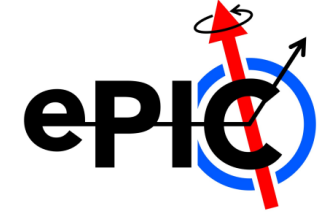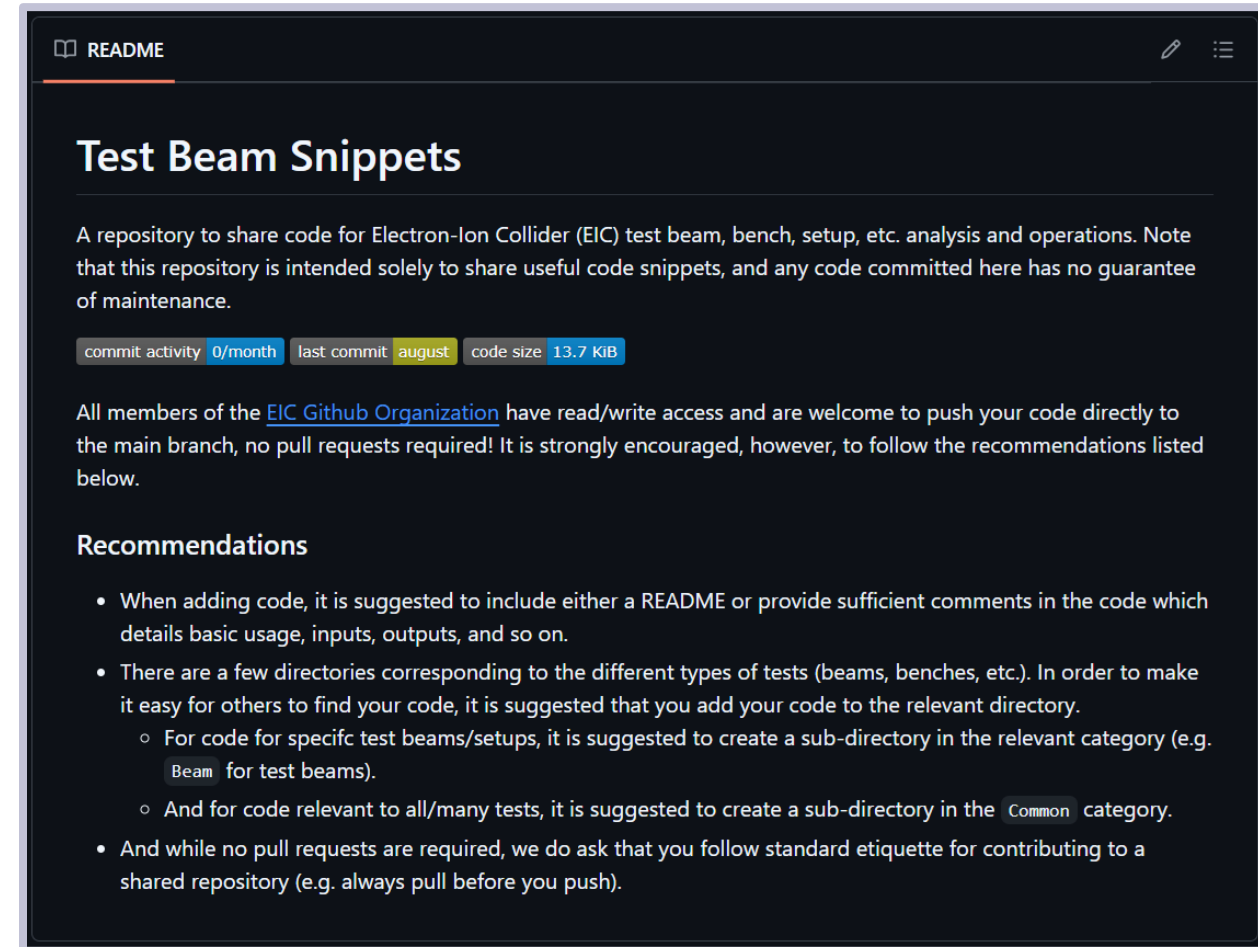# Test Beam Software:

