CodeRefinery (phase 3) report
version: 2025-03-08-13:30

Radovan Bast and Samantha Wittke

2025-03-18

Table of contents

1 Abstract

This report summarizes the achievements, feedback, and lessons learned during the phase 3 of the CodeRefinery project (March 2022 – February 2025).

2 Project partners and budget

The project operated with a budget of 38 person-months (PM) per year for 3 years (March 2022 – February 2025), with contributions from the following 8 funding partners:

- Nordic e-Infrastructure Collaboration (NeIC): 11 PM/year for project management, workshop coordination, and community management. NeIC assigned the Project owner (Tomasz Malkiewicz, CSC – IT Center for Science, Finland) to support the project manager and to chair the steering group. NeIC also provided administrative coordination following the PPS model, set up collaboration- and service agreeements, and processed invoices.

- Aalto University, Finland: 10 PM/year for lesson development, instructing, helping, outreach, streaming and recording, and transition to online teaching at scale.

- Center for Humanities Computing (CHC/DeiC), Denmark: 2 PM/year for instructing, helping, and outreach. The Danish e-Infrastructure Consortium (DeiC) operated the CodeRefinery GitLab service.

- CSC – IT Center for Science, Finland: 3 PM/year for lesson development, instructing, helping, and outreach. CSC provided the training coordinator and later the community manager (funded by NeIC).

- EuroCC National Competence Center Sweden (ENCCS): 2 PM/year for co-organizing events which were either led by ENCCS or CodeRefinery or Aalto Science-IT or PDC Stockholm, on Python, containers, GPU programming, and build tools.

- Norwegian Research Infrastructure Services (NRIS)/Sigma2, Norway: 6 PM/year for lesson development, instructing, helping, and outreach. NRIS/Sigma2 provided the project manager (funded by NeIC).

- USIT/University of Oslo, Norway: 2 PM/year for lesson development, instructing, helping, and outreach.

- National Academic Infrastructure for Supercomputing in Sweden (NAISS): 6 PM/year. NAISS and has supported CodeRefinery by providing teachers and helpers for CodeRefinery events as part of NAISS training. Staff based in Sweden has over the years contributed to lesson development, teaching, and hosting of individual events, on-site as well as online.

In addition to the in-kind contributions, the project received a **travel/conference budget** of 110k NOK/ year contributed by NeIC.

3 Project goals

3.1 Background and motivation

CodeRefinery is a **community-driven initiative designed to empower students, researchers, staff, and research software engineers** with the skills and tools needed to develop high-quality, efficient, reusable, and reproducible research software. By focusing on practical training in areas such as version control, documentation, testing, sharing, and reproducible workflows, we bridge the gap between research and software engineering. Our mission is to foster collaboration, improve research output, and promote open-source principles across academic disciplines. Open science is an important goal for universities and research institutions, and CodeRefinery is a key component in working towards this goal.

Since its beginning in 2016, CodeRefinery has contributed to enhancing the software development capabilities of the academic community in the Nordics and beyond. Over the past eight years, we have developed a comprehensive, openly maintained curriculum of lessons and courses that has been taught at over 40 workshops. **We reach circa 500 learners each year at all career stages and across all disciplines** and with a diverse range of programming expertise. We achieve this with the help of around 30 instructors, organizers, and facilitators.

CodeRefinery has cultivated a vibrant, collaborative community of instructors, learners, helpers, and partner organizations. Together, **we have built a robust cross-border training network** and a foundation for long-term improvements in research software development.

3.2 Project objectives

The following objectives were set for the phase 3 of the project:

- Organize and deliver workshops and training events (online and in-person)
- Instructor training and development
- Develop and maintain a lesson portfolio
- Build a community and network of instructors and volunteer helpers
- Operate and support a GitLab service for Nordic research software
- Support the community of Nordic research software engineers
- Develop a governance model for CodeRefinery as a community-driven organization
- Develop a business model for a sustainable development and delivery of training events

We will summarize our achievements and lessons learned in the results section.

The following project objectives were not achieved:

- Establishing CodeRefinery as an independent organization: Instead the steering group and the project team favored to continue the project using the in-kind funding model, without creating a separate legal entity.
- Supporting the Carpentries organization by paying a membership fee and by providing a focal point for Nordic Carpentries activities: We didn't have the necessary budget to fund a Carpentries training coordinator and the membership fee. The project manager did not have the administrative bandwidth to follow-up and coordinate Carpentries instructor training for the Nordic region.

