Improving Throughput of Science in CLM – via Software Testing

Erik Kluzek, Software Engineer Land Model Working Group Liason
National Center for Atmospheric Research (NCAR)

NESL

U.S. DEPARTMENT OF ENERGY

NSF

NCAR is sponsored by the National Science Foundation

Summary for the Impatient

- Scientists need to run automated test suites – sorry!
- Software engineers (S.E.'s) make this easy. There are many test-suites and test-lists available.
- Testing during development will speed up the process.
- I'd like to see more use of unit-testing.

NCAR is sponsored by the National Science Foundation

Science trumps Software

- Science advancement is where funding comes from.
- Beautiful software that is NOT advancing scientifically will kill us.
- So software development is a service to the science (view myself in a service role, help scientists out, don't write papers, don't go to outside meetings etc.).
- BUT
- The tool for that science advancement IS SOFTWARE (CESM IS A SOFTWARE PACKAGE).
- AND BAD SOFTWARE CAN GET IN THE WAY OF SCIENCE ADVANACEMENT.

NCAR is sponsored by the National Science Foundation

Scientific Method meets Software

- ***Reproducible*** -- You and/or someone else can reproduce your work.
- ***Correct*** -- The equations in your paper ARE what you've actually done.
- ***Experiment Controls*** **–** You can run a control for your experiment and have the ability to show that an experiment doesn't significantly change the climate from the control.
- ***Continual experiments*** -- The only way the science advances is with experiments. That means the code MUST continually change. Different groups use the same tool – so multiple configurations and options are mandatory.

NESL

NCAR is sponsored by the National Science Foundation

NSF

Software Methods for Science

- ***Reproducible*** – answers are the same when you run more than once, restarts cause answers to be identical, different number of processors gives same results.
- ***Correct*** -- The equations in your paper ARE what you've implemented in code.
- ***Experiment Controls*** – see reproducible. But, also that memory issues, machine precision, machine differences, or numerical instability don't cause large differences for small changes.
- ***Continual experiments*** **--** Since the code is continually changing – an agile method is the only Software Methodology that makes sense. Code needs to be continually refactored and unused features removed to be maintainable.

NESL

NCAR is sponsored by the National Science Foundation

NSF

SE's do all Testing is Unworkable

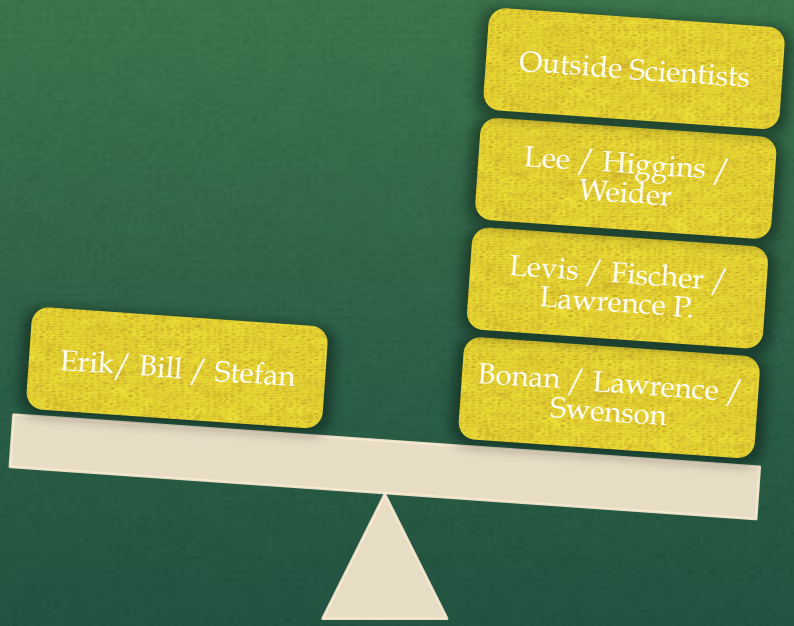
- My inclination has been that I'll do the Software testing for the scientists.
- I've discovered that solution is unworkable.
- It's a problem of scale.

NCAR is sponsored by the National Science Foundation

CLM Scientists to S.E. (>10X)

