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

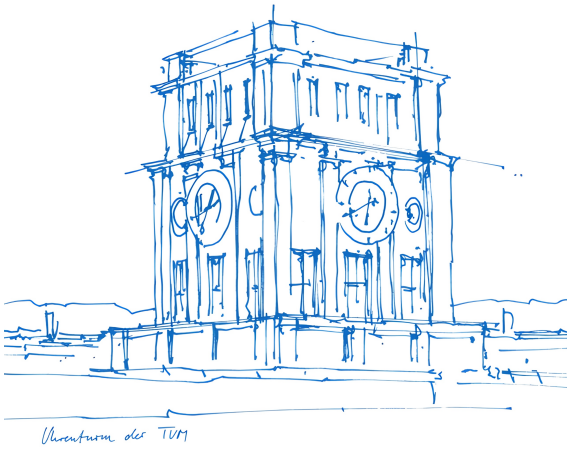
Open Source Lab

Pre Meeting

Fabian Sauter, Christian Menges, Alexander Stephan

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

Garching, July 9, 2024



Team

Fabian Sauter

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

Christian Menges

- christian.menges@tum.de
- Cloud Software Engineer
- 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!

Important Information

- **Website:** https://www.ce.cit.tum.de/cm/teaching/winter-term-2024-25/open-source-lab/
- **Registration:** 12.07.-16.07.2024 using matching.in.tum.de
- **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 (at least one session takes place in the evening (6 - 8 pm).
  - **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:** 10 students
- **Language:** English

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: 01.10.2024 - 31.03.2025

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.

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

[1] https://www.cit.tum.de/en/cit/studies/degree-programs/calculation-credits-grades/

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 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 by far less than we expect (compared to other students).
  - Default: 50% Reports and 0% Amount but can shift to 20% Reports and 30% Amount.
- Code Quality 30%
  - Linting, formatting, . . .
  - Dead code?
  - Commented out “TODO” code.
- General PR Quality 20%
  - Interaction with the community.
  - Do you react to suggestions/reviews in time?

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.

Previous Projects

- TUM-Dev: eat-api
- TUM Campus App - Android/iOS
- TUM-Live
- Linux Kernel
- Kernel Address Sanitizer (KASAN)
- Zulip
- Synapse (Matrix-Server)
- Element (Matrix-Client)
- nextcloud/server
- spot (Spotify client)
- Cataclysm: Dark Days Ahead
- Amaze File Manager
- Amaze File Utilities
- AI-on-the-edge-devic
- Logisim-evolution
- super productivity
- spicetify/marketplace
- CPR
- Swift-DocC
- Mealie
- j-lawyer-org
- calibre-web
- Xournal++
- Haaukins
- Jina
- raylib
- Gnome Project
- MUI Core
- pandas
- hex
- LLVM
- AppFlowy
- ruff
- rustlings
- uutils/coreutils
- Chromium
- Groovy
- Atrium
- NewPipe
- Owncast
- Anki
- Anki-Android
- Django
- scikit-learn
- Mainsail
- PrimeVue
- pygount
- Flow Launcher
- Mockito
- Manim
- i3status-rust
- openmediavault
- Openreads
- RiscEmu
- Librsvg
- Sway
- Waybar
- WLED
- xtgeo
- wl-clipboard
- twenty
- Nokogiri
- Minetest
- Ninja
- MLflow
- openj9
- FORT Validator
- Stirling-PDF
- dqlite
- Home Assistant
- matplotlib
- Tachiyomi
- kafka-ui
- ArchUnit
- drip
- OpenTTD
- OpenRCT2
- phpMyAdmin
- Haystack
- libobjc2
- BigBlueButton
- Geany
- dowhy
- Artemis
- . . .

Next steps

1. Drop us a **SHORT** email to get matched.
   - ☐ **With** at least one project you want to contribute to during the course. Be realistic!
   - ☐ Do you already have experience (we try to mix)? Any references e.g., Git{Hub, Lab}, Mailing Lists, ...
   - ☐ We will filter and prioritize internally since we expect to get ≥ 150 emails from interested students.
   - ☐ We will reply.
   - ☐ **Due date: 16.07.2024, 23:59**
2. Register for the course in the matching system.
3. If you are successfully matched to our course, we will send you an email with further information.
4. Familiarize yourself with the projects **during the semester break**, especially for larger projects.

Contact: opensourcelab@cm.in.tum.de

Remember to use your TUM mail address!