3.3 Related documents and resources

For more information about the history of the project, the project proposal, and the project plan, we refer to the following documents:

- [History of the project](#)
- [Proposal to the NeIC open call 2021 (phase 3 proposal)](#)
- [NeIC open call 2021 evaluation report](#)
- [Collaboration agreement (phase 3)](#)
- [Project proposal (phase 4)](#) (submitted 2025-01-23)

4 Results

4.1 Training events

Our past events and participation statistics are summarized on the web:

- https://coderefinery.org/workshops/
- https://coderefinery.org/about/statistics/

During this phase of the project we have delivered:

- 6 online CodeRefinery tools workshops to over 2000 registered participants. These workshops consist of 6 half-days of training, 2 on-boarding days, and sometimes also 2 bring-your-own-code sessions. These events involve 5-10 instructors, helpers, and exercise leads.

    - [Online, Sep 10–19, 2024](#)
    - [Online, Mar 12–21, 2024](#)
    - [Online, Sep 19–28, 2023](#)
    - [Online, Mar 21–23, 2023](#)
    - [Online, Sep 20–29, 2022](#)
    - [Online, Mar 22–23, 2022](#)

- 5 in-person CodeRefinery tools workshops (3 days).

    - [Tromso, Jan 27–30, 2025](#)
    - [Aachen, Sep 16–18, 2024](#)
    - [Gothenburg, Aug 27–29, 2024](#)
    - [Sozopol, Jun 25–27, 2024](#)
    - [Uppsala, May 9–11, 2023](#)

- 1 instructor training event (online; 4 sessions over 4 weeks).

    - Online, Aug 13, Aug 20, Aug 27, Sep 3, 2024

- 3 online Python for Scientific Computing workshops (3-4 half-days each).

    - Online, Nov 5–7, 2024
    - Online, Nov 7–10, 2023
    - Online, Nov 22–25, 2022

- Intro to Scientific Computing / HPC Kickstart.

    - Online, Feb 26, 2025
    - Online, Jun 4–6, 2024
    - Online, Jun 6–8, 2023
    - Online, Jun 7–10, 2022

- 1 Build Systems Course (online; 4 half-days).

    - Online, Oct 8–11, 2024

- Tuesday Tools & Techniques for High Performance Computing (online; 4 sessions over 4 weeks).

    - Online, Apr 16, Apr 23, May 7, May 14, 2024

Most of our events have the size and planning effort of a small conference.

To be able to deploy so many events and reach so many participants and contributors we have worked on the following improvements:

- Simplified the registration process.
- Simplified the naming of different roles and responsibilities in order to make it easier for new people to join the community and have a common understanding.
- Created a process for participants to send in a short report after the workshop to make it easier for us to hand out participation certificates.
- Collect feedback after each workshop day and publish it as part of the event.
- Use collaborative notes for questions and answers during the workshop and publish these as part of the event.

- "Bring your own classroom" concept to make it easier for organizations to join the workshop with their own participants.
- "Bring your own code" sessions to discuss and work on participants' own projects.
- Most workshops are streamed and recorded and we are typically able to publish recordings within the same day.
- Most of our workshops use co-teaching to improve the quality and the on-boarding experience for new instructors.

4.2 Post-workshop survey results

In early 2024 we sent out a survey to participants of our workshops from 2022 and 2023. We received 129 responses and this blog post summarizes the findings.

The survey results with all details and additional questions have been summarized in a blog post: https://coderefinery.org/blog/2024/08/10/post-workshop-survey/

You can also browse the survey questions and the Jupyter notebook which generated the plots. We have archived the results on Zenodo.
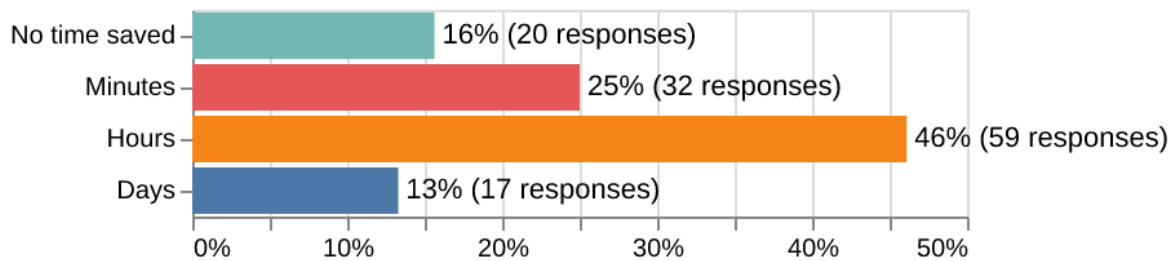
Figure 1: Plot estimating time saving
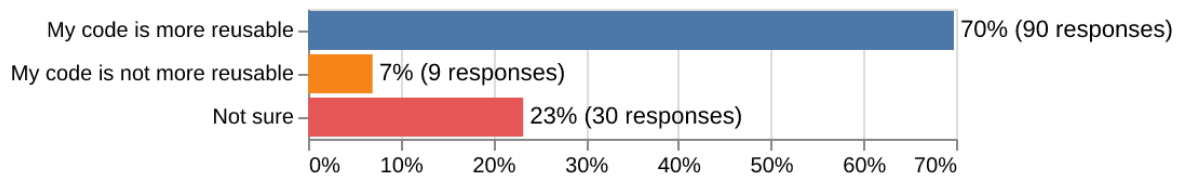
After attending the workshop, would you judge your code to be more reusable or not more reusable?

