Chair of Connected Mobility
TUM Department of Informatics
Technical University of Munich

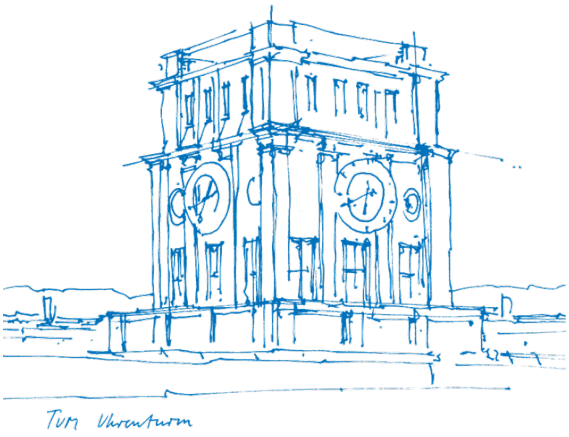
Open Source Lab

Introduction

Fabian Sauter, Christian Menges, Alexander Stephan

Chair of Connected Mobility
TUM Department of Informatics
Technical University of Munich

Garching, April 28, 2022



Important Information

- **Website:** https://www.in.tum.de/cm/teaching/summer-term-2022/open-source-lab/
- **Duration:**
  - Weekly theory lectures at the beginning.
  - Later biweekly meetings to check student's practical progress.
  - Time slots will be decided in cooperation with the participants.
  - **All lectures and meetings will be held online (virtual) using BBB, attendance is mandatory!**
- **Module ID:** IN0012 / IN2106 (Bachelor and Master practical course)
- **ECTS:** 10
- **Capacity:** 20 students
- **Language:** English (or German, in case all participants agree)

Team

Fabian Sauter

- fabian.sauter@in.tum.de
- Master Informatik
- https://github.com/com8
- https://gitlab.com/COM8
- Popular Programming Languages
  - C++
  - C#
  - C
  - Python
- Misc
  - XMPP Dev
  - Web: https://uwpx.org

Christian Menges

- christian.menges@tum.de
- https://github.com/Garfield96
- https://gitlab.com/Garfield96
- https://gitlab.lrz.de/ga87nad
- Popular Programming Languages
  - C/C++
  - Go
  - Rust
  - Python
  - Ruby

Alexander Stephan

- alexander.stephan@tum.de
- Bachelor Informatik
- https://github.com/alexanderstephan
- https://gitlab.com/alexanderstephan
- https://gitlab.lrz.de/alexanderstephan
- Popular Programming Languages
  - C/C++
  - Go
  - Java
  - TypeScript
  - Haskell

Thanks to **Sebastian Kappes** and **Martin Uhl** for their help with this course!

Outline

1 Organization

2 Projects

3 Open Source and FOSS

4 Getting Started

Course Goals

Understand what Open Source

- What is FOSS?
- How to start?
- How to maintain?
- Is GitHub supporting Open Source?

Learn how to contribute to Open Source projects:

- Creating issues.
- Creating pull request.
- Choosing a license.
- Automated testing (CI/CD).

Roadmap

Course duration: 25.04.2022 – 29.07.2022

Lectures

- Three weeks worth of lectures, every Thursday from 18:00-20:00.
- Week 1: Introduction and Git basics
- Week 2: Utilities and CI/CD
- Week 3: GitHub, GitLab and Licenses

Reports

- Starting at week 4, biweekly
- **No** slides needed.
- Show us what **you** have done in the last two week and what your plans are for the next two weeks.
- **Max.** 6 minutes. We will interrupt you!
- Please keep the PR selection on Moodle up to date.