Integration, Reproducibility (and Snippets)

Derek Anderson (JLab)
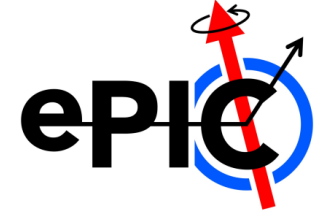
ePIC Weekly S&C Meeting

December 3rd, 2025

Context | TB Software Integration

o Part of discussion during <u>10.29.2025 Weekly S&C Meeting</u> focused on TB software integration
  - i.e. policies + infrastructure needed to make sure that
    a) Users can integrate TB code
    b) TB results are reproducible
  - As initial step towards this, introduced <u>Test Beam Snippets</u> repository

o Reasonable concerns were raised about proliferation of <u>snippets</u>-style repos
  - Ensuing discussion(s) are important!
    › Touch on reproducibility, findability, etc. broadly
  ☞ **Today:** providing status update on them

Test Beam Snippets

A repository to share code for Electron-Ion Collider (EIC) test beam, bench, setup, etc. analysis and operations. Note that this repository is intended solely to share useful code snippets, and any code committed here has no guarantee of maintenance.

`commit activity` `0/month` `last commit` `august` `code size` `13.7 KiB`

All members of the <u>EIC Github Organization</u> have read/write access and are welcome to push your code directly to the main branch, no pull requests required! It is strongly encouraged, however, to follow the recommendations listed below.

Recommendations

- When adding code, it is suggested to include either a README or provide sufficient comments in the code which details basic usage, inputs, outputs, and so on.
- There are a few directories corresponding to the different types of tests (beams, benches, etc.). In order to make it easy for others to find your code, it is suggested that you add your code to the relevant directory.
  - For code for specifc test beams/setups, it is suggested to create a sub-directory in the relevant category (e.g. `Beam` for test beams).
  - And for code relevant to all/many tests, it is suggested to create a sub-directory in the `Common` category.
- And while no pull requests are required, we do ask that you follow standard etiquette for contributing to a shared repository (e.g. always pull before you push).

Reco WG Discussion | Summary

- ○ **Action item:** follow-up discussion in <u>11.10.2025 Reco WG Meeting</u>
  - – Goal was to revisit snippets and evaluate current usage against intent
  - – Clarifying this helps to clarify how to integrate TB code

- ○ Discussion didn't converge, but was productive!
  - – Key points are:
    - › **There are things that snippets is doing well now (see right)**
    - › But 2 items need clarifying (next slide)
      - a) What is a snippet?
      - b) What role does repo play?

Things snippets does well now:

- – Has traction and is actively being used by collaborators now
  - › Low barrier to entry (no reviews) makes it easy for users to contribute
  - › Encourages *user-generated* content and engagement
- – It provides a *direct, single-point-of-entry* for library of snippets
  - › Repo can be cloned to any machine and copy-pasted from
  - › Provides a *single place* to point users for examples
- – Repo organization makes it is easy to search and find relevant snippets

Reco WG Discussion | Summary

o **Action item:** follow-up discussion in <ins>11.10.2025 Reco WG Meeting</ins>
- Goal was to revisit snippets and evaluate current usage against intent
- Clarifying this helps to clarify how to integrate TB code

o Discussion didn't converge, but was productive!
- Key points are:
  › There are things that snippets is doing well now (previous slide)
  › **But 2 items need clarifying (see right)**
     a) What is a snippet?
     b) What role does repo play?

What is a snippet? What should populate the repo?

- Nominally a small chunk of code to illustrate something
- But there is currently *also* analysis code in the repo...

What role does this repo play in our ecosystem?

