



# Summary

David E. Bernholdt, David Rogers  
Oak Ridge National Laboratory

Better Scientific Software Tutorial, SC20, November 2020



See slide 2 for  
license details

# 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, Anshu Dubey, Patricia A. Grubel, Rinku K. Gupta, Better Scientific Software tutorial, in SC '20: International Conference for High Performance Computing, Networking, Storage and Analysis, online, 2020. DOI: [10.6084/m9.figshare.12994376](https://doi.org/10.6084/m9.figshare.12994376)**
- Individual modules may be cited as *Speaker, Module Title*, in Better Scientific Software tutorial...

## Acknowledgements

- Additional contributors include: Mike Heroux, Alicia Klinvex, Mark Miller, Jared O'Neal, Katherine Riley, David Rogers, Deborah Stevens, James Willenbring
- 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.

# Software Under the Microscope

- Mar. 16: Neil Ferguson (Imperial College) briefed UK Parliament on epidemiological modeling of COVID-19 pandemic
  - Epidemiological models like this helped prompt government action, but have lots of assumptions
- April 1: Nicholas Lewis (independent climate science researcher in UK) can't easily see where some of the assumptions come from – publishes a blog article
  - “Moreover, the computer code... is old, unverified, and documented inadequately, if at all...”

<https://doi.org/10.25561/77482>

<https://www.nicholaslewis.org/imperial-college-uk-covid-19-numbers-dont-seem-to-add-up/>

<https://www.nature.com/articles/d41586-020-01003-6>

16 March 2020

Imperial College COVID-19 Response Team

## Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand

Neil M Ferguson, Daniel Laydon, Gemma Nedjati-Gilani, Sangeeta Bhatia, Adhiratha Boonyasiri, Zulma Cucunubá, Dorigatti, Han Fu, Katy Gaythorpe, Will Green, Arran Hall, Elsie Elvins, Hayley Thompson, Robert Verity, Erik Volz, Haowei Cao, Caroline Walters, Peter Winskill, Charles Whittaker, Chris Jarvis

On behalf of the Imperial College COVID-19 Response Team

WHO Collaborating Centre for Infectious Disease Modelling  
MRC Centre for Global Infectious Disease Analysis  
Abdul Latif Jameel Institute for Disease and Emergency Analytics  
Imperial College London

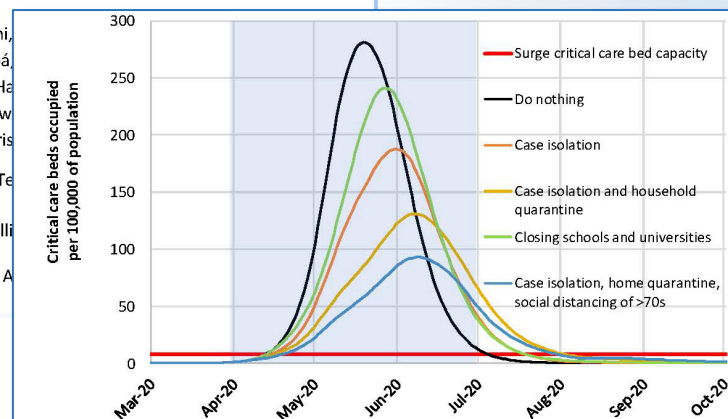


Figure 2: Mitigation strategy scenarios for GB showing critical care (ICU) bed requirements. The black line shows the unmitigated epidemic. The green line shows a mitigation strategy incorporating closure of schools and universities; orange line shows case isolation; yellow line shows case isolation and household quarantine; and the blue line shows case isolation, home quarantine and social distancing of those aged over 70. The blue shading shows the 3-month period in which these interventions are assumed to remain in place.

## Imperial College UK COVID-19 numbers don't seem to add up

### Introduction and summary

A study published two weeks ago by the COVID-19 Response Team from Imperial College (Ferguson20<sup>1</sup>) appears to be largely responsible for driving UK government policy actions. The study is not peer reviewed; indeed, it seems not to have been externally reviewed at all. Moreover, the computer code used to produce the estimates in the study – which on Ferguson's own admission is old, unverified and documented inadequately, if at all – has still not been published. That, in my view, shows a worrying approach to a matter of vital public concern.

# The Press Picks Up the Story

*Headline and quotes from the Fox News article*

“In our commercial reality, **we would fire anyone for developing code like this** and any business that relied on it to produce software for sale would likely go bust,” David Richards, co-founder of British data technology company WANdisco, told the Daily Telegraph.

“**Models must be capable of passing the basic scientific test of producing the same results given the same initial set of parameters**...otherwise, there is simply no way of knowing whether they will be reliable,” said Michael Bonsall, Professor of Mathematical Biology at Oxford University.

CORONAVIRUS · Published May 16

## Imperial College model Britain used to justify lockdown a 'buggy mess', 'totally unreliable', experts claim

Scientists from the University of Edinburgh have further claimed that it is **impossible to reproduce** the same results from the same data using the model. The team got different results when they used different machines, and even different results from the same machines.

“There appears to be a bug in either the creation or re-use of the network file. If we attempt two completely identical runs, only varying in that the second should use the network file produced by the first, the results are quite different,” the Edinburgh researchers wrote on the Github file. A fix was provided, but it was **the first of many bugs** found within the program.

