



# Working in software teams

---

Data Sciences Institute  
University of Toronto

**Simeon Wong**

# Asking questions

- Zoom chat during class
  - Feel free to post and answer questions at any time
  - I will pause for questions occasionally, and review questions from the chat
- Pre- / Post-class office hours with Tong
- Email
  - [simeonm.wong@mail.utoronto.ca](mailto:simeonm.wong@mail.utoronto.ca)
  - [tong.su@mail.utoronto.ca](mailto:tong.su@mail.utoronto.ca)

# Daily scrum

Write in the chat:

1. How you'll use one thing you've learned so far
2. One thing you're struggling with / wish the course addressed

How to write robust software in a team that we, our colleagues, and the public can trust and use with confidence.

# Previously at the DSI...

- Testing code
  - Write code with testing in mind
  - `pytest` and GitHub Actions for automated testing
- Python packages
  - Python code is usually distributed as packages on PyPI
  - Use `setuptools` to describe dependencies
  - Uploading and installing packages from GitHub using `pip`

# Course overview

1. Configuration files & Environment variables
2. Using and writing Application Programming Interfaces (APIs)
3. Handling errors
4. Testing software
5. Building Python packages
- 6. Working in software teams using GitHub features**

# Today's learning objectives

- I can track and manage software projects using GitHub
- I can ensure code is reviewed and tested using Pull Requests
- I know the principles of software project governance

## Review

# Why Git?

**Scenario:** Alex is a developer for a business-intelligence dashboard project.

- Alex's current big project is upgrading a financials chart on the dashboard home page
- Halfway through, Alex's colleague Jordan publishes a bug-fix on one of the analysis functions Sam uses for his chart
- Alex needs to integrate the updated analysis function into his work before shipping his upgraded chart to ensure it is correct.



## Review

# Git: Track changes for software

- Addresses unique challenges with working on software collaboratively
- Tracks changes to code over time
  - Backup of your work
  - Revert to older snapshots / point-in-time
- Merge changes from different features developed simultaneously

## Review

# Git commands

### Setup

- git init
- git clone

### Getting info

- git status
- git log

### Making edits

- git add
- git commit

### Incorporating changes

- git pull
- git merge

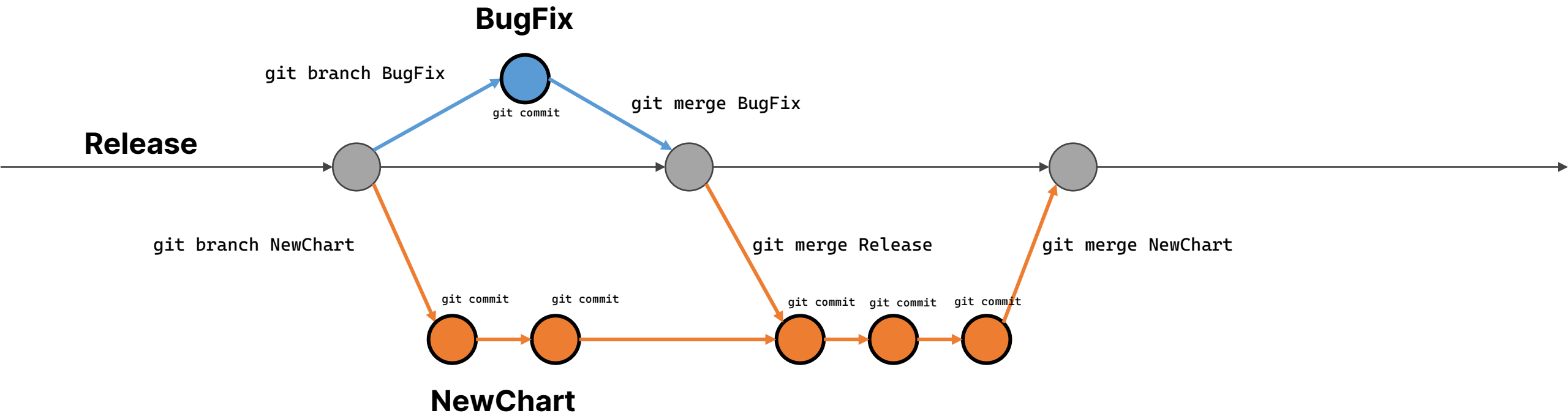
### Navigating through branches and commits