Figure 2: Plot about whether code is more reusable
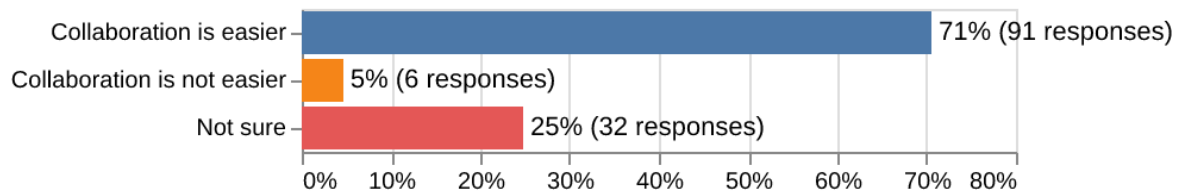
After attending the workshop, has it become easier or not for you to collaborate on software development with your colleagues and collaborators?

Figure 3: Plot about whether collaboration is easier
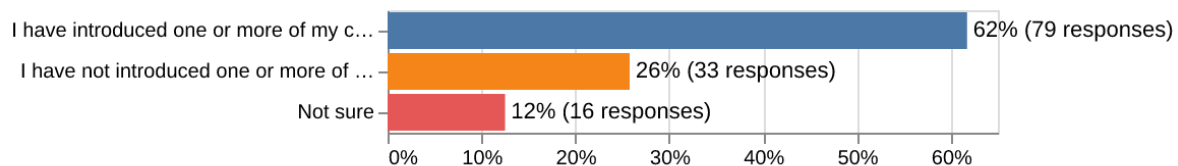
Have you introduced one or more of your colleagues to new tools or practices as a result of the workshop?

Figure 4: Plot about whether colleagues have been introduced

10 — 50% (64 responses)
9 — 16% (21 responses)
8 — 19% (24 responses)
7 — 9% (11 responses)
6 — 2% (3 responses)
5 — 1% (1 response)
4 — 1% (1 response)
3 — 2% (3 responses)
0 — 1% (1 response)
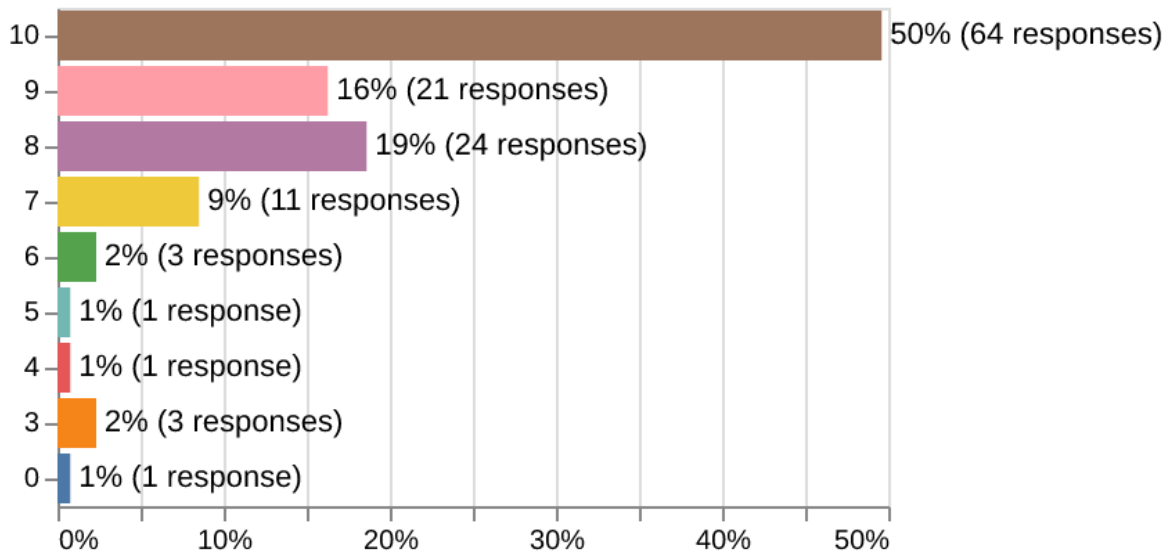
How likely is it that you would recommend CodeRefinery workshop to a friend or colleague?

0 means definitely not. 10 means definitely yes.

Figure 5: How likely are you to recommend?