- Two possible roles: *documentation* and *analysis preservation*
- If *analysis preservation*, then
  › Benchmarks or similar better since testing, maintenance guaranteed
  › Requires *significant* labor though...
- If *documentation,* then
  › Some overlap in functionality with GitHub features
     » **Template repositories:** provide skeletons users can fill in
     » **Topics:** organize repos

Reco WG Discussion | Next Steps
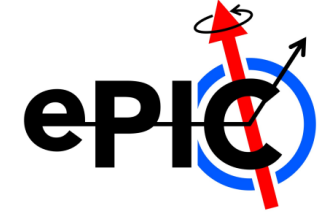
For discussion: some possible next steps

a) **Integrate Preet & Bill's Beam Envelope Script into eic-shell**
- ☞ (With their permission!)

b) **Merge TB snippets into snippets**
- Mitigates proliferation of snippety repos
- But still provides channel for TB code exchange

c) **Keep snippets as-is with caveats in steps (d) and (e)**
- We should definitely start exploring template repos + topics!

d) **Clarify what goes where**
- **In epic/EICrecon/etc:** TB geometries, TB algorithms, etc.
- **In individual repos:** *larger scale* projects, e.g. epic-lfhcal-tbana
- **In snippets:** smaller scale code *to share*

e) **Activate curator role for snippets**
- Charged with curating snippets, coordinating issue resolution, etc.
- Can start as individual, but should evolve to a team taking shifts
- ☞ **Agents could be huge help with role**
  - › e.g. summarizing directory contents
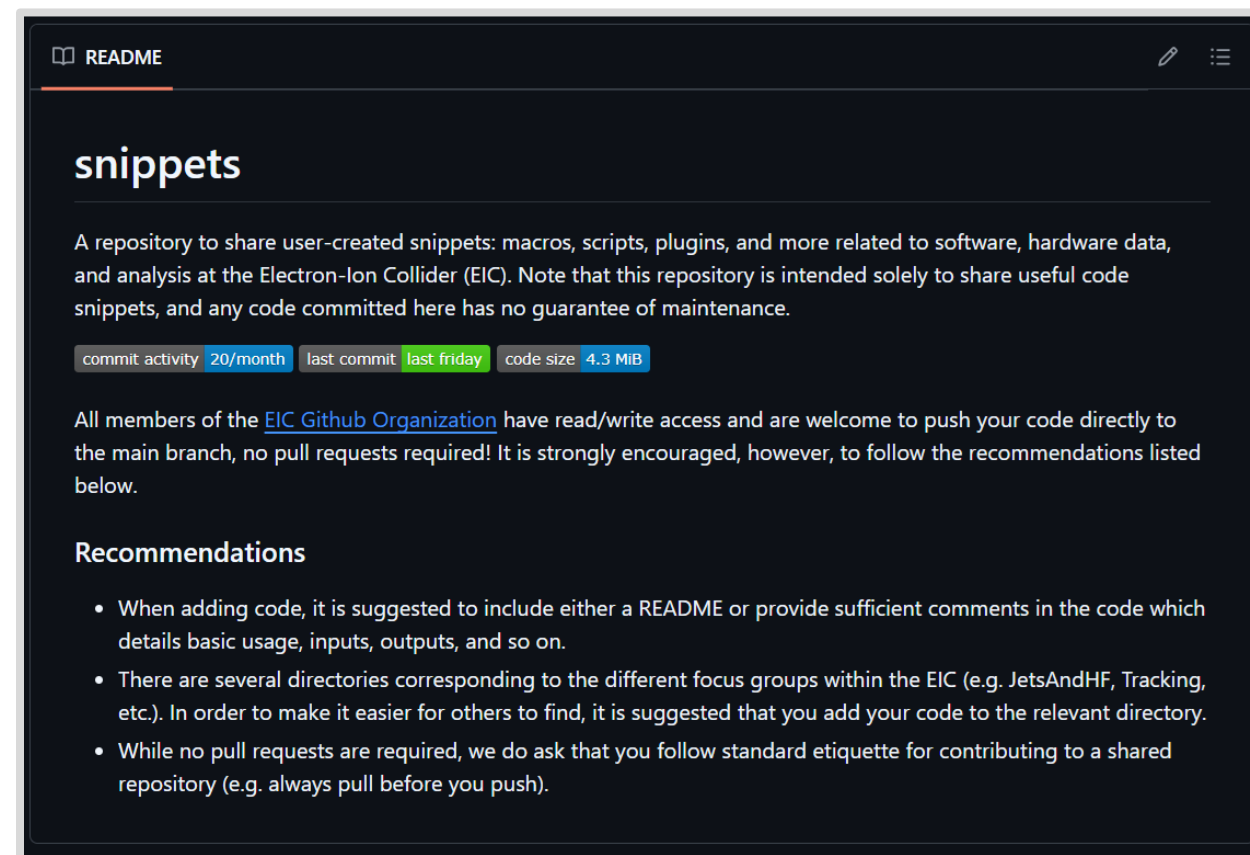 - › Or generating missing docs for snippets
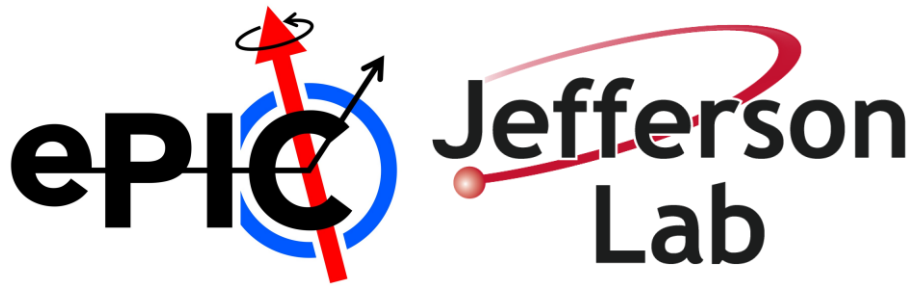
Thanks!