- git checkout
- git switch

## Review

# Alex's new feature

1. Alex starts working on NewChart
2. Jordan starts working on BugFix
3. Jordan ships BugFix
4. Alex integrates BugFix
5. Alex keeps working on NewChart
6. Alex ships NewChart



# Why software project management?

**Scenario:** Alex is a developer for a business-intelligence dashboard and is upgrading a financials chart on the dashboard home page

- Alex wants to gather ideas and feedback:
  - How the chart should look
  - How to make the analysis engine the most efficient
- Alex needs a place to track major discussion points on this feature

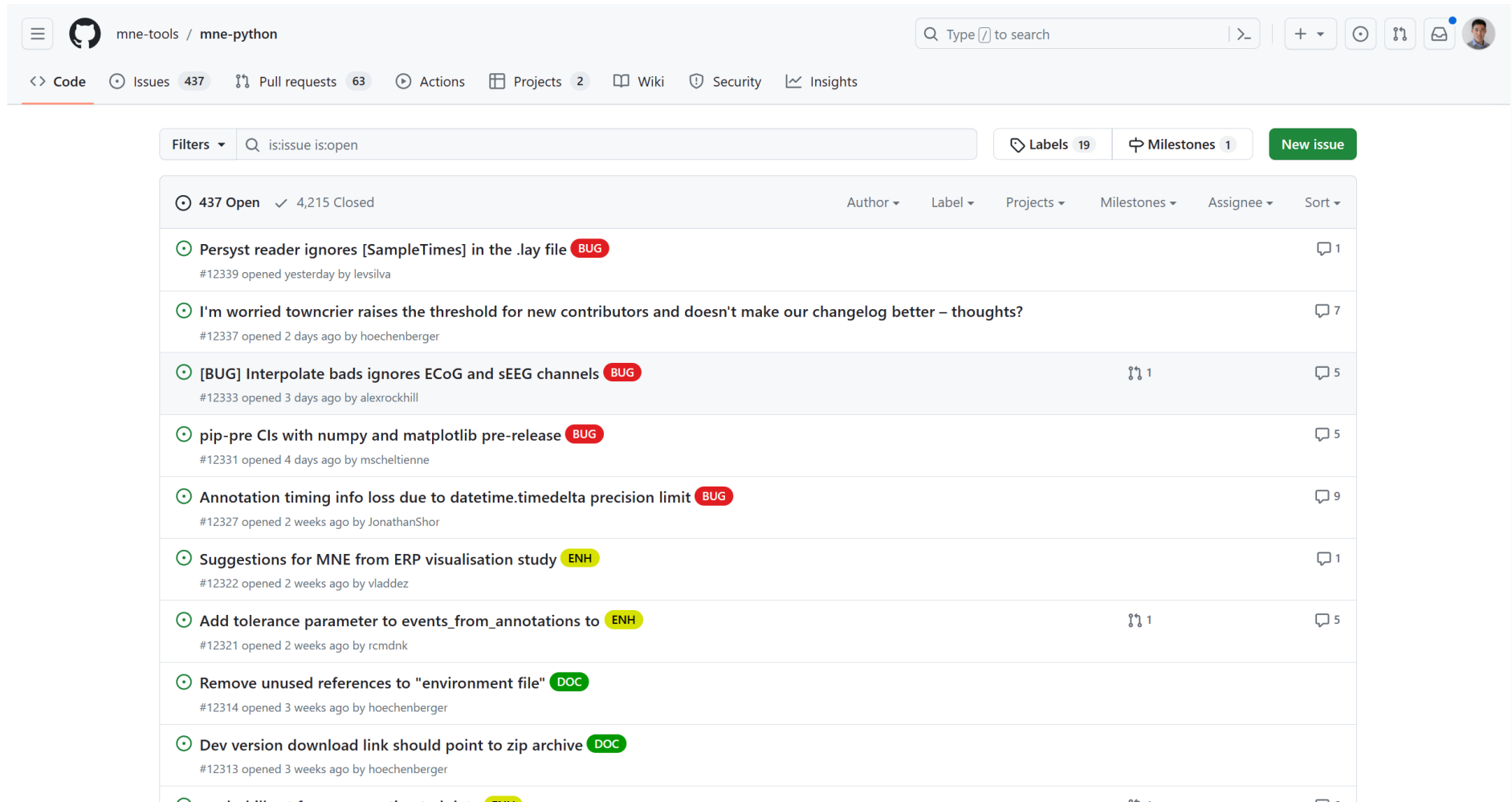
## Working in software teams

# GitHub

- Centralized hosting for Git repositories
- Project management features
  - Issues
    - Description of proposed or requested changes: update, features, bug-fixes, etc..
    - Can associated one or more pull-requests (proposed code changes)
    - Discussion thread
  - Projects
    - Linked to GitHub Issues
    - View Issues in tabular / Gantt chart view

## Working in software teams

# GitHub Issues tracks proposed changes



mne-tools / mne-python

Search: Type / to search

Code Issues 437 Pull requests 63 Actions Projects 2 Wiki Security Insights

Filters is:issue is:open Labels 19 Milestones 1 New issue

437 Open 4,215 Closed

Author Label Projects Milestones Assignee Sort

- Persyst reader ignores [SampleTimes] in the .lay file **BUG** 1  
#12339 opened yesterday by levsilva
- I'm worried towncrier raises the threshold for new contributors and doesn't make our changelog better – thoughts? 7  
#12337 opened 2 days ago by hoechenberger
- [BUG] Interpolate bads ignores ECoG and sEEG channels **BUG** 1 5  
#12333 opened 3 days ago by alexrockhill
- pip-pre CIs with numpy and matplotlib pre-release **BUG** 5  
#12331 opened 4 days ago by mscheltienne
- Annotation timing info loss due to datetime.timedelta precision limit **BUG** 9  
