

# Why Senior Researchers Should Support Python Workflows

While many researchers learn Python independently, they often lack structured support at the project level.

Supporting Python-based research does not mean you have to write code yourself. It means recognising that **code is part of the research process and creating the conditions for high-quality, reusable, and credible work.**

By actively supporting Python workflows, you can strengthen the quality, speed, and long-term value of your group's research.

**How Your Support Can Make a Difference**

## Reduce Technical Debt

**Technical debt** happens when quick solutions are chosen instead of well-structured code.

At first, the script may work, but over time, problems build up:

- The code becomes hard to understand, even for the original author.
- Minor changes cause unexpected failures because the code is fragile.
- New team members waste time trying to fix or replace old work.

In research, technical debt leads to delays, errors, and lost knowledge.

Good habits early on — like clear naming, documentation, and modular coding — stop technical debt from accumulating and keep projects strong over time.

## Accelerate Onboarding

When a new student or collaborator joins an ongoing project, messy or undocumented code can delay their progress for weeks.

If projects have clean folders, clear instructions, and good coding standards, new team members can:

- Understand workflows faster
- Trust the results
- Start contributing sooner

Good project structure saves everyone time and energy, and allows the team to grow more easily.

## Strengthen Reproducibility and Open Science

Reproducibility is not an extra task at the end — it must be built into the project from the start.

Using simple practices like version control, documented data sources, and scripted workflows makes it much easier to:

- Regenerate tables, plots, and results
- Share analysis steps openly
- Meet journal and funder expectations for open science

Research that is reproducible earns greater trust, wider use, and longer impact.

Helping your team embed reproducibility early also prepares them for future research careers where open practices are increasingly valued.

If you are worried about the security of your project code, consider the Chalmers Git portal. [www.git.chalmers.se](http://www.git.chalmers.se)

## **Ensure Proper Resource Allocation**

Technical work — like data extraction, cleaning, code development, and setting up environments — takes real time.

If these efforts are not recognised and planned for, deadlines slip and workloads become unfair. Good planning protects both project success and staff wellbeing.

Supervisors who take technical needs seriously help by:

- Identifying manual tasks that can be automated (or not automated)
- Ensuring that time consuming manual work is never repeated
- Allowing enough time for coding, documenting and debugging
- Budgeting for research assistants, data engineers, servers, IT Support or software where needed
- Avoiding burnout by sharing workloads wisely

## Raise Research Quality and Visibility

Clean, documented Python projects have a life beyond the original study. They can be reused, cited, extended, and adapted into new collaborations.

Labs/Research groups that support good technical practices tend to:

- Publish faster and more often
- Build stronger reputations
- Attract more funding and talented students
- Deliver work that shapes their fields for years to come

Each well-managed project is a foundation for future research. Good practices today lead to more opportunities tomorrow.



# Why This Matters in a Research Career

Technical skills and reproducibility practices are now vital parts of being a good researcher.

They show seriousness, integrity, and readiness for future roles in academia, industry, or public service. By creating well supported and well documented open-source code you can also attract talented researchers to join the group in the future. Programmers love open-source!

By supporting Python workflows thoughtfully, supervisors do more than just help students complete theses.

They train them in being more efficient, thoughtful and upskill them for the future.

In short:

**You don't need to be a coder — just being aware of what's needed makes you a better.**

