Online]. Available: https://en.linuxadictos.com/the-free-software-foundation-receives-a-donation-1-million-dollars-bitcoins.html. Accessed: Sep. 02, 2025*



The GNU Project

- The GNU Project, launched in 1983 by Richard Stallman, is a pioneering initiative aimed at developing a complete free **Unix-like** operating system.
- It was founded on the belief that software should be free to use, modify, and distribute, thereby empowering users with control over their computing.

Image Source: *O. Tange, "GNU Parallel," GNU.org. [Online]. Available: https://www.gnu.org/software/parallel/ . Accessed: Sep. 02, 2025.*



The GNU Project

GNU Toolkit: The project encompasses a wide range of essential tools, including:

- ✓ **GCC** (**GNU Compiler Collection**): A powerful compiler for several programming languages.
- ✓ **Emacs**: A versatile text editor that has become a favorite among developers.
- ✓ **Bash**: The Bourne Again Shell, which serves as a command-line interface for users.

GNU General Public License (GPL):

This license ensures that the software remains free and open for all users, protecting the rights of both developers and users.

Image Source: *O. Tange, "GNU Parallel," GNU.org. [Online]. Available: https://www.gnu.org/software/parallel/ . Accessed: Sep. 02, 2025.*

The Making of Linux

- Developed by **Linus Torvalds** in 1991, Linux has become one of the most significant operating systems in the world, powering everything from personal computers to servers, smartphones, and embedded systems.
- **Kernel**: The Linux kernel is the core component that manages hardware resources and provides essential services for applications and system processes.
- **GNU Tools**: To create a complete operating system, Linux relies on GNU tools and utilities, which provide essential functions like file management, text editing, and system calls.

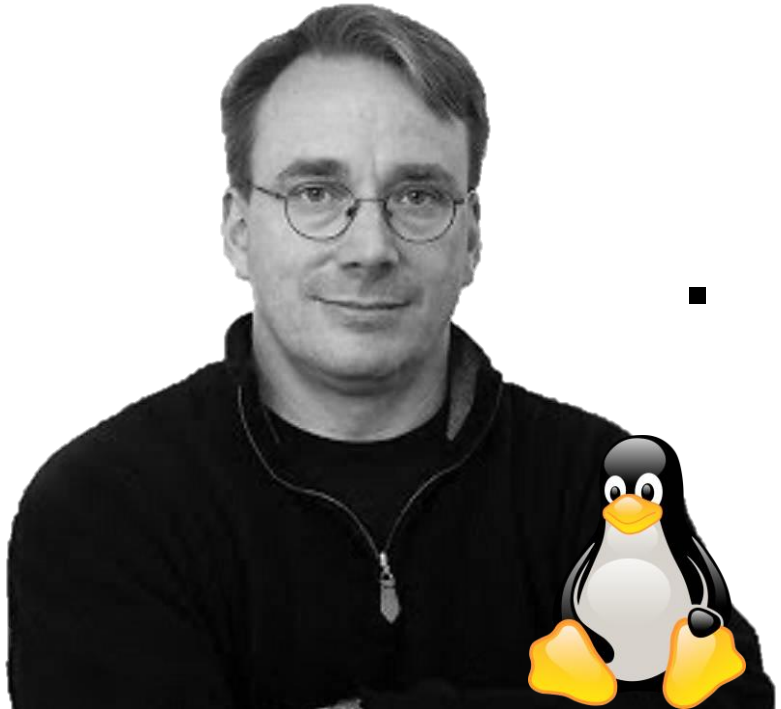


***Image source**: Editors, TheFamousPeople.com, "Linus Torvalds Biography," TheFamousPeople.com, updated June 01, 2024. [Online]. Available: https://www.thefamouspeople.com/profiles/linus-torvalds-3972.php. Accessed: Sep. 02, 2025.*

The Making of Linux

- **1991**: Release of Linux 0.01, the first version made publicly available.
- **1994**: Linux 1.0 is released, marking its maturity as a stable operating system.
- **2000s** Onward: Linux gains popularity in enterprise environments, leading to the emergence of numerous distributions (distros) like Red Hat, Ubuntu, and Debian.

***Image source**: Editors, TheFamousPeople.com, "Linus Torvalds Biography," TheFamousPeople.com, updated June 01, 2024. [Online]. Available: https://www.thefamouspeople.com/profiles/linus-torvalds-3972.php. Accessed: Sep. 02, 2025.*

History of Open source

Early Days of Computing (1950s-1970s)

Free Software Movement begins with figures like Richard Stallman advocating for software freedom

1983: Birth of Free Software

GNU Project launched by Richard Stallman to create a free Unix-like operating system. Introduction of copyleft concept

1991: Linux Revolution

Linux operating system created by Linus Torvalds, becoming a prominent example of open-source software

1998: Open Source Initiative

Term "open source" popularized by Eric Raymond's Cathedral and Bazaar model. OSI founded to promote open-source licenses

2000s-Present: Corporate Embrace

Major companies like IBM, Google, and Microsoft begin embracing and contributing to open-source projects

Advancements in Open Source

- **Linux** released to the public along with its source code (1991)
- The publication of the **Python interpreter** source code (1991)
- The launch of the **Apache HTTP Server** (1995)
- The coining of the term “**open source**” (1998)
- The release of the **Netscape browser’s** source code (1998)
- The release of **Open Office’s** source code, the free software counterpart to Microsoft Office (2000)
- The 20th century laid a strong foundation for **OSS**.

Why do developers contribute to Open Source Projects?

Why do developers contribute to Open Source Projects?