I would prefer pre-recorded lectures c... — 28% (36 responses)
I would prefer online teaching (lecture... — 48% (62 responses)
I would prefer in-person teaching — 23% (30 responses)
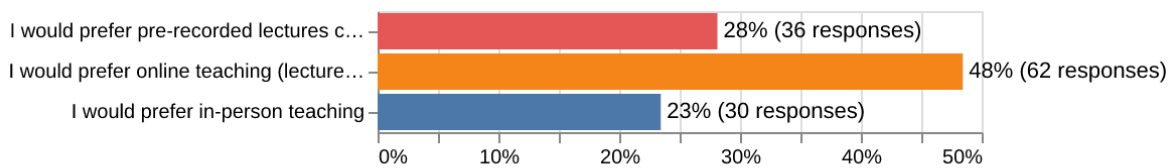
Would you prefer pre-recorded lectures, live online teaching, or in-person teaching?

Figure 6: Would you prefer pre-recorded lectures, live online teaching, or in-person teaching?
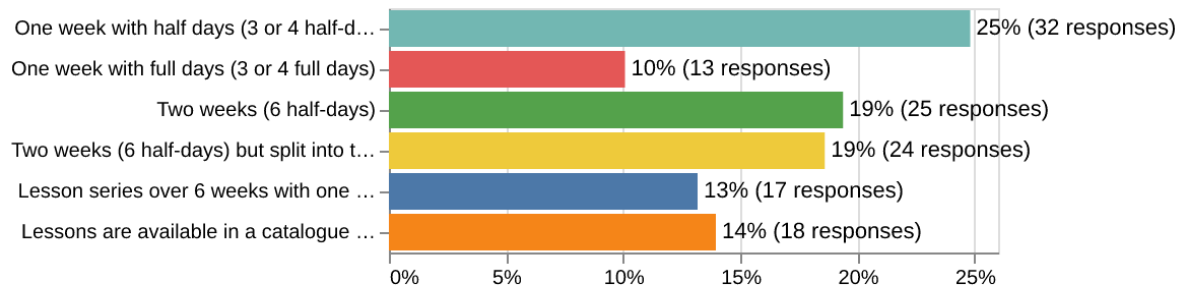
What would be the ideal format for you?

Figure 7: What would be the ideal format for you?

Participation style

Figure 8: Participation style
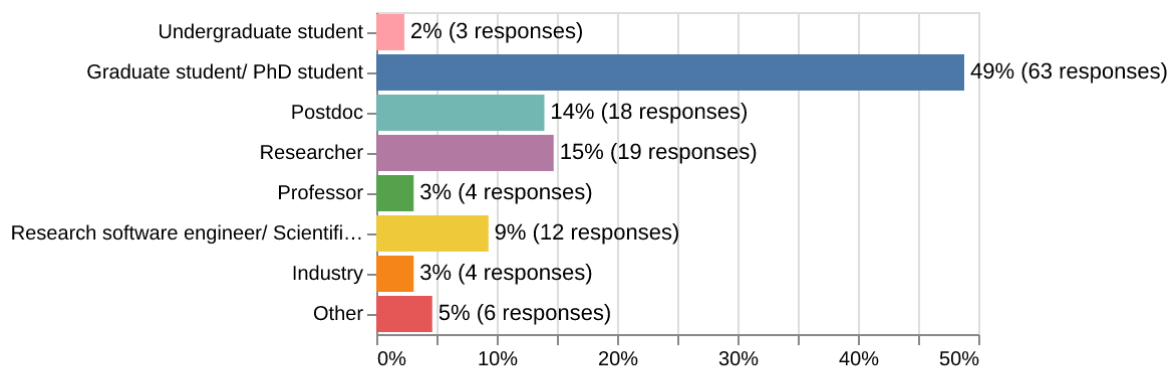
Career stage

Figure 9: Career stage

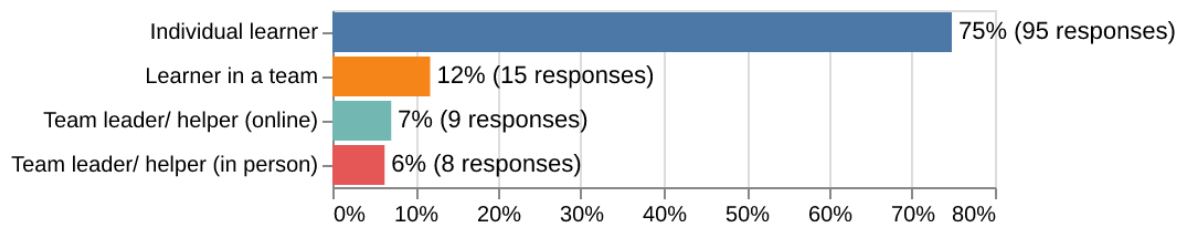| Academic discipline	
Physical Sciences	23% (30 responses)
Earth and Related Environmental Scie...	14% (18 responses)
Biological Sciences	10% (13 responses)
Chemical Sciences	9% (11 responses)
Computer and Information Sciences	9% (11 responses)
Other Engineering and Technologies	5% (6 responses)
Health Sciences	4% (5 responses)
Psychology	3% (4 responses)
Mathematics	3% (4 responses)
Mechanical Engineering	2% (3 responses)
Civil Engineering	2% (3 responses)
Environmental Engineering	2% (3 responses)
Medical Engineering	2% (2 responses)
Medical Biotechnology	2% (2 responses)
Clinical Medicine	2% (2 responses)
Other Medical and Health Sciences	2% (2 responses)
Other	2% (2 responses)
Other Social Sciences	2% (2 responses)
Chemical Engineering	1% (1 response)
Electrical Engineering, Electronic Engi...	1% (1 response)
Economics and Business	1% (1 response)
History and Archaeology	1% (1 response)
Nano-technology	1% (1 response)
Other Natural Sciences	1% (1 response)