#12327 opened 2 weeks ago by JonathanShor
- Suggestions for MNE from ERP visualisation study **ENH** 1  
#12322 opened 2 weeks ago by vladdiez
- Add tolerance parameter to events\_from\_annotations to **ENH** 1 5  
#12321 opened 2 weeks ago by rcmdnk
- Remove unused references to "environment file" **DOC**  
#12314 opened 3 weeks ago by hoechenberger
- Dev version download link should point to zip archive **DOC**  
#12313 opened 3 weeks ago by hoechenberger

# Working in software teams

# GitHub Issues allows for discussion

mne-tools / mne-python

Issues 437 Pull requests 63 Actions Projects 2 Wiki Security Insights

## ENH: Custom referencing #12283

qian-chu opened this issue last month · 8 comments

qian-chu commented last month · edited

**Describe the new feature or enhancement**

In some cases, especially intracranial EEG, re-referencing of channels should be local (e.g. within the same shaft/grid/strip) rather than global (i.e., ref all channels to one or an average of channels). This thus requires some flexible re-ref schemes, but the existing functions `set_eeg_reference` and `set_bipolar_reference` are not super convenient for such purposes. On the basis of the proposed function, multiple re-ref schemes can be added (for example see <https://doi.org/10.1016/j.neuroimage.2018.08.020>)

**Describe your proposed implementation**

A new preprocessing function `mne.set_custom_reference(inst, ref_dict, copy=True, on_bad='warn', verbose=None)` Where `ref_dict` contains a dictionary with keys being the source channel names and values being ref channel names. For example: `ref_dict = {'A1': 'A1', 'A2': 'A1', 'A3': 'A1', 'A4': 'A1'}` performs monopolar electrode shaft reference (ESR); `ref_dict = {'A1': 'A2', 'A2': ['A1', 'A3'], 'A3': ['A2', 'A4'], 'A4': 'A3'}` performs Laplacian reference. I'm happy to open a pull request draft if someone can confirm the value of the proposed function.