1 ECTS $\widehat{=}$ 30 working hours[1]
300 working hours for this course / 15 weeks = 20 hours per week

[1] https://www.ma.tum.de/en/studies-information/study-programs-mathematics/Calculation-credits-grades.html

Grading

- No final report required
- All interesting topics should be described in the documentation of the projects or the corresponding PR.
- LOC not relevant
- Intermediate presentation (no fancy slides required)
- Code quality
- Interaction with the community
- Interaction with the advisors

Note: Spamming or creating other unnecessary burden to the community will result in failing the course immediately. Remember, **you are representing TUM.**

Outline

1 Organization

2 Projects

3 Open Source and FOSS

4 Getting Started

Requirements for Projects

- Open Source (must be open-contribution)
- No ”personal” projects
- Active user base
- At least 10 active users (1000+ recommended)
- The project should be large enough to last you at least a couple of weeks.
- Contributions can be new features, bug fixes or performance improvements (PRs fixing typos are not accepted by us)
- Without previous experience working on extremely large and complex projects, such as GCC, Linux Kernel, Postgres, etc. is not recommended.
- We recommend to pick one of the projects listed below, since these projects are in widespread use and we can help you in case of problems.

Project suggestions: https://www.moodle.tum.de/mod/page/view.php?id=2034441

Outline

1 Organization

2 Projects

3 Open Source and FOSS

4 Getting Started

Definitions

What is open source?

Definitions

What is open source?

> *The term open source refers to something people can modify and share because its design is publicly accessible.*

Definitions

What is open source?

> *The term open source refers to something people can modify and share because its design is publicly accessible.*

What is open source software?

Definitions

What is open source?

> *The term open source refers to something people can modify and share because its design is publicly accessible.*

What is open source software?

> *Open source software is software with source code that anyone can inspect, modify, and enhance.*

Source: https://opensource.com/resources/what-open-source

The open source way

Following the "*open source way*" requires the following principles:

Transparency:

Collaboration:

Release early and often:

Inclusive meritocracy:

Community:

Source: https://opensource.com/open-source-way

The open source way

Following the "*open source way*" requires the following principles:

Transparency: Everyone has access to the information and materials necessary for doing our best work.

Collaboration:

Release early and often:

Inclusive meritocracy:

Community:

Source: https://opensource.com/open-source-way

The open source way

Following the "*open source way*" requires the following principles:

Transparency: Everyone has access to the information and materials necessary for doing our best work.

Collaboration: Work together and implement open standards so everyone else is able to contribute in the future.

Release early and often:

Inclusive meritocracy:

Community:

Source: https://opensource.com/open-source-way

The open source way

Following the "*open source way*" requires the following principles:

Transparency: Everyone has access to the information and materials necessary for doing our best work.

Collaboration: Work together and implement open standards so everyone else is able to contribute in the future.

Release early and often: Rapid prototypes can lead to rapid discoveries.

Inclusive meritocracy:

Community:

Source: https://opensource.com/open-source-way

The open source way

Following the "*open source way*" requires the following principles:

Transparency: Everyone has access to the information and materials necessary for doing our best work.

Collaboration: Work together and implement open standards so everyone else is able to contribute in the future.

Release early and often: Rapid prototypes can lead to rapid discoveries.

Inclusive meritocracy: The best ideas should win. Include diverse perspectives.

Community:

Source: https://opensource.com/open-source-way

The open source way

Following the "*open source way*" requires the following principles:

Transparency: Everyone has access to the information and materials necessary for doing our best work.

Collaboration: Work together and implement open standards so everyone else is able to contribute in the future.

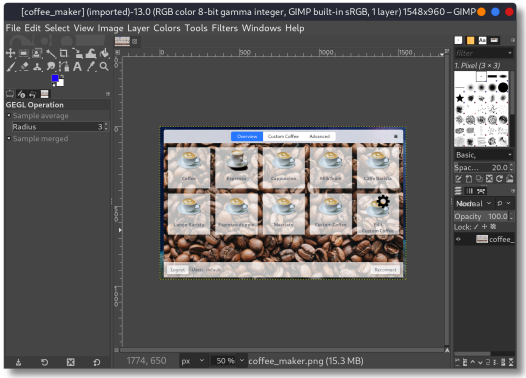
Release early and often: Rapid prototypes can lead to rapid discoveries.

