




# Motivation and Overview

Greg Watson (he/him)
Oak Ridge National Laboratory

Developing a Testing and Continuous Integration Strategy for your Team tutorial @ Exascale Computing Project Annual Meeting

Contributors: David E. Bernholdt (ORNL), Anshu Dubey (ANL), Patricia A. Grubel (LANL), Rinku K. Gupta (ANL), Katherine M. Riley (ANL), Gregory R. Watson (ORNL)

# License, Citation and Acknowledgements

## License and Citation

- This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

- **The requested citation the overall tutorial is: Gregory R. Watson and David M. Rogers, Developing a Testing and Continuous Integration Strategy for your Team tutorial, in Exascale Computing Project Annual Meeting, online, 2022. DOI: 10.6084/m9.figshare.19608927**
- Individual modules may be cited as *Speaker, Module Title*, in Better Scientific Software tutorial…

## Acknowledgements


- 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 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 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 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.


IDEAS productivity

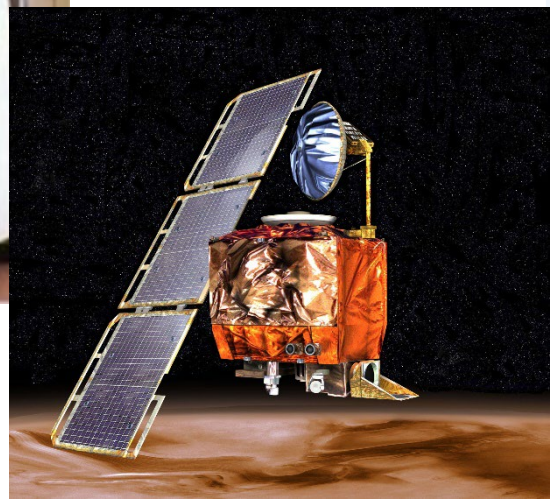
ECP EXASCALE COMPUTING PROJECT

Science through computing is,
at best,
as credible as the software that produces it!

# High-Consequence Software-Related Scientific Failures

## Therac-25 (1985-1987)

- Computer-controlled radiation therapy system
- **Poor software design, development and testing practices** allowed flaws that let to at least six cases of substantial radiation overdoses, three fatal



## Mars Climate Orbiter (1999)

- Incorrect trajectory adjustment caused loss of the orbiter as it was supposed to enter Martian orbit
- Discrepancy in the units used in two different software components
- One component **didn't follow specifications**
- **Inadequate testing** at the interface
- Concerns raised earlier in the mission were ignored because they **weren't properly documented**

*Just two of many examples*

IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

# Challenges Developing Scientific Applications Today

## Technical

- All parts of the model and software system can be under research
- Requirements change throughout the lifecycle as knowledge grows
- Verification complicated by floating point representation
- Real world is messy, so is the software
- Increasing architectural diversity

## Sociological

- Competing priorities and incentives
  - Sponsors often care more about scientific publications than software per se
  - Balancing development and maintenance with research
- Limited resources
- Need for interdisciplinary interactions
  - Many different kinds of expertise to be successful

IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

# Best Practices for Scientific Software Development

## Baseline

- Invest in extensible code design
- Use version control and automated testing
- **Institute a rigorous verification and validation regime**
- **Define and enforce coding and testing standards**
- Clear and well-defined policies for
  - Auditing and maintenance
  - Distribution and contribution
  - Documentation

## Desirable

- Provenance and reproducibility
- Lifecycle management
- Open development and frequent releases

***This tutorial will focus primarily on scientific software as distinct from more generic software engineering best practices***

IDEAS productivity

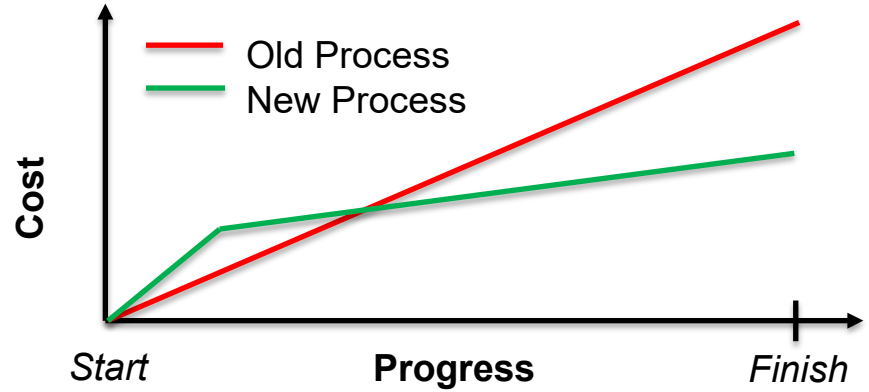
ECP EXASCALE COMPUTING PROJECT

# Continual, Incremental Software Process Improvement

Target: your project should include "just enough" software engineering so that you can meet your short-term and longer-term scientific goals effectively

1. Identify your team's "pain points" in your software development processes
   - Help: RateYourProject assessment tool: https://rateyourproject.org/
2. Set a goal for something to improve
   - Target processes and behaviors, not just tasks
   - Pick something that you can address in a few months that will give you a noticeable benefit
3. Agree on a plan to address it, identify markers of progress and what is "done"
   - Write them down
   - Help: Progress tracking card examples: https://bssw-psip.github.io/ptc-catalog/catalog
4. Work your plan, track your progress
5. When you are done, celebrate…

…then pick a new pain point to address




*The new process costs something to implement, but it pays off over time*

Productivity and Sustainability Improvement Planning
https://bssw.io/psip

A goal of BSSw.io is to provide resources for improving your software processes. If you find useful resources that aren't on BSSw.io, consider contributing. Its easy and quick.

IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

# Agenda

The agenda is also available on the tutorial web page. Visit https://bssw-tutorial.github.io and click on the link for today's tutorial

| Time (EDT) | Module | Title | Presenter |
|---|---|---|---|
| 2:30 PM | 0 | Introduction and Setup | Gregory R. Watson (ORNL) |
| 2:35 PM | 1 | Motivation and Overview | Gregory R. Watson (ORNL) |
| 2:40 PM | 2 | Software Testing Introduction | Gregory R. Watson (ORNL) |
| 3:00 PM | 3 | Testing Complex Software | David M. Rogers (ORNL) |
| 3:25 PM | 4 | Continuous Integration | David M. Rogers (ORNL) |
| 3:55 PM | 5 | Summary | David M. Rogers (ORNL) |
| 4:00 PM | | *Adjourn* | |