Questions/Discussion

Backup | Snippets | Discussion Context

o Introduced Test Beam Snippets repository at 10.29.2025 Weekly S&C Meeting
  - Concerns were raised about proliferation of snippets-style repos
  - ☞ **So now is a good time to revisit snippets and evaluate its usage**

o **Outline of slides:**
  1) Brief retrospective on snippets
  2) Snapshot of some usage metrics
  3) Concerns and possible paths to ameliorate them
  4) Open discussion

📖 README

snippets

A repository to share user-created snippets: macros, scripts, plugins, and more related to software, hardware data, and analysis at the Electron-Ion Collider (EIC). Note that this repository is intended solely to share useful code snippets, and any code committed here has no guarantee of maintenance.

`commit activity 20/month` `last commit last friday` `code size 4.3 MiB`

All members of the EIC Github Organization have read/write access and are welcome to push your code directly to the main branch, no pull requests required! It is strongly encouraged, however, to follow the recommendations listed below.

Recommendations

- When adding code, it is suggested to include either a README or provide sufficient comments in the code which details basic usage, inputs, outputs, and so on.
- There are several directories corresponding to the different focus groups within the EIC (e.g. JetsAndHF, Tracking, etc.). In order to make it easier for others to find, it is suggested that you add your code to the relevant directory.
- While no pull requests are required, we do ask that you follow standard etiquette for contributing to a shared repository (e.g. always pull before you push).

o Made initial proposal for a repository in
11.01.2023 Weekly S&C Meeting
- Google slides were hosted in my ISU account
  › Have since been deleted, and no notes
    were recorded in live notes...
  › **Lesson learned** 😭
- Will do my best to reconstruct motivation,
  references, etc for the repo

o Followed up with a proposed rollout strategy in
11.15.2023 Weekly S&C Meeting
  ☞ **Initial commit was on 11.16.2023**

o Proposal was made in context of broader discussion on analysis preservation
  – Hence the *AnalysisUtils* name in initial proposal
  – **Point of reference:** the sPHENIX/analysis repository

o **Aim was to provide stable platform for users to share/store analysis-related code**
  – Not intended to be substitute for a full analysis preservation strategy
  – But would fill dual role of
    1) Storing user code, and
    2) Collecting code snippets for documentation
  – Latter is motivation for repo name

o **Key features of repo:**
  – No barrier to commit
    › No PR required,
    › No CI, quality checks, etc.
  – All code is *user*-generated
  – Currently no curation *or maintenance* performed

o **During proposal:** concerns were voiced about lack of curation and maintenance
  – Discussed possible strategy for amelioration, **e.g. appointing a curator or team of curators**
  – Would be responsible for upkeep (e.g. making sure contacts are ID'd for each snippet)