**Describe possible alternatives**

Alternatively we can expand `mne.set_eeg_reference` instead of writing a new function.

**Additional context**

No response

qian-chu added the **ENH** label last month

welcome bot commented last month

Hello! 🎉 Thanks for opening your first issue here! ❤️ We will try to get back to you soon. 🙌

drammock commented last month

Open ENH: Custom referencing #12283  
qian-chu opened this issue last month · 8 comments

qian-chu commented last month

My colleague @AlexLepauvre is also interested in this topic. Alex, from your perspective (especially having seen different iEEG data from multiple centers) what might be most useful to the community?

mscheltienne commented 3 weeks ago

I'm a bit in favor of expanding the existing API instead of creating a new one. `mne.set_eeg_reference(ref_channels=dict(...))` seems like an elegant solution with `ref_channels : str | list[str] | dict[str, str | list[str]]`.

drammock commented 3 weeks ago

thinking further about this, I do really like the `ref_channels=dict(...)` approach: it's flexible and seems (to me at least) easy to understand, and probably not too hard to implement.

However, if we implement this, we probably can't prevent users from applying it to regular scalp EEG data. So one thing we should also think about though is our mechanisms for keeping track of whether EEG reference has been applied; how will we do that for this referencing approach? what will we do during `apply_inverse` when it checks for (and doesn't find) the desired situation of "EEG average reference applied"? (we do have mechanisms for overriding that particular safeguard --- and maybe that's already enough --- but there may be some documentation / user education to be done around the possible downsides of this kind of referencing)

AlexLepauvre commented 3 weeks ago

Hello everyone,


As @qian-chu mentioned, this is something for which we made a custom implementation in my project using data from various sites, using laplace referencing (`ch2 = ch2 - mean(ch1, ch3)`), so I would be happy to help out for the PR once we have figured out the details. In our case, we created a referencing dictionary for each single subject semi-automatically, which was then used to re-reference the channels.

In addition, we also took into account bad channels information. Any channel marked as bad was removed from the referencing scheme. If for a given channel, both references were marked as bad, the channel was also discarded, and if only one of the two was marked as bad, then bipolar referencing was applied.

One nice additional feature (especially for ECoG channels) would be electrodes relocation. In case of bipolar referencing, it is best to relocate the channel as the midpoint between the two references. In the case of surface electrode, that might result in a point that is actually not on the cortical surface anymore. The relocated electrodes should be projected to the surface. This would be interesting to consider as well.


# Working in software teams


## GitHub Issues + Pull Requests

 Open

**[BUG] Interpolate bads ignores ECoG and sEEG channels #12333**  
alexrockhill opened this issue 3 days ago · 5 comments · May be fixed by [#12336](#)

intraoperative. I think in that case, tossing the whole epoch or the whole channel are not ideal so an interpolation seems like the best choice.







**mscheltienne** commented 2 days ago Member ...

I think for ecog the existing spherical interpolation is reasonable. Perhaps seeg should interpolate per lead

+1 for both, I never worked with intracranial data, do we store information about the grid and lead geometry?


Raising a `NotImplementedError` could be detrimental in an edge-case with a `Raw` containing multiple channel types that can be interpolated. For instance, combined stereo-EEG with scalp-EEG or with MEG recordings. I would however improve the `RuntimeWarning` to mention the channel types and bad channels that were *not* interpolated.






**alexrockhill** commented 2 days ago Contributor Author ...

Maybe I'll just add a spline interpolation on an sEEG shaft at the same time, shouldn't be too hard.



 alexrockhill linked a pull request [2 days ago](#) that will close this issue

**[ENH] Add support for ieeg interpolation #12336** Open



Add a comment

Write Preview

H B I L < > Link List Table Quote Spoiler



# GitHub Issues: Assignees

## Add multi-class decoding #711

New issue

Open hoechenberger opened this issue on Jan 21, 2023 · 0 comments



hoechenberger commented on Jan 21, 2023 · edited

Member

Just some thoughts and ideas, again mostly as a memo to myself:

