

AI for Software Development and Maintenance

Presented by



<https://pesoproject.org>

Member of



*Consortium for the Advancement
of Scientific Software*

<https://cass.community>

With prior support from



See slide 2 for
license details

David E. Bernholdt (he/him)
Oak Ridge National Laboratory

Software Sustainability track @ Argonne Training Program on Extreme-Scale Computing summer school

License, Citation and Acknowledgements

License and Citation



- This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/) (CC BY 4.0).
- **The requested citation the overall tutorial is:** David E. Bernholdt, Software Sustainability track, in Argonne Training Program on Extreme-Scale Computing, St. Charles, Illinois, 2026. DOI: [10.6084/m9.figshare.33139910](https://doi.org/10.6084/m9.figshare.33139910).
- Individual modules may be cited as *Speaker, Module Title, in Tutorial Title, ...*

Acknowledgements

- This work was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Next-Generation Scientific Software Technologies (NGSST) and Scientific Discovery through Advanced Computing (SciDAC) programs.
- This work was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research (ASCR), and by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.
- This work was performed in part at the Argonne National Laboratory, which is managed by UChicago Argonne, LLC for the U.S. Department of Energy under Contract No. DE-AC02-06CH11357.
- This work was performed in part at the Lawrence Livermore National Laboratory, which is managed by Lawrence Livermore National Security, LLC for the U.S. Department of Energy under Contract No. DE-AC52-07NA27344.
- This work was performed in part at the Los Alamos National Laboratory, which is managed by Triad National Security, LLC for the U.S. Department of Energy under Contract No. 89233218CNA000001
- This work was performed in part at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.
- This work was performed in part at Sandia National Laboratories. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

AI is a New Tool for Scientific Software, and Rapidly Evolving

- This session is meant to be an exploration of current usage (or not) of AI tools in developing and maintaining scientific software, and resulting experiences (good or bad)
 - Focus on using AI in conjunction with the software per se, distinct from using AI tools as part of the scientific approach (e.g., hypothesis generation, research assistant, surrogate models, classification, analysis, etc.)
- I am not advocating for any particular approach or usage model
- This session is short, so we won't be able to have detailed discussions
 - Mostly looking for shows of hands
 - With a few examples or clarifying comments (keep them brief)

What are you Using (or Experimenting with) AI For?

- Understanding existing code
- Debugging code
- Peripheral maintenance tasks
 - E.g., documentation and other tasks that don't directly involve the code itself
- Code reviews
- Modifying or generating code
 - Test generation
 - Refactoring code (i.e., changing code without changing functionality)
 - Adding new features
 - Implementation advice (e.g., investigating algorithms, getting help with APIs, etc.)
 - Generating non-trivial new code bases from scratch (i.e., vibe coding)
- Other uses

AI in Projects?

- **Level of formality**

- My AI activities are for personal use only – nobody else has access to the results
- My AI activities relate to a larger project, but nobody else knows (or cares) that I'm using AI
- Some or all of my project team is using AI, but there are no project-level policies or strategies
- My project has discussed? defined? documented? AI usage strategies and/or policies

- **Acceptance of AI-generated code**

- My project will not accept it (e.g., licensing concerns, ethical concerns, etc.)
- My project accepts it without any special annotation or acknowledgment
- My project accepts it only if it is clearly marked as such

- AI-code gets more? less? similar? scrutiny than human-written code in my project

- I'm aware of the legal implications of AI-generated code

- Copyrightability and licensing
- Concerns over licensing of training sources

Your Experience of AI for Software Maintenance and Development, to Date?

- I'm using commercial? free/open? LLMs
- I'm using general? specialist (i.e., code-focused)? LLMs
- The good?
- The bad?
- Is it hype or is it real?
- Does *AI for software maintenance and development* help or hurt your science, overall?
- I wish...

Final Discussion

Strawman

The availability of AI tools doesn't fundamentally change our expectations of "good" scientific software and "good" development/maintenance practices – it merely changes how they might be achieved

- If you disagree, what expectations need to change to accommodate AI?
- If you agree, what should we be doing to ensure we produce good software with AI help?