

SOFTWARE ENGINEERING AND COMMUNITY CODES TRACK IN ATPESC

**ANSHU DUBEY
KATHERINE RILEY**

WSSSPE 2016
13 September 2016

MOTIVATION

- To bring knowledge of **useful** software engineering practices to HPC scientific code developers
 - Not to prescribe any set of practices as **must use**
 - Be informative about practices that have worked for some projects
 - Emphasis on adoption of practices that help productivity rather than put unsustainable burden
 - Customization as needed – based on information made available
- We do it through examples and case studies
 - References for available resources
 - Suggestions for further reading



INTEREST FROM STAKEHOLDERS

- Facilities
 - Well prepared users better use machines
- Funding agencies
 - No reinventing the wheel over and over again
- Researchers
 - Offload some of the nitpicky work
 - Go further in getting results
- Community as a whole
 - More credibility with reproducible results and process



TOPICS

- Have evolved over the years
- Unchanging ones
 - Versioning, testing, documentation, build and configuration
 - Some form of provenance and sociology
- Changes
 - Early years had more specific community presentations
 - Idea was to give concrete examples to make the students appreciate the impact
 - 2016 had some advanced topics such as code coverage in testing, continuous integration
 - Also included software refactoring



EVOLUTION SPECIFICS

- Early years software carpentry material for basic topics
- IDEAS project helped produce material tailored for scientific computing
 - Was presented in a series of highly successful webinars
- Community specific presentations were getting repetitive
 - Replaced with advanced topics and one presentation that demonstrated the use of many topics in one project
- **Student feedback from 2016 tells us we need to change more content**



RESPONSE

- Has varied from year to year and within the year
 - Somewhat reflective of student bias
 - Those who appreciate the track tend to mention having confronted difficulties without using software engineering, or some exposure to the need for it
 - Also reflective of student sophistication
- There has been a dramatic difference in the last couple of years
 - Generally high level of awareness



EXPERIENCE FROM 2016

- A sudden jump in awareness
- We thought we had good material
 - Similar material in webinar series received excellent reviews and massive attendance
 - For ATPESC we made it more interactive
- We found we were preaching to the choir on the motivation side
- Several presentations found substantial chunks to be redundant
 - Students were already familiar with it



TAKEAWAYS

- Student body still has those that see the whole track as a waste of their time
 - We ignore them
 - But, it is a more hand wavy topic compared to others in the program because formalization of issues specific to scientific computing is still limited
 - Experience reports can be repetitive, but general coverage without examples is not very helpful either
- We, the lecturers need better coordination among ourselves, and be prepared with many back up slides on advanced issues in the topic we are covering.



WHERE WE ARE NOW

- The balancing act has become tougher
 - Definite need for more advanced material
- It may be feasible to introduce more advanced topics
 - Performance portability
 - Higher degree of abstractions in design
- In general we welcome community input and suggestions
 - Experience others may wish to share