Academic discipline

Figure 10: Academic discipline

11

4.3 Lesson material and video recordings

The project has developed and is maintaining and updating a broad curriculum of lessons and manuals.

4.3.1 Standard lessons

These are the lessons which are included in the "standard" CodeRefinery workshop, and what we believe that every computational scientist (even those with very small amounts of computation) should know.

- Version control with **focus on collaboration** and **not only for the command line**:

  – Introduction to version control (day 1-2): **Why we want to track versions and how to go back in time to a working version.** This lesson brings you from zero to using Git and GitHub for own projects.

  – Collaborative distributed version control (day 3): This lesson builds on "Introduction to version control" and we apply branching and learn about pull requests (merge requests), forks, and collaboration using Git and GitHub.

- Reproducible research (day 4): **Preparing code to be usable by you and others in the future.** We focus here on 3 aspects of reproducible programs and computations: documenting dependencies, environments, and computational steps in a reproducible way. We touch on containers.

- Social coding and open software (day 4): **What can you do to get credit for your code and to allow reuse.** We motivate and give an overview over software and data licensing and software citation best practices.

- How to document your research software (day 5): Here we give an overview of the different ways how a code project can be documented: from small projects to larger projects. Markdown and Sphinx are central tools in this lesson.

- Jupyter notebooks (day 5): **A tool to write and share executable notebooks and data visualization.** This lesson gives an overview of what Jupyter notebooks are, when they can be particularly useful, and shows best practices for reusable and reproducible notebooks.

- **Automated testing** (day 6): **Preventing yourself and others from breaking your functioning code.** In this lesson we talk about motivation for testing, about test design, but also about some tools that are typically used for automated testing of software.

- **Modular code development type-along** (day 6): **Making reusing parts of your code easier.** The focus of this lesson is how to partition and organize projects as they grow from one screen-full to larger and how to make code portions reusable across projects and across notebooks.

4.3.2 Shorter formats

- **Reproducible research software development using Python (using ML example)**: 3-days format tailored for people who write Python code for machine learning.
- **Python Progression: Moving Beyond the Basics**: This aims to be a "best of" from **Python for Scientific Computing**, distilled into a two half-day workshop.
- **Reproducible research software development using Python**: 2.5 days format tailored for people who write Python code.
- **CodeRefinery "mini-workshop"**: Three sessions, each 1.5 - 2 hours long. Making an image processing pipeline reproducible, reusable, and extensible. Uses Python in examples.
- **"CodeRefinery in one hour"**: If you want to get an impression of the topics discussed in this workshop, check out the materials of the CodeRefinery in one hour workshop (**slides** and **exercises**).

4.3.3 Video recordings

A number of video recordings of CodeRefinery lessons and workshops and **Research Software Hour** sessions can be found on our **YouTube channel**.

4.3.4 Installation guides

- **Installation instructions** (see your workshop page for which ones are actually needed for you)

4.3.5 Instructor training and manuals

- Train the trainer workshop: Do you teach the use of computers and computational tools? Are you curious about scaling your training and learn about tested and proven practices from CodeRefinery workshops and other trainers? Four self-contained sessions on tools and techniques for computational training offer a great chance to enhance your teaching skills and learn about new tools and practices. What you will learn is also used a lot outside CodeRefinery, whenever good beginner friendly training is needed.

- CodeRefinery manuals, our guides and hints on running CodeRefinery: As the project and community grows, documentation becomes more important and this is our place where we document our guides and methods and solutions.

4.3.6 Special topics

- Tuesday Tools & Techniques for High Performance Computing: this is a series of four self-contained episodes on best practices in high performance computing and a great chance to enhance your computational skills:

  - HPC Resources: RAM, CPUs/GPUs, I/O
  - Day-to-day working on clusters
  - Containers on clusters
  - Parallelization and workflows

- Collaborating and sharing using GitHub without command line: In this lesson we try to introduce concepts around GitHub collaboration without using the command line at all, only using the web interface or GitHub desktop. This lesson has been picked up and developed further by others and is now part of the Carpentries Incubator.

