WI-Project: Open-Source Development

Best Practice Session

Learning objectives

Our objectives for today are to

- Discuss current challenges and how they can be addressed
- Facilitate the practical application of Git commands and Python coding from previous sessions

The focus is on **helping teams organize their work effectively**. To this end, we

- Encourage you to share challenges, errors or lessons learned
- Do not introduce new commands, except when you ask for them or when they are useful for your work

Questions and work status

- Warm-up questions
- Why are there errors in my environment?
- How does CoLRev work?
- How can we organize and split tasks?
- How should we select merging strategies?
- Open issues

Quick warm-up questions

Do we have to change the README.md?

- You do not have to change the top-level README.md file, but you have to update README.md files in subdirectories to document your work.

How can I commit if the pre-commit hooks fail?

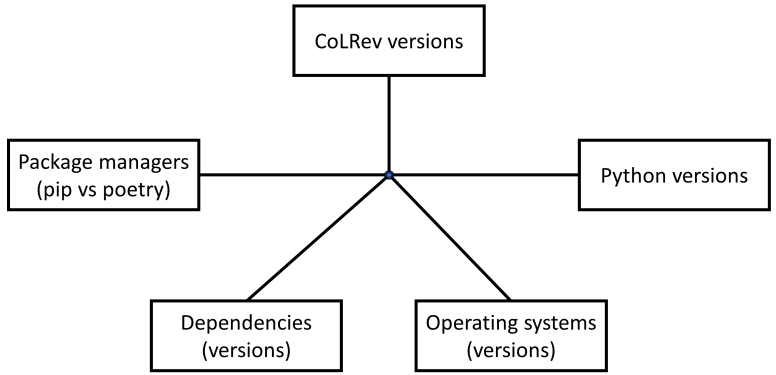
- The pre-commit hooks can be skipped using `git commit -n -m "commit message..."`. The `-n` flag stands for `--no-verify` and skips the pre-commit hooks.
- However, pre-commit hooks should pass to ensure good code quality. Some of the failing tests have been fixed recently.

How can I get access rights to create branches in the fork?

- Ask the team lead to invite you to the fork (see step 2).

Work status: Environment setup

- Errors may be raised only in specific settings, i.e., versions of operating systems, Python, package managers, dependencies, and CoLRev
- What we do to identify and fix errors:
  - Run matrix tests covering 16 different environments
  - Reduce dependencies
  - Fix errors that are reported
- What you can do to avoid errors:
  - Use supported environments, such as GitHub Codespaces or Ubuntu
  - Avoid cutting-edge versions (of operating systems and Python)
  - Report errors



Work status: Environment setup

Are you confident with your setup, including the fork, local development, running your code, executing pre-commit hooks to improve code quality?

Summarize the work status per group:

- What are the challenges we can discuss?
- What was particularly helpful? Are there any insights you can share with the other teams?

Work status: Where to contribute

- Understanding of CoLRev
  - **User workflow**: Start with the video and check `gitk` after each step
  - **Architecture**: Starting point: API chart and reference and modules
  - **Package development and SearchSources**: Starting point: package development resources, CEP 003 - SearchSources
- As a team, do you know where and how to contribute your code, i.e., the modules, classes to use or create?

Work status: How to organize tasks

- How did you distribute tasks in the team?
- Who works on what, which branches did you create, does regular sync work for you?

When tasks are distributed, and you work alone, work in local non-shared branches (e.g., `api_retrieval`):

- Rebase on (parent) feature branch to keep your branch "up-to-date" (`git rebase unpaywall_search`)
- Once the branch is online, use merge commits

Merging into a target branch, i.e., your shared feature branch:

- Squash if you have worked on a single coherent task, which should be combined in a single commit
- Rebase if you would like to preserve a simple linear history
- Merge commit otherwise

Open questions?

We value your feedback and suggestions

We encourage you to share your feedback and suggestions on this slide deck:

 <img src="../assets/iconmonstr-pencil-lined.svg" alt="Edit" width="32" height="32"> Suggest specific changes by directly modifying the content
 <img src="../assets/iconmonstr-info-12.svg" alt="New Issue" width="32" height="32"> Provide feedback by submitting an issue
Your feedback plays a crucial role in helping us align with our core goals of **impact in research, teaching, and practice**. By contributing your suggestions, you help us further our commitment to **rigor**, **openness** and **participation**. Together, we can continuously enhance our work by contributing to **continuous learning** and collaboration across our community.

Visit this page to learn more about our goals: 🚀 🛠️ ♻️ 🙏 👨‍🎓 .