- new setting: `decoding_multiclass_conditions`
- full-epochs decoding:
  - visualize a confusion matrix
- full-epochs decoding based on CSP:
  - one confusion matrix per time-freq bin?
- time-by-time decoding:
  - performance traces, one per condition, across time



hoechenberger added `enhancement` `mvpa` labels on Jan 21, 2023



hoechenberger self-assigned this on Feb 10, 2023

Assignees



hoechenberger

Labels

`enhancement`

`mvpa`

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

Notifications

Customize



Subscribe



UNIVERSITY OF  
TORONTO

# Working in software teams

# GitHub Projects

mne-tools / Projects / 2022 MNE-Python Code Sprint

2022 MNE-Python Code Sprint

View 1

Filter by keyword or by field

**No Status 5**

- mne-bids-pipeline #574  
ENH: Add movement compensation and related functions
- mne-python #1280  
ENH: combined gradiometers (plotting + stats)
- mne-python #9022  
viz.plot\_compare\_evoked() calculates RMS for all channel types, but claims to be calculating GFP
- mne-python #11058  
Noisy point-spread functions
- mne-python #10888  
support "sensitivity" and "timebase" style scaling in raw data browsers

**Todo 1**

- mne-bids-pipeline #581  
ENH: Improve support for infant MRI templates

**In Progress 1**

- mne-python #11064  
ENH: Support TD data

**Done 24**

- mne-python #10184  
add spectrum class
- mne-python #10946  
Montage in MRI space conversion saving to epoch.fif
- openmeeg #562  
Create codeql-analysis.yml
- openmeeg #495  
MAINT: Replace release url with GitHub Actions
- mne-bids-pipeline #580  
Making MNE-BIDS-Pipeline a package
- mne-python #10987  
apply\_inverse\_raw() calls \_check\_channels even though no EEG channels
- mne-python #10988  
All EEG and EOG channels have reading HCP data using read\_raw\_bids
- mne-python #11011  
ENH: Add OpenMEEG for 3-lead MEG
- mne-python #11057

microsoft / Projects / Fluent UI React Component Roadmap

Fluent UI React Component Roadmap

FY24 Roadmap All Up Freezer

-milestone:Freezer

35 Discard

Jun 2023 Jul 2023 Aug 2023 Sep 2023 Oct 2023 Nov 2023 Dec 2023 Jan 2024 Feb 2024

FY23Q4 FY24Q1 FY24Q2 FY24Q3

12 26 10 24 7 21 4 18 2 16 30 13 27 11 25 8 22 5 19 4 18 1 15 29 13 27 10

8 Toast #26638

9 Tag #26001

10 MessageBar #22579

11 ScrollView #27384

12 Segment #27389

13 SimpleListCompat #29920

14 SearchBox #26648

15 BasicList #27382

16 TeachingBubble #26639

17 Coachmark Component #28389

18 Rating #26535

19 Nav #26649

20 SwatchPicker #28606

21 TimePickerCompat #26642

22 [Feature]: Make v9 Dialog Draggable like v8 #28359

23 PeoplePicker #26652

24 [Feature] Advanced Dropdowns - Filter, Clearable #29399

25 Animation and Motion #29400

26 CalendarCompat #26646

27 Carousel #26647

28 ColorPickerCompat #26643

Segment #27389

SearchBox #26648

TeachingBubble #26639

Coachmark Component #28389

Rating #26535

Nav #26649

SwatchPicker #28606

TimePickerCompat #26642

[Feature]: Make v9 Dialog Draggable like v8 #28359

PeoplePicker #26652

[Feature] Advanced Dropdowns - Filter, Clearable #29399

Animation and Motion #29400

CalendarCompat #26646

Carousel #26647

ColorPickerCompat #26643

# Why software project management?

**Scenario:** Alex is a developer for a business-intelligence dashboard project.

- Alex's finishes his big project: upgrading a financials chart
- Avery, the engineering manager wants to make sure:
  - Alex's feature has been peer-reviewed
  - The new charting code integrates well into the dashboard
- Alex wants to make sure his colleagues can understand how to use his code