- Data visualization using Python: This material can be a useful and pragmatic starting point for somebody who starts with Python and wants to get up to speed with data visualization. Exercises are mostly using Vega-Altair but also other libraries are mentioned.

- CMake workshop (based on ENCCS version): This material can give a useful starting point into CMake. It contains exercises and was taught as part of a hackathon focused on porting codes to CMake.

- Modular code development: This is an older version of the modular code development lesson based on slides and group work. This can work if the learners prefer discussions and presentation instead of demo and discussion.

- Git branch design: Unfortunately we don't teach this lesson very often but it contains probably useful content about how to organize branches in a project. It also contains a discussion and exercises about rebasing which is compared to merging.

- Mixed Martial Arts: Interfacing Fortran, C, C++, and Python: We haven't taught this lesson in a while so it will need some updates but it can be a useful starting point when developing material for mixed-language development and interfacing programs written in different languages.

4.3.7 Derivative materials and programs

Derivative work based on our material is collected on this webpage.

Other organizations mentioning CodeRefinery lessons:

- https://de-rse.org/learn-and-teach/learn/
- https://de-rse.org/learn-and-teach/teach/
- https://www.esciencecenter.nl/training-materials/ (see also this tracking issue)

4.4 Community management

The CodeRefinery project is more than a set of lessons, repositories, and workshops. It is a community which reached a size where it needs active management. It was important for the project to employ a dedicated community manager in the second half of the project phase.

Below we highlight some of the activities and achievements in the context of community management and engagement:

- We have put more emphasis on improving the on-boarding process for new instructors and helpers: Through better prepared meetings, through co-teaching, train-the-trainer workshops, and with the help of operational manuals.
- We held two in-person team meetings: A kick-off event in Espoo in 2022 and a writing retreat in Tromsø in 2024.

- We have celebrated 8 years of the project with a blog post.
- We hold weekly calls with the project team.
- In order to make it easier for team members and volunteers to contribute to the project and get an overview of available tasks and open issues, we have a task and roadmap page as well as a project board.

We are well connected with other projects and communities with a similar scope:

- OLS
- Digital Research Academy (DRA)
- INTERSECT
- The Carpentries
- UNIVERSE-HPC
- Suresoft
- EVERSE

A number of CodeRefinery members are involved in Nordic research software engineers (RSE) organization and we have helped organizing a couple of events:

- Nordic-RSE conference 2024
- Finnish RSE meetup
- Nordic-RSE unconference 2023
- Nordic-RSE unconference 2022

4.5 Ambassador program

In 2024 we started the CodeRefinery ambassador program as a way to engage and support people that like the project but do not have the resources or background knowledge to support the project as a partner, co-instructor or helper. Since the start of the program in 2024 we held 5 ambassador calls with 5-15 participants each to get to know each other, share updates about the project and discuss open questions and ideas.

To date, 26 people have signed up for the ambassador e-mail list. Ambassadors are from different organizations and backgrounds from all over Europe. We got great feedback and were able to provide support where needed; e.g. shared stickers for on-site workshops, provided a short slide summary of the project, and event posters to print and distribute at the ambassadors organizations.

4.6 Social media, newsletters, and chat

During phase 3 we have added further channels to our outreach activities:

- LinkedIn: https://www.linkedin.com/company/coderefinery-research-software-development/ (372 followers)
- Mastodon: https://fosstodon.org/@coderefinery (316 followers)
- BlueSky: https://bsky.app/profile/coderefinery.org (76 followers)

Summary of number of followers/subscribers to our newsletters and chat:

- Newsletter: 193
- Chat digest: 26
- Ambassadors: 26
- Zulip chat: 499

In February 2025 we discontinued posting on Twitter (820 followers).

4.7 Open House events

Towards the end of phase 3, we have restarted to provide CodeRefinery Open House events, a concept we have experimented with during phase 2 of the project.

The two recent events had the following focus:

- Open House focusing on week 2 of our 2-week workshops.
- Open House focusing on advanced version control ("Git masterclass"; see blog post).

4.8 Steering group and stakeholder engagement

During phase 3 of the project we have held 12 steering group meetings (4 meetings per year).

The meeting minutes are publicly available.

4.9 Governance and sustainability

We worked out a **governance charter** (draft to be approved) for the project to define the roles and responsibilities of different actors involved in the project. The goal is to increase flexibility and lower the barrier for new partners.

4.10 Blog posts

We have published a number of blog posts during phase 3. We believe that blog posts are a good way to communicate with the community and to document our activities. We use them as "organizational memory".