<https://www.foxnews.com/world/imperial-college-britain-coronavirus-lockdown-buggy-mess-unreliable>  
<https://www.telegraph.co.uk/technology/2020/05/16/coding-led-lockdown-totally-unreliable-buggy-mess-say-experts/>

# What you May Not Have Heard

- April 22: Imperial collaborates with Microsoft to refactor and clean up the code, which is released on GitHub
- May 10: Phil Bull rebuts criticisms of the Imperial code
  - Which spurs further discussions within some groups focused on scientific software
- May 29: CODECHECK independently reproduces results of Imperial's Report 9


<https://github.com/mrc-ide/covid-sim/>  
<https://philbull.wordpress.com/2020/05/10/why-you-can-ignore-reviews-of-scientific-code-by-commercial-software-developers/amp/>  
<http://doi.org/10.5281/zenodo.3865491>

*tl;dr: Many scientists write code that is crappy stylistically, but which is nevertheless scientifically correct (following rigorous checking/validation of outputs etc). Professional commercial software developers are well-qualified to review code style, but most don't have a clue about checking scientific validity or what counts as good scientific practice. Criticisms of the Imperial Covid-Sim model from some of the latter are overstated at best.*

CODECHECK certificate 2020-010

<https://doi.org/10.5281/zenodo.3865491>



| Item      | Value   |
|-----------|---|
| Title     | Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand. March 16, 2020. |
| Authors   | Neil Ferguson  , COVID-19 Response Team    |
| Reference | Imperial College Preprint <a href="https://doi.org/10.25561/77482">https://doi.org/10.25561/77482</a>                           |

# Some Observations

- Your code is likely to live longer than you expect, and may be used in ways you don't expect by people you don't know – plan for it!
- Increasingly, consequential decisions are made based on computational results
  - The codes generating those results may (justifiably) be subject to greater scrutiny
- The scientific credibility of software is strongly connected to good software engineering practices
  - Documentation
  - Testing, verification, and (where possible) validation
  - Code readability and quality metrics

Question: Should we excuse scientific software for being “crappy stylistically”?

Hint: crappy code can hide bugs

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

# Today, We Covered Many Topics...

- Project management
- Collaboration around software development
- Designing software for flexibility and extensibility
- Testing strategies for complex software systems
- Systematic refactoring of large, complex software systems
- Continuous integration testing
- Reproducibility

# And there are Many More We Didn't Have Time For

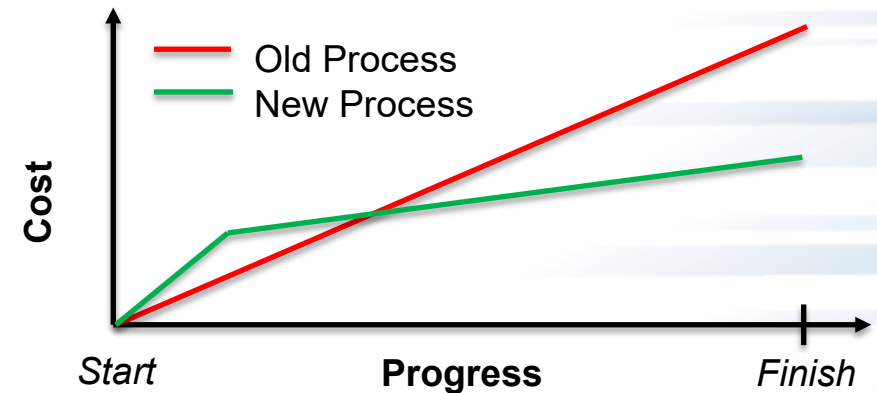
- Documentation
  - Licensing
  - Packaging and distribution
  - Issue tracking
  - Configuration and build
  - Debugging strategies
  - Building and sustaining communities around software
  - Software publication and citation
  - Requirements gathering
  - Understanding and debugging floating-point math
  - Performance and performance portability
  - ...
- **Also important topics, but...**
  - Less distinction between research software and other software
  - More informational resources available
  - Next-level concerns for starting researchers
  - There's only so much time in the day!

But you're a researcher.  
You can't afford to spend  
“all” of your time on  
software engineering.

# A Final Recommendation: 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

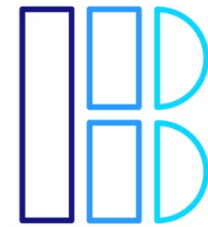
1. Identify your team’s “pain points” in your software development processes
  2. Set a goal for something to improve
    - Target processes and behaviors, not just tasks
    - Pick something that you can address in a few months that will give you a noticeable benefit
  3. Agree on a plan to address it, identify markers of progress and what is “done”
    - Write them down
  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](https://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

ECIP  
EXASCALE  
COMPUTING  
PROJECT

# Thanks, and Keep in Touch!

- **Please complete an evaluation of this tutorial**
  - <https://submissions.supercomputing.org/?page=Submit&id=TutorialEvaluation&site=sc20>
- Email comments and questions to [bssw-tutorial@lists.mcs.anl.gov](mailto:bssw-tutorial@lists.mcs.anl.gov)
- We'll provide feedback on pull requests in the hands-on repository
- Follow us:
  - IDEAS Productivity project
    - Website: <https://ideas-productivity.org/>
    - Announcement list: <http://eepurl.com/cQCyJ5>
  - Better Scientific Software
    - Resources: <https://bssw.io/>
    - Monthly digest: <https://bssw.io/pages/receive-our-email-digest>
    - RSS feed: <https://bssw.io/items.rss>