Inclusive meritocracy: The best ideas should win. Include diverse perspectives.

Community: Shared values guide decision making.

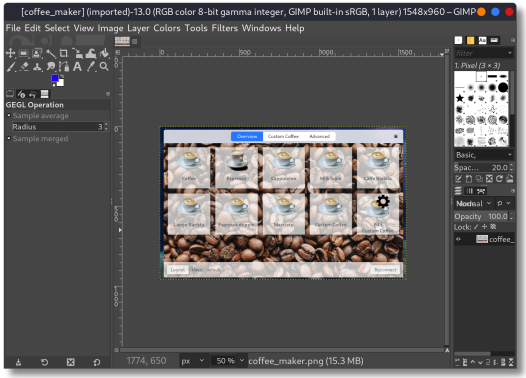
Source: https://opensource.com/open-source-way

Example: GIMP



Is GIMP ”open source”?

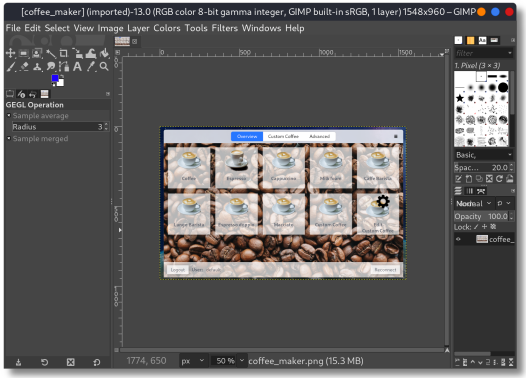
Example: GIMP



Is GIMP ”open source”?

- **Can be modified:** Yes
- **Can be shared:** Yes
- **Source Code**: https://www.gimp.org/source/#gimp-source-code

Example: GIMP

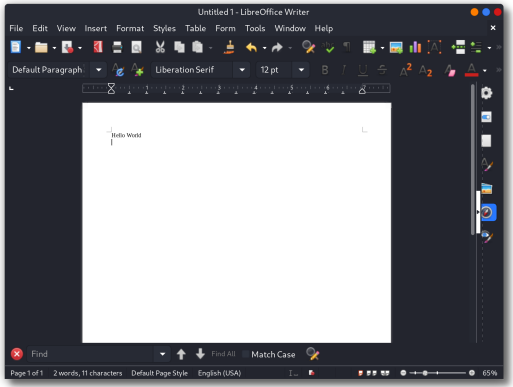


Is GIMP ”open source”?

- **Can be modified:** Yes
- **Can be shared:** Yes
- **Source Code**: https://www.gimp.org/source/#gimp-source-code

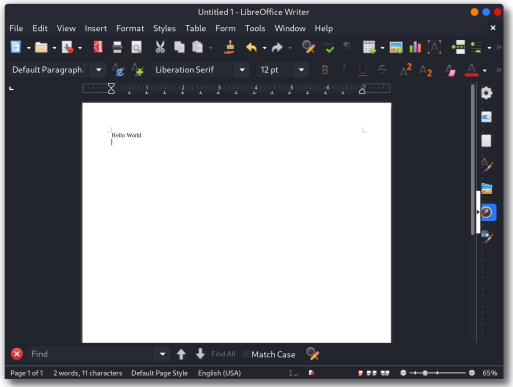
$\Rightarrow$ YES

Example: LibreOffice



Is LibreOffice ”open source”?