Here is a list of blog posts published during phase 3:

- CodeRefinery Open House - advanced git / git masterclass (2025-01-31): Report from our open house session on advanced git / git masterclass on 14.01.25.
- CodeRefinery - Celebrating eight years (2024-09-19): Continuing eight years of collaboratively teaching good enough research software engineering practices.
- CodeRefinery train the trainer workshop (2024-09-09): Report and insights gathered during our train the trainer workshop in August/September 2024.
- Bring your own classroom to a CodeRefinery workshop (2024-08-19): Join a CodeRefinery workshop with your own classroom and help us scale.
- Results from our 2024 post workshop survey (2024-08-10): In this blog post we summarize the feedback and suggestions collected from participants of workshops from 2022 and 2023.
- Citation Information for Open Source Lessons (2024-07-30): Discussing the current state and potential future applications of the Citation File Format for lessons.
- Summary of business and financing models -planning session Tromsø 2024 (2024-06-17): Discussing possible business and financing models for CodeRefinery.
- Why should your organization support CodeRefinery and how? (2024-06-07): Organizations can support the CodeRefinery project in several ways to make it sustainable and can have a mutually advantageous collaboration.
- In-person meeting summary: Community (2024-04-20): What is the CodeRefinery community to you?.

- In-person meeting summary: Onboarding (2024-04-20): Onboarding new people to the community.
- Drafting a governance structure for 2025 and beyond (2024-04-19): We are working on defining a mission, vision, guiding principles, and a decision body for CodeRefinery in 2025 and beyond. This is an early draft.
- We have completely changed our Git lessons. Hopefully to the better. (2024-04-19): We don't start on the command line anymore. Instead, we start in the web interface of a Git hosting service and offer several tracks to follow the lesson.
- The Modular Code Developement Lesson (2024-04-19)
- CodeRefinery Social Media Strategy (2024-04-19)
- Asynchronous vs. Live Online Teaching: Understanding the Pros and Cons (2024-04-18)
- In-person meeting: install instructions update (2024-04-04): We looked at the installation instructions for CodeRefinery workshops and tried to figure out how to improve them.
- In-person meeting summary: video production (2024-04-04): The video production session provided a hands-on experience in what goes on behind the scenes. It wasn't that hard.
- Help us do the next CodeRefinery workshop (2024 H2) (2024-03-25): Our workshop in 2024 March went well. But we need help if we want to do this again.
- Anyone want a streaming training workshop? (2024-03-17): CodeRefinery's streaming setup is getting fairly professional. Anyone want to learn how to reproduce it?.
- Recap of the first week of the spring 2024 workshop (2024-03-15): Looking back at week one of the CodeRefinery workshop and the big rework of the version control lessons.
- Help us make usable IDE/git install instructions (2024-02-29): We want our workshops to be more usable than before, and that means aligning with best practices.
- Workshop format (2024-02-27): Discussions on Workshop format and how it affects lesson development.
- Summary from a brainstorming meeting about the project future (2024-01-24): Screenshots and summary from our concept board.
- Lessons learned from the September 2023 online workshop (2023-12-05)
- What we plan to improve for the September 2023 workshop (2023-06-25): We encourage the community to give us feedback on these ideas.
- Lessons learned from the March 2023 online workshop (2023-04-12)
- CodeRefinery workshop, 21-23 and 28-30 March 2023 (2023-02-14): Our March workshop

is soon, and you can join alone, as a team, or as a volunteer.

- Learner teams in courses (2022-11-28): Teams can provide high-quality interaction even in large courses.
- Livestreaming courses (2022-11-14): Livestream courses have changed how we teach by allowing unprecedented scaling possibilities.
- Lessons learned from the Sep 2022 online workshop (2022-11-08)
- CodeRefinery Mastodon account (2022-11-08): CodeRefinery has started a Mastodon social media account.
- Video publishing supports more learning styles (2022-11-08): Publishing course videos supports more learning styles and doesn't have to affect privacy.
- Reverse hybrid teaching (2022-11-07): 'Reverse hybrid' is instructors remote, students local - possibly spread among different locations in small groups.
- Co-teaching and scaling up (2022-10-31): Co-teaching replaces (forced?) learner interaction with natural interaction between co-instructors.
- Parallel chat ("HackMD") and scaling teaching (2022-10-24): Parallel chat via a text document enables interactively at scale when online teaching.
- Python for Scientific Computing open for registration and collaborators (2022-10-21)
- CodeRefinery teaching strategies and the future of teaching (2022-10-17)
- Outcomes from online hackathon about measuring the impact of CodeRefinery workshops (2022-05-18)
- Our plans to improve our workshop registration process (2022-05-04)

4.11 Presentations and other contributions to conferences

Zenodo community

We added the Zenodo CodeRefinery community. This way, presentation slides don't disappear, they can be cited, and you can find them all in one place.

Everyone can submit their own presentations or derivative work to the CodeRefinery community on Zenodo. This community is not only for lessons but also presentations and reports.

