Thoughts on Software Engineering

Aaron Ridley

Version Control

- Important to keep track of changes across platforms
- Collaboration:
  - Don’t keep your own distribution, use capabilities of repository (git, etc.)
  - Talk to others about what you are doing
  - Actually share changes with others!
    - You can’t force people to do this!
- Make distributions and save them
  - Runs should be tagged with distributions
  - I do a poor job of doing this. I feel shame.

Testing

- Short tests to see if code is functional
  - Can catch unexpected consequences of software changes
  - Does not catch physics errors
  - Should be on the core functionality of the code
- Longer tests to see if code is working correctly
  - Should have a variety of tests for physics
  - When physics are updated, it can catch problems
- Maintaining tests is important and takes a long time
  - Can serve as examples for people to use the code

National Academy Report on Open Code

- Should all code that is developed under federal grants be forced to be open?
  - Some people believe quite strongly one way or the other
  - Comparison to data. Comparison to hardware.
- Acknowledgement that software development is expensive
  - Roughly 1/3 of code is writing code
  - The rest is commenting, documenting, testing, supporting.
- Reproducibility/Repeatability
  - Should you be able to exactly reproduce a figure from a paper?
    - What does this prove?
  - Should you be able to reproduce the idea of a paper using some sort of code?
    - Maybe the same code? Maybe a different code?
- Recommendations:
  - Do nothing: probably not going to happen
  - Force open code: a lot of resistance to this
    - Every proposal MUST have software development plan
  - Bribery: offer carrots to researchers to participate
    - Make special calls for opening codes
    - Make additional funds available for calls to support open code