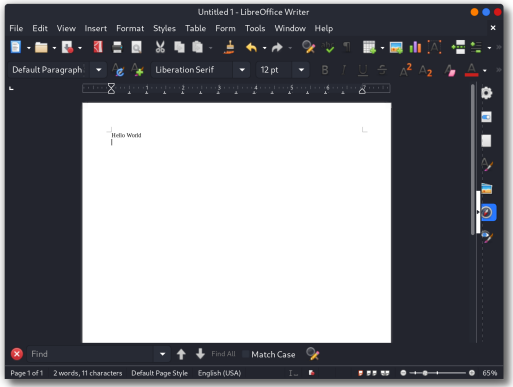
Example: LibreOffice



Is LibreOffice ”open source”?

- **Can be modified:** Yes
- **Can be shared:** Yes
- **Source Code**: hhttps://gerrit.libreoffice.org/admin/repos

Example: LibreOffice

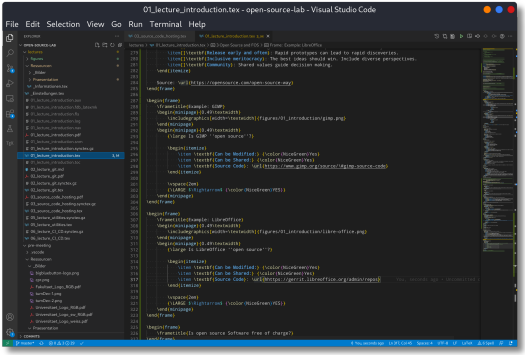


Is LibreOffice ”open source”?

- **Can be modified:** Yes
- **Can be shared:** Yes
- **Source Code**: hhttps://gerrit.libreoffice.org/admin/repos

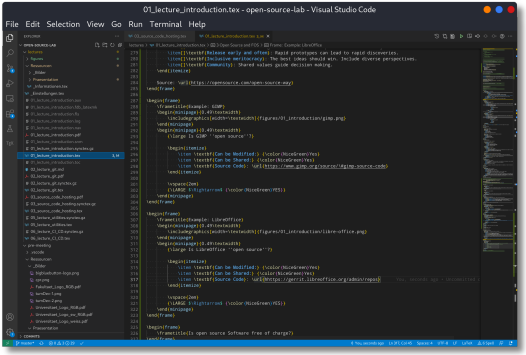
$\Rightarrow$ YES

Example: Visual Studio Code



Is Visual Studio Code ”open source”?

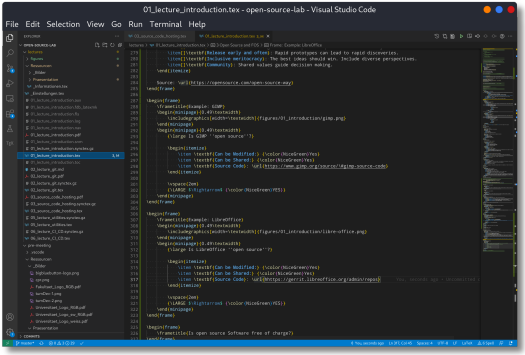
Example: Visual Studio Code



Is Visual Studio Code ”open source”?

- **Can be modified:** Yes
- **Can be shared:** Yes
- **Source Code**: https://github.com/microsoft/vscode

Example: Visual Studio Code

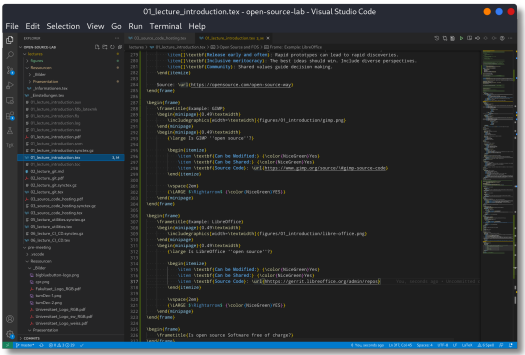


Is Visual Studio Code ”open source”?

- **Can be modified:** Yes
- **Can be shared:** Yes
- **Source Code**: https://github.com/microsoft/vscode

$\Rightarrow$ YES

Example: Visual Studio Code



Is Visual Studio Code ”open source”?