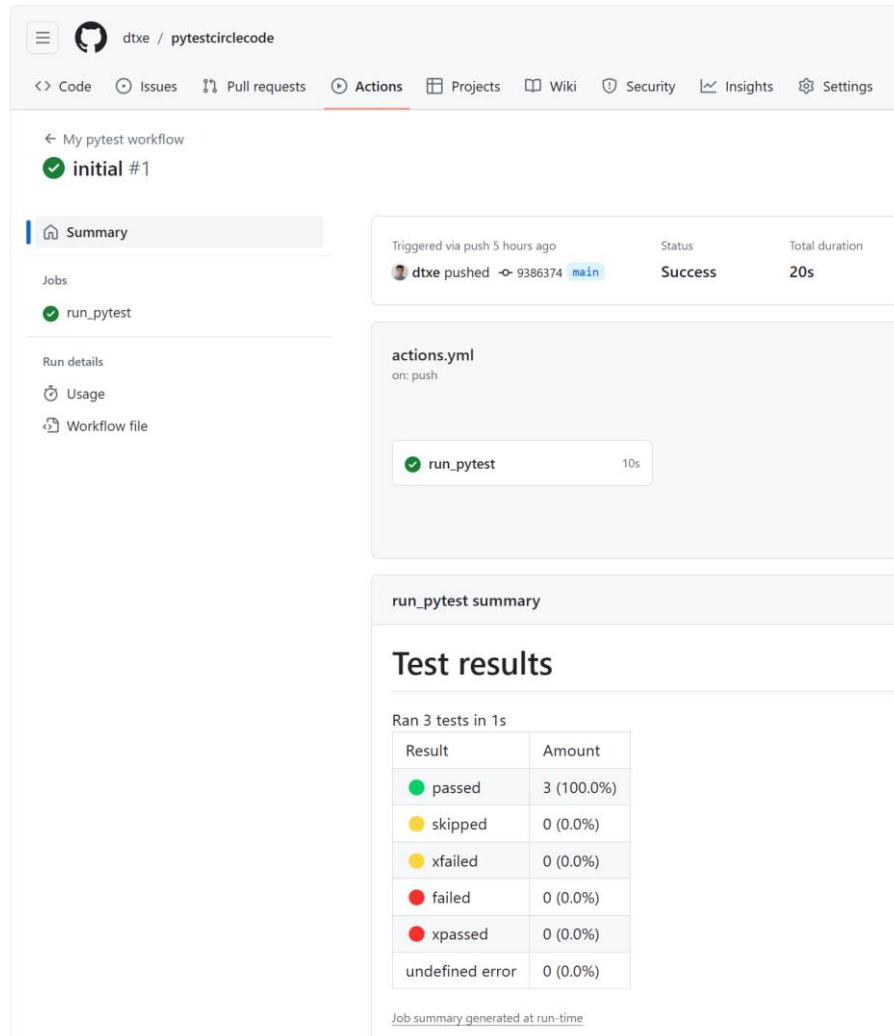
## Working in software teams

# GitHub

- Continuous Integration
  - Alex's code is automatically tested with `pytest` on GitHub actions
- Code review and approvals
  - Alex can request a colleague to peer review his code and make comments overall, or even line-by-line
- Granular read/write access controls
  - Pull requests on the main repository can have requirements
  - e.g. must be approved by one or more colleagues, must pass all unit tests

## Working in software teams

# Continuous integration code testing



The screenshot shows the GitHub Actions interface for a workflow named 'My pytest workflow'. The workflow is in a 'Success' state, triggered by a push to the 'main' branch. The 'run\_pytest' job completed successfully in 10 seconds. The 'Test results' section shows that 3 tests passed (100.0%) and all other categories (skipped, xfailed, failed, xpassed, undefined error) have 0 results (0.0%).

dtxe / pytestcirclecode

<> Code Issues Pull requests **Actions** Projects Wiki Security Insights Settings

