Chair of Connected Mobility
TUM School of Computation, Information and Technology
Technical University of Munich

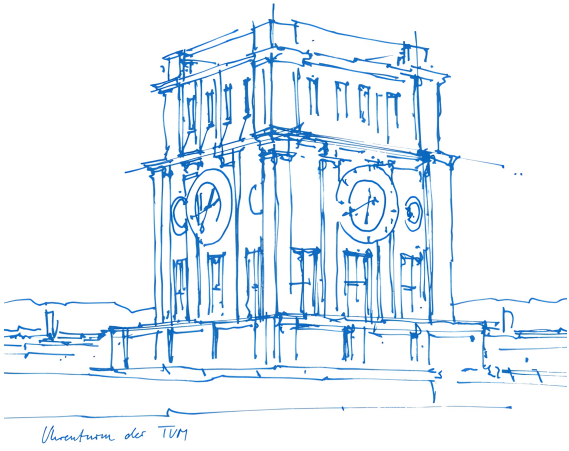
Open Source Lab

Introduction

Fabian Sauter, Christian Menges, Alexander Stephan

Chair of Connected Mobility
TUM School of Computation, Information and Technology
Technical University of Munich

Garching, October 16, 2023



Important Information

- **Website:** https://www.ce.cit.tum.de/cm/teaching/winter-term-2023-24/open-source-lab/
- **Duration:**
  - Weekly theory lectures at the beginning.
  - Later biweekly meetings to check students' 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

Team

Fabian Sauter

- fabian.sauter@in.tum.de
- R&D Embedded Software Engineer at APSensing
- https://github.com/com8
- https://gitlab.com/COM8
- Popular Programming Languages
  - C/C++
  - C#, Python, Go
- Misc
  - GTK, Bluetooth
  - Reverse Engineering
  - Linux Distro Development
  - IT Security

Christian Menges

- christian.menges@tum.de
- Cloud Software Engineer at SAP
- https://github.com/Garfield96
- https://gitlab.com/Garfield96
- Popular Programming Languages
  - C/C++
  - Go
  - Rust
  - Python
  - Ruby
- Misc
  - Kubernetes
  - Performance Engineering

Alexander Stephan

- alexander.stephan@tum.de
- Master Informatik
- https://github.com/alexanderstephan
- https://gitlab.lrz.de/alexanderstephan
- Popular Programming Languages
  - C/C++
  - Go
- Misc
  - TUM-Live
  - Cloud / DevOps

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

Outline

Course Goals

Understand Open Source:

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

Course Goals

Understand 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).

Course Goals

Understand 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).

Have fun contributing to your favorite Open Source project while getting paid for it with 10 ECTS :)

Roadmap

Course duration: 16.10.2023 - 09.02.2024

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

Roadmap

Course duration: 16.10.2023 - 09.02.2024

Lectures

- **Week 1:** Introduction, Git Basics and Getting Started
- **Week 2:** Open Source, FOSS and Advanced Git
- **Week 3:** First Presentation Session
- **Week 4:** Open Source Platforms (e.g., GitHub, GitLab) and Licenses
- **Week 5:** Second Presentation Session
- **Week 6:** Utilities and CI/CD
- **Week 7 until the end:** Biweekly Progress Report Presentations

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

Roadmap

Course duration: 16.10.2023 - 09.02.2024

Lectures

- **Week 1:** Introduction, Git Basics and Getting Started
- **Week 2:** Open Source, FOSS and Advanced Git
- **Week 3:** First Presentation Session
- **Week 4:** Open Source Platforms (e.g., GitHub, GitLab) and Licenses
- **Week 5:** Second Presentation Session
- **Week 6:** Utilities and CI/CD
- **Week 7 until the end:** Biweekly Progress Report Presentations

Reports

- Starting at week 3, biweekly
- **No** slides needed.
- Show us what **you** have done in the last two weeks and what your plans are for the next two weeks.
- **Max.** 6 minutes. We will interrupt you!
- Please keep the PR selection in the Nextcloud up to date.
- Zulip reports start next week.

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

Roadmap

Course duration: 16.10.2023 - 09.02.2024

Lectures

- **Week 1:** Introduction, Git Basics and Getting Started
- **Week 2:** Open Source, FOSS and Advanced Git
- **Week 3:** First Presentation Session
- **Week 4:** Open Source Platforms (e.g., GitHub, GitLab) and Licenses
- **Week 5:** Second Presentation Session
- **Week 6:** Utilities and CI/CD
- **Week 7 until the end:** Biweekly Progress Report Presentations

Reports

- Starting at week 3, biweekly
- **No** slides needed.
- Show us what **you** have done in the last two weeks and what your plans are for the next two weeks.
- **Max.** 6 minutes. We will interrupt you!
- Please keep the PR selection in the Nextcloud up to date.
- Zulip reports start next week.

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 - most of the time.
- Intermediate presentation (no fancy slides required).
- Code quality.
- Interaction with the community.
- Interaction with the advisors.

Grading

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

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

Grading

- Reports and Amount 50%
  - Your biweekly reports.
  - Communication with us in case something goes wrong.
  - Are you able to keep your report below **max.** 6 minutes?
  - The amount is only relevant in case the amount of code you produce is far less than we expect (compared to other students).
  - Default: 50% Reports and 0% Amount but can shift to 20% Reports and 30% Amount. (Amount $\approx$ LOC)

Grading

- Reports and Amount 50%
  - Your biweekly reports.
  - Communication with us in case something goes wrong.
  - Are you able to keep your report below **max.** 6 minutes?
  - The amount is only relevant in case the amount of code you produce is far less than we expect (compared to other students).
  - Default: 50% Reports and 0% Amount but can shift to 20% Reports and 30% Amount. (Amount $\approx$ LOC)
- Code Quality 30%
  - Linting, formatting, . . .
  - Dead code?
  - Commented out “TODO” code.
  - Correct license for used resources e.g. icons.