2024

- "Enabling Open Science Through Research Code" meet-up series, co-hosted by RSSE Africa and RSE Asia, Episode 2: Enabling Reproducibility through Research Code, Nov

14, 2024 (blog post).

- CarpentryConnect in Heidelberg, Germany in November 2024: Poster and lightning talk on "Teaching 'good enough' research software engineering skills" (Samantha Wittke)

- "The CodeRefinery project for training in research software engineering" at The HPC Training Ecosystem in Europe (Online), Oct 22, 2024 (Johan Hellsvik)

- Bring your own classroom to a CodeRefinery workshop at RSECon in Newcastle, UK in September 2024 (Samantha Wittke).

- Participation in Panel discussion on "Towards adoption of FAIR4RS: equipping researchers with FAIR research software skills" at RSECon 2024 (Samantha Wittke)

- "Practical software licensing" at DataverseNO webinar, Jun 10, 2024 (Radovan Bast)

- "CodeRefinery - My pathway into Research Software Engineering" at RSEHPC/ISC24, Jun 10, 2024 (Samantha Wittke)

- "Competence building, the CodeRefinery way" at the NeIC 2024 conference in Tallinn, May 28, 2024 (Radovan Bast)

- Presentation to the Open Science Group, Linköping University, May 7, 2024 (Johan Hellsvik)

- SIGHPC Education Webinar CodeRefinery - collaboratively training the next generation of Research Software Engineers, presentation slides, recording, Apr 25, 2024 (Samantha Wittke, Radovan Bast)

2023

- Research Software Hour flashtalk at Supercomputing 23 BoF on Scientific Software and the People Who Make it Happen: Building Communities of Practice, Nov 14, 2023 (Radovan Bast) - blogpost on the session

- CodeRefinery talk at RSECon23: "Experiences from teaching basic RSE skills with CodeRefinery", Sep 05, 2023 (Samantha Wittke, Radovan Bast)

- CodeRefinery flashtalk in BoF at ISC - Challenges in reverse hybrid teaching, May 23, 2023 (Matias Jääskeläinen)

- CodeRefinery flashtalk and participation in panel discussion Education in HPC: A Lifelong Effort, EuroHPCSummit, Mar 22, 2023 (Johan Hellsvik)

- "CodeRefinery: What We Learned from Teaching Software Engineering Practices to Students and Researchers in Nordics and Beyond") at SIAM Conference on Computational Science and Engineering 2023 (Radovan Bast)

- CodeRefinery short talk "CodeRefinery - a hub for FAIR Software practices" at deRSE

as part of the Teaching and Learning Research Software Engineering workshop, Feb 21, 2023 (Samantha Wittke)

2022

- "About the CodeRefinery project" at the EuSSI training bazaar - Nov 21, 2022 (Matias Jääskeläinen)
- "CodeRefinery - a hub for FAIR Software practices" at DeiC conference 2022 (Radovan Bast)

4.12 Mentions of CodeRefinery in other publications

- I. A. Cosden, K. McHenry and D. S. Katz, "Research Software Engineers: Career Entry Points and Training Gaps," in Computing in Science & Engineering, vol. 24, no. 6, pp. 14-21, Nov.-Dec. 2022, doi: https://doi.org/10.1109/MCSE.2023.3258630 or on arxiv; page 7.
- US Research Software Engineer Association, & IEEE Computer Society. (2023). Research Software Engineers: Creating a Career Path—and a Career. Zenodo. https://doi.org/10.5281/zenodo.10073233; page 19.
- Barker, M., Breitmoser, E., Broadbent, P., Chue Hong, N., Hettrick, S., Lampaki, I., Quinn, A., & Taylor, R. (2024). Software and skills for research computing in the UK. Zenodo. https://doi.org/10.5281/zenodo.10473186; page 15.
- Goth F, Alves R, Braun M et al. Foundational Competencies and Responsibilities of a Research Software Engineer [version 1; peer review: 2 approved]. F1000Research 2024, 13:1429 (https://doi.org/10.12688/f1000research.157778.1).

4.13 GitLab code repository hosting

The CodeRefinery GitLab service began during the project's first funding period (2016-2018) and was initially hosted by CSC (Finland). Although it has limited usage, it is valued by its user base, who appreciate having a GitLab instance where data remains in Europe (specifically Scandinavia) under a non-profit/academic governance.

- Service: https://source.coderefinery.org
- Documentation: https://coderefinery.org/repository/

Usage statistics

- Number of users: 822
- Number of users active in the last year: 267
- Number of users active in the last 100 days: 135
- Number of projects: 1993
- Number of projects active in the last year: 193
- Number of projects active in the last 100 days: 106

We are in the process of migrating the project from being administrated by the CodeRefinery project to being managed by the Danish e-Infrastructure Consortium (DeiC).