← My pytest workflow

✓ initial #1

Summary

Jobs

✓ run\_pytest

Run details

Usage

Workflow file

Triggered via push 5 hours ago

dtxe pushed -> 9386374 main

Status: Success

Total duration: 20s

actions.yml

on: push

run\_pytest 10s

run\_pytest summary

### Test results

Ran 3 tests in 1s

Result	Amount
passed	3 (100.0%)
skipped	0 (0.0%)
xfailed	0 (0.0%)
failed	0 (0.0%)
xpassed	0 (0.0%)
undefined error	0 (0.0%)

Job summary generated at run-time

# Working in software teams

## Assigned reviewers and approvers

ENH: Add Forward.save and hdf5 support #12036 <> Code

**Merged** larsoner merged 2 commits into `mne-tools:main` from `larsoner:fwd` on Oct 1, 2023

Conversation 0 Commits 2 Checks 22 Files changed 6 +69 -8

**larsoner** commented on Sep 30, 2023 Member

Closes #12022  
Closes #12031

No need for latest.inc update for #12031 I think because it's more or less an improvement to the error that gets raised

@mscheltienne feel free to mark for merge-when-green if you're happy. Feel free to test it with your gigantic forward solution but in principle if it fails it's some HDF5 bug.

1

ENH: Add Forward.save and hdf5 support 1d3a7c6

**larsoner** requested review from **drammock** and **agramfort** as code owners 4 months ago

FIX: Link d48ee8f

**agramfort** approved these changes on Oct 1, 2023 View reviewed changes

**larsoner** merged commit `bd4d1d6` into `mne-tools:main` on Oct 1, 2023  
26 checks passed View details Revert

**larsoner** deleted the `fwd` branch 4 months ago

**larsoner** added the **ENH** label on Oct 2, 2023

Add a comment

**Reviewers**

- agramfort** ✓
- drammock** 1

**Assignees**

No one assigned

**Labels**

- ENH**

**Projects**

None yet

**Milestone**

No milestone

**Development**

Successfully merging this pull request may close these issues.

- ☒ `mne.io.read_info()` of CTF.ds returns IsADirectory...
- ☒ Add `mne.Forward.save` and enable saving large F...

**Notifications** Customize

☒ **Subscribe**

You're not receiving notifications from this thread.

2 participants

## Working in software teams

# Branch protection rules

- Repository settings > Branches > Branch protection rules

The screenshot shows the GitHub repository settings for 'dtxe / pytestcirclecode'. The 'Settings' tab is selected in the top navigation bar. On the left sidebar, the 'General' section is expanded, and 'Branches' is selected under 'Code and automation'. The main content area is titled 'Branch protection rules' and displays a message: 'You haven't protected any of your branches'. Below this message is a button labeled 'Add branch protection rule'. The message text states: 'Define a protected branch rule to disable force pushing, prevent branches from being deleted, and optionally require status checks before merging. [Learn more about protected branches](#)'.

# Using GitHub features wisely

- GitHub features are part of a comprehensive code quality strategy
- **Recall from code testing:**  
Balance between impact/risk and engineering effort
  - Don't overburden your contributors and teammates with hoops to jump through if they don't contribute meaningfully to code quality



# Other software project tools

## Code repositories

- Atlassian Bitbucket
- GitLab
- AWS CodeCommit
- Azure Repos
- Google Cloud Source
- Self-hosted over SSH / WebDAV

## Project tools

- Atlassian Jira
- GitLab
- Redmine
- ClickUp
- Monday.com
- Asana

---

# What questions do we have?

# What makes up a project?

- A **dataset** being used by several research projects?
  - raw data, programs used to tidy the data, tidied data, text files describing license and provenance
- A **set of annual reports** written for an NGO?
  - jupyter notebooks, copies of html and pdf versions of the report, a text file containing links to datasets used in the report (which can't be store don Github since they contain personal identifying information)
- A **software library** providing an interactive glossary of data science terms in both Python and R?
  - files needed to create a package, a Markdown full of terms and definitions, a Makefile with targets to check cross references, etc

# What makes up a project?

- What people have meetings about
- **Governance:** If the same group needs to get together on a regular basis to talk about something, that *something* probably deserves its own repository
- **Continuity:** And if the list of people changes slowly over time but the meetings continue, that's an even stronger sign