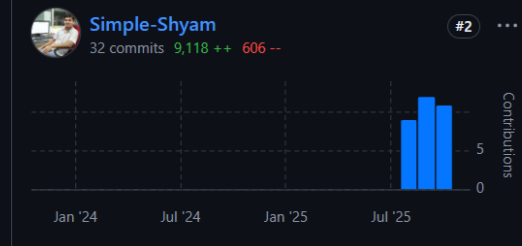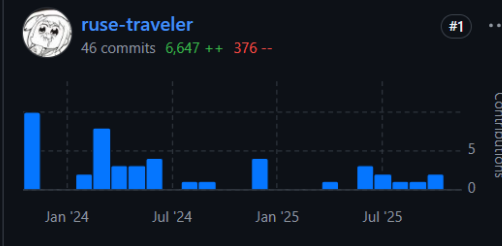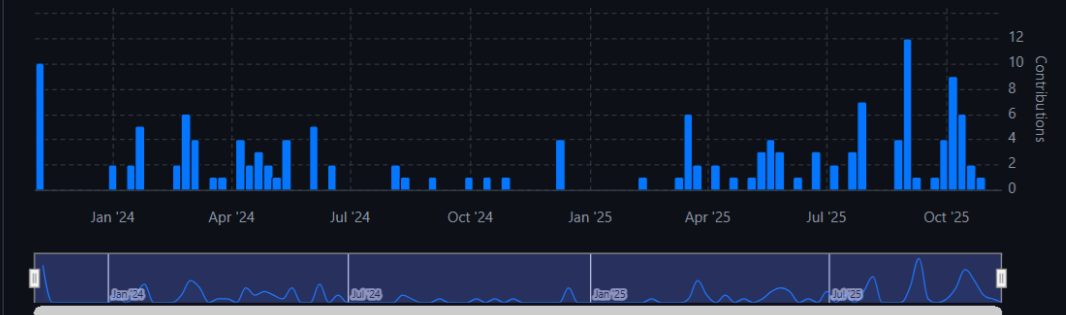- Repo averages 20 commits/month (see screenshot on slide 1)
  - 14 unique contributors
 - 2025 most active year yet (especially last 2 months!)

Code frequency
Additions and deletions per week
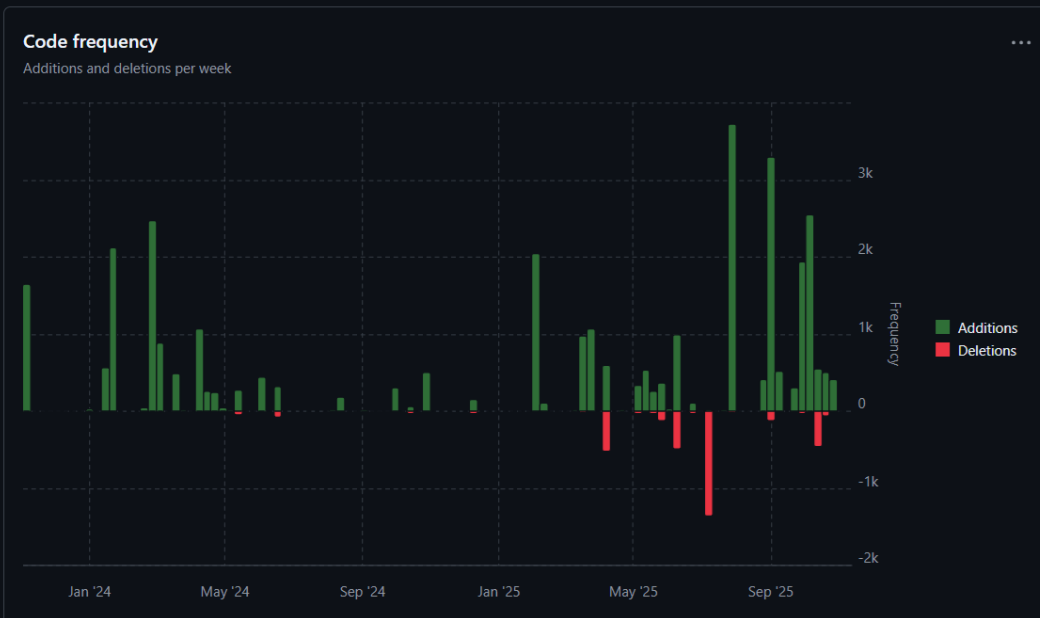
Commits over time
Weekly from Nov 11, 2023 to Nov 8, 2025