SE's

Scientists



NCAR is sponsored by the National Science Foundation

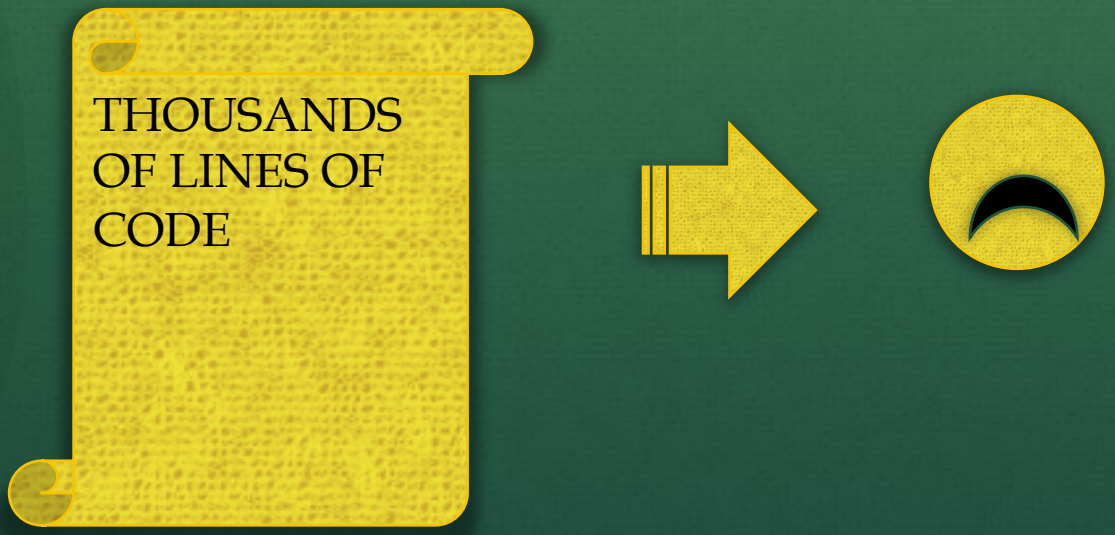
Finding Problems in a Million Lines of Code

- You can only actively debug a small number of lines of code per day. If a problem can be anywhere in CESM (a million lines of code) it can take months (or could honestly be years) to find.
- If only the new code is suspect – you can find problems by closely examining the new code differences. If the new code is large – that's a problem as well.
- Modularity also helps with finding problems – but CLM has a global type that is used by all subroutines through side-effects rather than clear arguments.
- You may test your changes for your own case, but have problems when your code is run for other configurations. Automated testing helps you with this.