- **Can be modified:** Yes
- **Can be shared:** Yes
- **Source Code**: https://github.com/microsoft/vscode

⇒ YES, but actually NO!

The actual executable (binary) you download is **not** open source since it is licensed under a not-FOSS license and contains telemetry/tracking.
Alternative: VSCodium: https://vscodium.com/

Is Open Source Software (OSS) free of charge?

Is Open Source Software (OSS) free of charge?

NO.

What is FOSS?

FOSS stands for: **Free**-and-Open-Source-Software

> *In simple terms, FOSS is software that all allows users to not only freely run the program for any purpose, but also provides users access to the code. Moreover, it also allows them to modify as they wish, as well as freely distribute copies of the original version or their altered version.*

Source: https://www.fosslinux.com/25393/what-is-foss-and-how-does-it-differ-from-freeware.htm

TLDR: If you have to pay for it, it usually is not FOSS.

Example: Linux Kernel

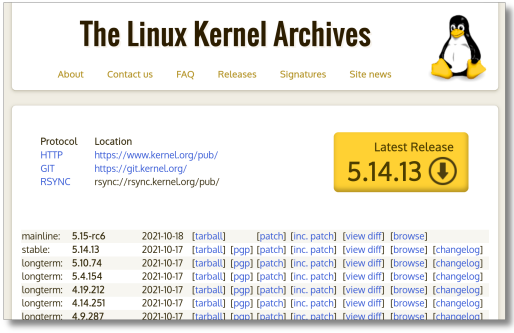
Is Linux Kernel ”FOSS”?

Example: Linux Kernel

Is Linux Kernel ”FOSS”?

- **Is is open Source:** Yes
- **Is it free of charge:**: Yes

Example: Linux Kernel



Is Linux Kernel ”FOSS”?

- **Is is open Source:** Yes
- **Is it free of charge:**: Yes

$\Rightarrow$ YES

But only since it’s under the GNU GPL (General Public License).
This happened one year after its initial release.

> *Anybody who tells me I can’t use a program because it’s not open source, go suck on rms. I’m not interested. 99% of that I run tends to be open source, but that’s my choice, dammit.*

Source: Linus Torvalds

Example: Conversations

Is Conversations ”FOSS”?

Example: Conversations

Is Conversations ”FOSS”?

- **Is it open source:** Yes
- **Is it free of charge:**: No

Example: Conversations

Is Conversations ”FOSS”?

- **Is it open source:** Yes
- **Is it free of charge:**: No

$\Rightarrow$ NO

Example: Conversations

Is Conversations ”FOSS”?

- **Is it open source:** Yes
- **Is it free of charge:**: No

$\Rightarrow$ NO

An other popular open source but not FOSS example is Redhat Enterprise Linux. It is open source, but you need a license to properly use it.
To make money, open source companies usually sell paid support for their FOSS software.

Outline

1 Organization

2 Projects

3 Open Source and FOSS

4 Getting Started

Contributing

Some guidelines for contributing to projects (issues/PRs/...).

- Not fluent in English? Use a translator like DeepL or Google Translate.
- What is the expected behavior?
- What is the actual behavior?
- What environment are your working in?
- If available, use Markdown[2] for highlighting!
- Give a bit of context and some examples like a small project or screenshots.

Further information: https://developers.google.com/blockly/guides/modify/contribute/write_a_good_issue

Take your time!
A well written and formatted issue increases your chances for a fast reply drastically!

[2] https://guides.github.com/pdfs/markdown-cheatsheet-online.pdf

Contributing

Some examples. . .

- https://github.com/libcpr/cpr/issues/605
- https://github.com/libcpr/cpr/issues/595
- https://github.com/libcpr/cpr/issues/627
- https://github.com/libcpr/cpr/issues/611
- https://github.com/libcpr/cpr/issues/607