ruse-traveler #1
46 commits 6,647 ++ 376 --

Simple-Shyam #2
32 commits 9,118 ++ 606 --

ShujieL #3
12 commits 4,978 ++ 1,437 --

bschmookler #4
10 commits 4,061 ++ 1,088 --

o Most users push directly to main branch, following recommended practice
- Network graph for 10.2025, 11.2025 (top panel) typical for history of repo
- Only 1 instance of a user forking repo (was for the Inclusive Skimmer)

o **As expected:** majority of commits *add* code rather than change/delete code
- Illustrated by frequency plot in lower panel

o **By design:** repo accumulates code in a very "geological" way
- Snippets can become out-of-date, and currently have no protocol to deal with this
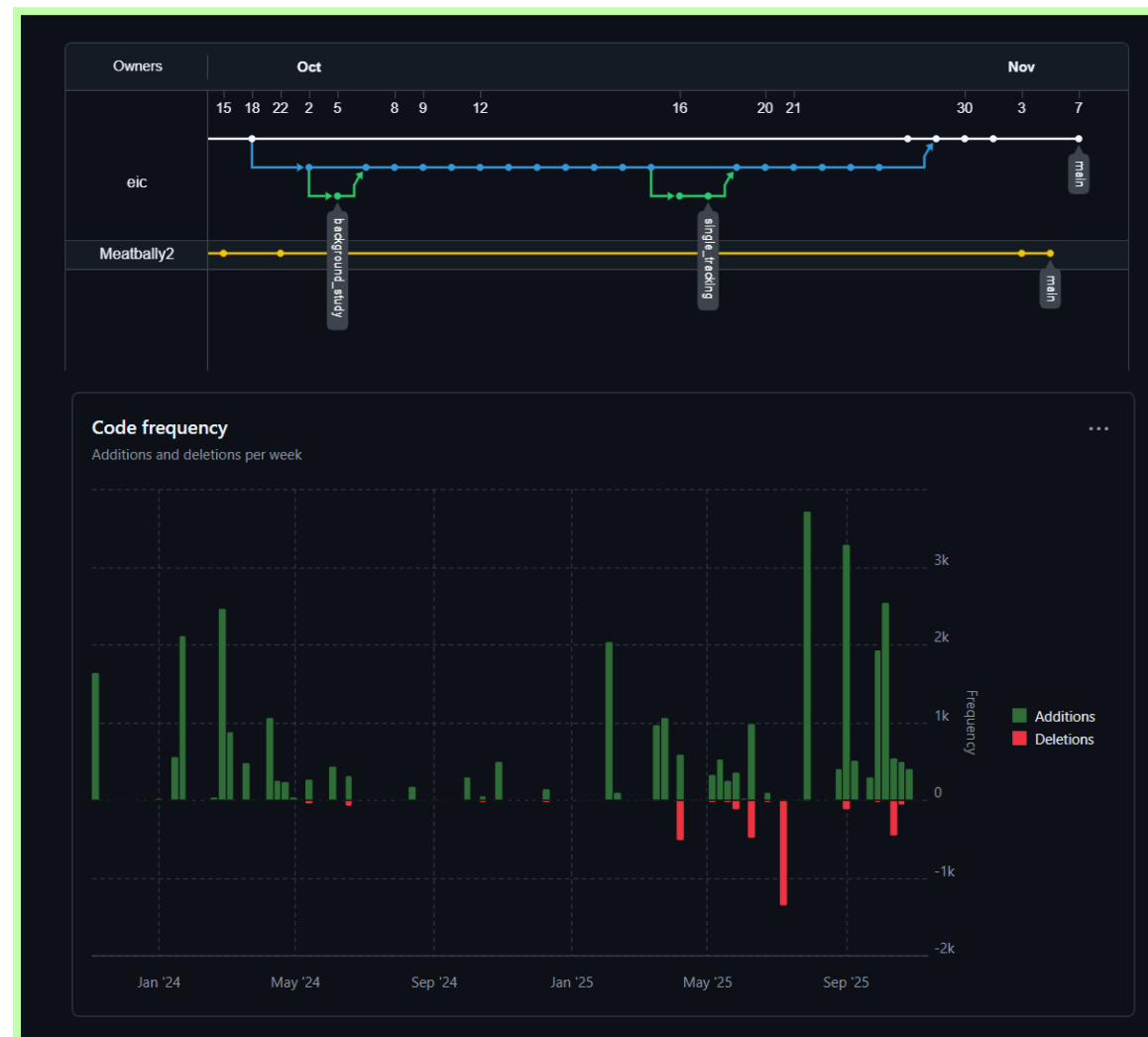
o **Concern 1:** code snippets can grow out of date, and there is no mechanism for catching them
- Can have a deleterious effect on user learning!
- Due to mixed-bag nature of repo, an automated check to identify out-of-date code is tricky…