# Copyright licenses

- A license dictates how project materials can be used and redistributed
  - Can affect the willingness of contributors
  - The choice of license is crucial to the project's long-term sustainability
- Consider contributors' contractual obligations
  - e.g. students and faculty may have a copyright on the research work they produce, but university staff members may not.
- Including an explicit license avoids legal messiness and should be chosen early on
  - **Generally, licenses do not apply retroactively**

# Copyright licenses

- Do we want to license the work at all?
- Is the content we are licensing source code?
- Do we require people distributing derivative works to also distribute their code?
- Do we want to address patent rights?
- Is our license compatible with the licenses of the software we depend on?
- Do our institutions have any policies that may overrule our choices?
- Are there any copyright experts within our institution who can assist us?

# Project governance

# Copyright licenses

Great summary here:

<https://choosealicense.com/licenses/>

If you're looking for a reference table of every license on choosealicense.com, see the [appendix](#).

## GNU AGPLv3

Permissions of this strongest copyleft license are conditioned on making available complete source code of licensed works and modifications, which include larger works using a licensed work, under the same license. Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. When a modified version is used to provide a service over a network, the complete source code of the modified version must be made available.

### Permissions

- Commercial use
- Distribution
- Modification
- Patent use
- Private use

### Conditions

- Disclose source
- License and copyright notice
- Network use is distribution
- Same license
- State changes

### Limitations

- Liability
- Warranty

[View full GNU Affero General Public License v3.0 »](#)

## GNU GPLv3

### Permissions

- Commercial use

### Conditions

- Disclose source

### Limitations

- Liability

# The Code of Conduct

1. Promotes fairness within a group
2. Ensures all members that this project takes inclusion seriously
3. Ensures that everyone knows what the rules are
4. Prevents anyone who misbehaves from pretending that they didn't know what the did was unacceptable

Often listed in a file named `CONDUCT.md` in project directory.

**Example:** <https://www.contributor-covenant.org/>



## Project governance

# Decision making

- Every team has a power structure: formal (accountable) or informal (unaccountable).
- Importance of explicit governance in groups larger than six people.
- **Objective:** Establish who makes decisions and how to reach consensus.

## Project governance

# Decision making: Martha's rules

- Anyone who wants to sponsor a proposal must file one at least 24 hours in advance. It must include:
  - a one-line summary
  - the full text of the proposal
  - any required background information
  - pros and cons
  - possible alternatives
- A quorum is established in a meeting if half or more of voting members are present
- Once a person has sponsored a proposal, they are responsible. The group may not discuss it unless the sponsor is present

## Project governance

# Decision making: Martha's rules

- After the sponsor presents the proposal, cast a sense vote:
  - Who likes the proposal?
  - Who can live with it?
  - Who is uncomfortable with it?
- If everyone likes or can live with it, it passes with no further discussion.
- If most of the group is uncomfortable, it is sent back to the sponsor for further work. (The sponsor can decide to drop it)
- If some members are uncomfortable, a time is set to discuss, moderated by the meeting moderator.
  - After 10 minutes or so, the moderator calls a yes or no vote.
  - If the majority is yes, it passes.
  - Otherwise, it is returned to sponsor for further work.

## Project governance

# Decision making: meeting tips

1. Decide if there actually needs to be a meeting.
2. Write an agenda.
3. Include timings in the agenda.
4. Prioritize.
5. Make one person responsible for keeping things moving.
6. Require politeness.
7. No interruptions.
8. No distractions.
9. Take minutes.
10. End early.

## Project governance

# Document everything!

Make sure your project description, licenses, code of conduct, decision making structures, meeting details are **discoverable** for newcomers!

---

# What questions do we have?

How to write robust software in a team that we, our colleagues, and the public can trust and use with confidence.

# Building software summary

1. Track your changes and work collaboratively in Git
2. Use configuration files
3. Handle and raise errors with meaningful messages
4. Document, test, and package your work for others to use

**Go forth and build great software! 🦾**