Grading

- Reports and Amount 50%
  - Your biweekly reports.
  - Communication with us in case something goes wrong.
  - Are you able to keep your report below **max.** 6 minutes?
  - The amount is only relevant in case the amount of code you produce is far less than we expect (compared to other students).
  - Default: 50% Reports and 0% Amount but can shift to 20% Reports and 30% Amount. (Amount $\approx$ LOC)
- Code Quality 30%
  - Linting, formatting, . . .
  - Dead code?
  - Commented out “TODO” code.
  - Correct license for used resources e.g. icons.
- General PR Quality 20%
  - Interaction with the community.
  - Do you react to suggestions/reviews in time?

Outline

Project Requirements

- Open Source (must be open-contribution)
- No “personal” projects
- Active user base
- At least 10 active users (1000+ recommended)
- 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 picking 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=2751671

Outline

Idea of Open Source

The term open source refers to something people can modify and share because its design is publicly accessible.[2]

[2] https://opensource.com/resources/what-open-source

Is It Just Software?

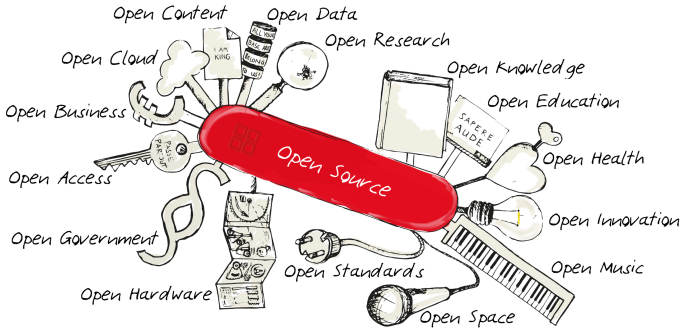


Figure 1 Open Source Swiss Knife[3]

[3] https://upload.wikimedia.org/wikipedia/commons/c/c7/121212_2_OpenSwissKnife.png

OSS in a Nutshell

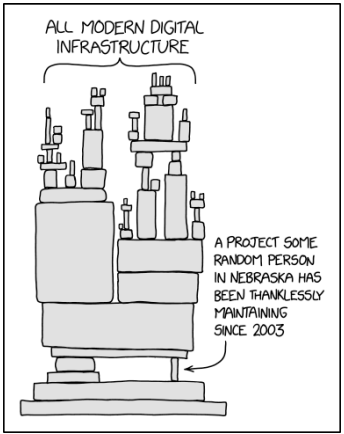


Figure 2 Cartoon about OSS[4]

[4] https://xkcd.com/2347/

Role of Open Source in Industry

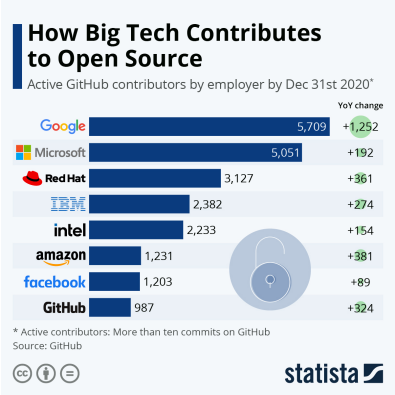


Figure 3 Statistic about OSS contributions by big companies[5]

[5] https://www.statista.com/chart/25795/active-github-contributors-by-employer/

Outline

Origins of the “Open Source Idea”

- **Patent for Two-Stroke-Engines** by George B. Selden in the 19th century
- Emergence of a **monopoly**
- 1911 Henry Ford challenged the patent successfully
- Foundation of the *Automobile Manufacturers Association*
- Members agreed to share patents from now on

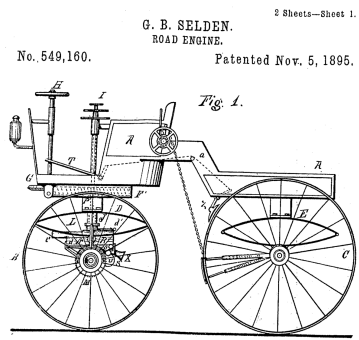


Figure 4 Road Engine Patent

Computer Age

- Software and hardware was primarily developed by research facilities (strong emphasis of **Openness** und **Exchange**)
- Hacker and DIY scene (users were also developers)
- **Milestone**: First operating systems, e.g., `UNIX`

Figure 5
Brian Kernighan demonstrates UNIX

Source: https://youtu.be/tc4ROCJYbm0

Commercialization of Software

Who can afford to do professional work for nothing? [6]

- Bill Gates

- **Emergence of a software industry** due to cheaper and more flexible hardware
- **An open letter to hobbyists**: *Altair BASIC* was copied
- Different concepts emerged:
  - **Proprietary Software**: property of a company, source code not publicly available, distribution and modification is prohibited, e.g., EULA license for Microsoft XP
  - **Open Source Software**: property of the community, source code is publicly available, distribution and modification is allowed, e.g., GPL license for Linux

[6] http://www.blinkenlights.com/classiccmp/gateswhine.html