Learning & Skill Development

Learn new technologies, work on diverse codebases, gain mentorship and collaboration experience

Community & Purpose

Sense of belonging, contributing to the larger community, making a positive impact on technology

Career Advancement

Showcase coding skills, build portfolio, network with professionals, discover new opportunities

Personal Satisfaction

Hobby contribution, solving problems, personal fulfillment from helping others

Income Generation

Generate revenue through support, funding opportunities, consulting services

Why to use Open Source? (User Vs Developer)

From an end-user perspective

- Reduces the cost of software acquisition
- Security
- Flexibility and customization
- Wider choice and innovation
- Community and support

Why to use Open Source? (User Vs Developer)

From a software process and productivity perspective

- Enlarging the user community, Collaboration and recognition
- Scalable division of labor
- Short feedback loops
- Greater opportunity for analysis
- Learning and skill development
- Improved Software
- Openness and transparency
- Vender lock-in avoidance
- Faster development cycle

Types of Open Source Contributions

Code Contributions

Bug Fixes: Identifying and fixing bugs in the software.
Feature Development: Implementing new features or enhancements to existing functionality.
Refactoring: Improving the code structure and readability without changing its external behavior.

Documentation

User Manuals: Writing or editing user guides and manuals to help users understand how to use the software.
API Documentation: Providing clear and comprehensive documentation for APIs to assist developers in integrating with the software.
Contribution Guidelines: Creating or updating guidelines for how others can contribute to the project.

Types of Open Source Contributions

Testing

Unit Testing: Writing tests to verify that individual components of the software work as expected.
Bug Reporting: Identifying and reporting bugs or issues in the software, along with detailed descriptions and steps to reproduce them.
Quality Assurance: Participating in testing phases to ensure the software is stable and reliable before releases.

Design

User Interface (**UI**) **Design**: Contributing to the visual design of the software, including layout, color schemes, and user experience (UX) improvements.
Graphic Design: Creating logos, icons, and other visual assets for the project.

Types of Open Source Contributions

Translation and Localization

Language Translation: Translating documentation, user interfaces, and other materials into different languages to make the software accessible to a wider audience.
Cultural Adaptation: Adapting content to suit cultural norms and preferences in different regions.

Community Engagement

Support: Answering questions and providing support to other users through forums, chat rooms, or issue trackers.
Mentorship: Guiding new contributors by helping them understand the project and how to get involved.

Types of Open Source Contributions

Project Management

Maintaining Issues: Organizing and triaging issues in the project's issue tracker to identify priorities and assign tasks.
Release Management: Coordinating and managing the release process, including versioning and changelogs.

Advocacy and Promotion

Blogging and Writing: Writing articles, blog posts, or tutorials about the software to promote it and share knowledge with the community.
Speaking Engagements: Representing the project at conferences or meetups to raise awareness and attract new contributors.

Open Source Programs and Projects

Open Source Programs and Projects

Google Summer of Code (GSoC)

https://summerofcode.withgoogle.com/

Outreachy

https://www.outreachy.org/

MLH Felowship

https://fellowship.mlh.io/

GirlScript Summer of Code (GSSoC)

https://gssoc.girlscript.tech/

Season of KDE

https://mentorship.kde.org/sok/

Digital Ocean Hacktoberfest

https://hacktoberfest.com/

Free Software Foundation internship program

https://www.fsf.org/volunteer/internships

Open Source Programs and Projects

Web Dev Projects:

Freecodecamp: https://github.com/freeCodeCamp/freeCodeCamp

Moment: https://github.com/moment/moment

React: https://github.com/facebook/react

Gatsby: https://github.com/gatsbyjs/gatsby

Material UI: https://github.com/mui-org/material-ui

Cloud Related Projects:

- Envoy: https://github.com/envoyproxy/envoy
- Docker: https://github.com/docker
- Kubernetes: https://github.com/kubernetes/kubernetes

Open Source Programs and Projects

Machine Learning Projects:

Scikit Learn: https://github.com/scikit-learn/scikit-learn

Tensorflow: https://github.com/tensorflow/tensorflow

PyTorch: https://github.com/pytorch/pytorch

OpenCV: https://github.com/opencv/opencv

Brain Teaser

1. What is the key principle that defines open-source software?

A. It must always be distributed for a fee.

B. Its source code is publicly available for anyone to view, modify, and distribute.

C. It is developed and maintained by a single, commercial entity.

D. Its source code is only available to certified developers.

Brain Teaser

1. What is the key principle that defines open-source software?

A. It must always be distributed for a fee.

B. Its source code is publicly available for anyone to view, modify, and distribute.

The key principle of open-source software is the free and open availability of its source code to the public.

C. It is developed and maintained by a single, commercial entity.

D. Its source code is only available to certified developers.

Brain Teaser

2. Which historical figure is largely considered the founder of the modern free software movement?

A. Bill Gates

B. Richard Stallman

C. James Gosling

D. Linus Torvalds

Brain Teaser

2. Which historical figure is largely considered the founder of the modern free software movement?

A. Bill Gates

B. Richard Stallman

Richard Stallman is recognized for founding the free software movement and initiating the GNU Project.

C. James Gosling

D. Linus Torvalds

What is Open Source Software?

[1] A. Nduta, "A Brief History of Open Source," FreeCodeCamp, Apr. 3, 2023. [Online]. Available: https://www.freecodecamp.org/news/brief-history-of-open-source/#:~:text=In%20fact%2C%20open%20source%20is%20what%20powers%20the,usage%20is%20that%20there%20are%20lots%20of%20contributors. [Accessed: Sep. 2, 2025].

Thank you!

"Open source software is a testament to the power of collaboration; it transforms ideas into innovations, empowering individuals and communities to build a better future together."

Lecturer: Biniam Behailu
Addis Ababa Science and Technology University