o **Due to this:** repo doesn't do much for analysis preservation
- No guarantee that code uploaded to snippets will work with future versions of the software stack…

o **Concern 2:** it's not immediately clear *what* code goes in snippets
- Some code might be better served as a dedicated software project with its own repo
- And there is (partly by design) little guidance on what should go in the repo

o As a result, the role this repo fills in our software ecosystem is unclear…

o There are key aspects that I want to preserve while addressing concerns
  – Namely:
    › As low as possible barrier to entry
    › User-generated nature of repo
  – These (to me) are the best parts of the repo

o **Proposed strategy:**
  1) Activate "curator" role
  2) Require PRs with minimal review
  3) Define protocol for code maintenance
  4) Shift focus of repo from *code storage* to *code exchange*
  5) Provide guidance on what *kind* of code should go in repo

o **Curator Role:** would coordinate code additions and maintenance
  – Would be in charge of reviewing PRs and contacting authors when issues arise
  – For now, single person in role would be okay, but would be ideal to evolve to a rotating position with shifts

o **PRs and Maintenance:**
  – Only requirement is that contributed code has documentation attached
    › Can provide script in shell to generate template documentation
    › Primary goal is to ID a contact for snippet
  – When issue ID'd with a snippet, user should file a GH issue and curator will contact/assign author

o **What goes into snippets?** *Snippets*, i.e. "smaller scale code" that *users* think are worth sharing
  - Tricky to define positively, but can work on a "negative" definition
  - Goal would be to catch code that sits in the grey zone where it's not quite big enough to be a full-fledged software project but is still worth sharing…
  - Guidelines will be posted on front-page of repo once ready

o **Emphasizing Exchange:** aiming to take a more active role in curation
  - Repo would be – firstly – a channel to exchange code
    - ☞ i.e. users contribute smaller code they think is useful
  - Snippets identified as being better served in other contexts (e.g. a tutorial) should get moved to suitable location
    - ☞ Also applies to snippets that grow enough in scope

o **This will require active communication with PWGs and DSCs!**