NCAR is sponsored by the National Science Foundation

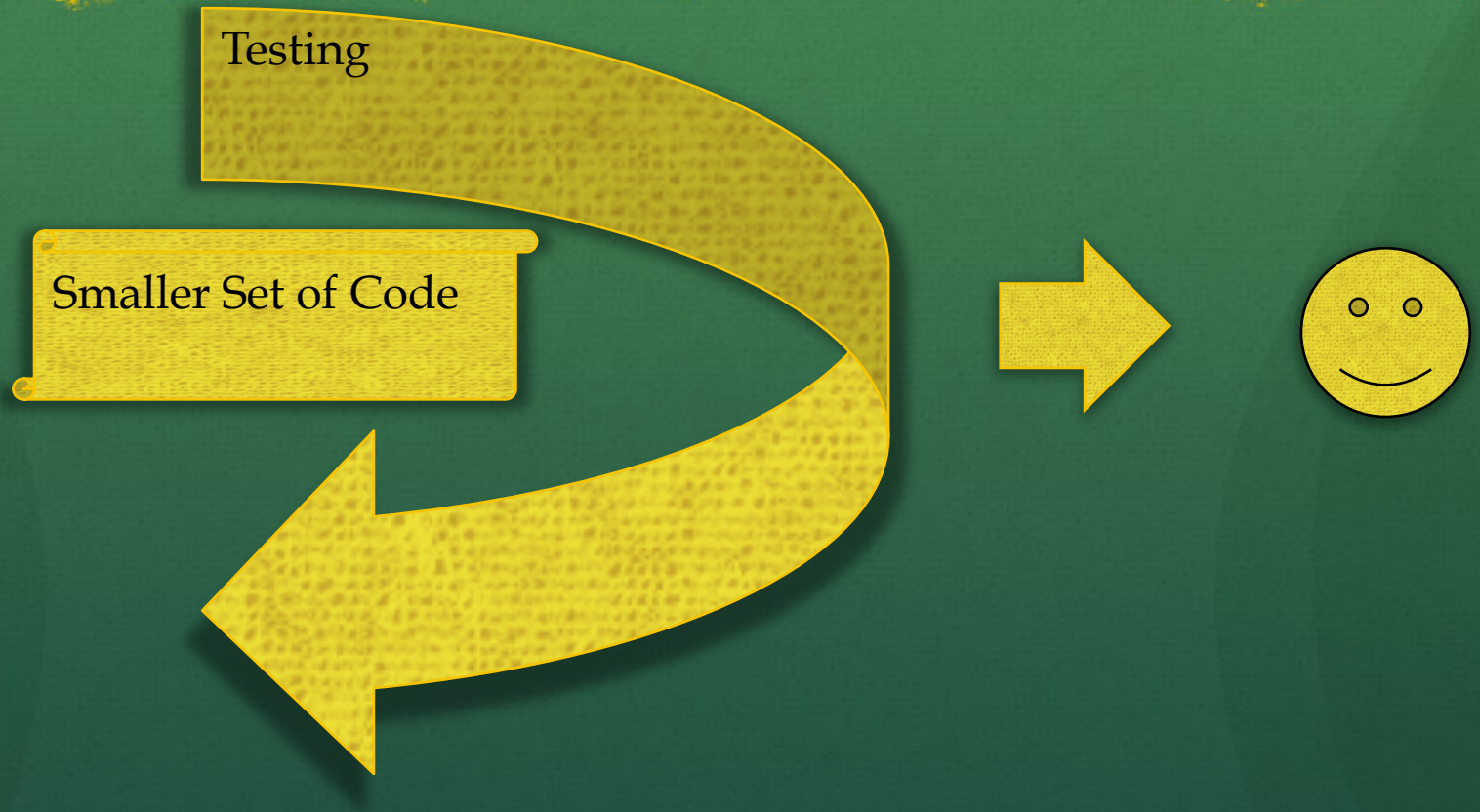
Scale Problem with Code

- SE's only get code late in process. Over the last year I've been integrating code changes from several thousand to fifty thousand lines. Problems can take months to solve.



NCAR is sponsored by the National Science Foundation

Test During Development



NCAR is sponsored by the National Science Foundation

Suggestions for Test Driven Development for Science

- Scientists DO test for correctness using various methods.
- Sometimes those tests are unit-tests that instead of throwing away we could continue to use (i.e Gordon Bonan).
- Sometimes they are test cases or options that could continue to be maintained.
- Typically that testing is for your own science – let's use tools the S.E. have provided to make sure other options and configurations continue to work and test restarts, multi-processing etc.

NCAR is sponsored by the National Science Foundation

Automated Testing

- Automated testing for restarts, and different processor counts, that everyone can use.
- Automated testing that runs in both production and debug modes (with float trapping and subscript checking on).
- Testing on multiple machines and compilers (rule of 3)
- Test to try to exercise ALL of the code (and code options) and bounds of branch points (when you put your testing hat on you actually try to BREAK the code. Better to find a problem early than later).
- Run different configurations, namelist options and resolutions. Global low resolution simulations to exercise code over a wide range of states. Also run with a variety of coupling options to data or active model components.
- Unit tests to validate science of individual modules.
- Long control simulations with scores of people examining to validate science.
- Bit for bit experiments to validate answers don't change from long control simulations.

NCAR is sponsored by the National Science Foundation

Software Engineering Myths

NCAR is sponsored by the National Science Foundation

Software Engineering Myths

- I should leave the old code commented out to show what was done before
- I should add new capability that will be used later – but don't worry about testing it.
- If I write the code and carefully examine it – it's probably fine.
- Code that's already written doesn't need to be tested again.

NCAR is sponsored by the National Science Foundation

Test Driven Development

- Words To Live By:
- If you didn't test something – it's broken.
- If you don't continue to test everything you need – it'll get broken.

NCAR is sponsored by the National Science Foundation

Testing Levels

- Different types of tests catch different types of problems.
- System Testing: (Run the whole CESM)
  - Tests the final package. Global simulations test extreme conditions.
  - Can start near end of month, end-of-year for end-of-month/year issues.
  - Can also run single-point/small region studies for multi-year to find other types of problems.
- Unit Testing (Test single modules or subroutines)
  - FAST!
  - Can also test error conditions.
  - Easier to establish that all lines of code are tested.
  - Can test parts of code for 100's (or even thousands) of simulation years

NCAR is sponsored by the National Science Foundation

If you only have a hammer…

- Every problem looks like a nail…
- The bulk of our CESM testing is at the system level – but it doesn't work well for everything…
  - Even at low resolution, building, and running even CLM-only cases for a few simulation days is time-consuming.
  - Running ALL resolutions is IMPOSSIBLE!
  - Running for many simulation-years is impractical even for single-point.
  - Awkward to test error conditions.
  - Don't really know if you are hitting everything that should be tested.

NCAR is sponsored by the National Science Foundation

Improving Our Testing -- Unit Testing

- I've been advocating unit testing since 2001 when I did a talk on Extreme Programming (that's 11 years ago)!
- There are now eight unit-tests in csm_share.
- Unit-testers for Utilities: ESMF, ESMF-WRF-timemgr, and PIO
- Four unit tests for perl5lib.
- A unit tester for CLM and datm build-namelist.

NCAR is sponsored by the National Science Foundation

Summary for those that Lasted Through…

- Scientists need to run automated test suites – sorry!
- Software engineers make this easy. There are many test-suites and test-lists available.
- Testing during development will speed up the process.
- I'd like to see more use of unit-testing.

NCAR is sponsored by the National Science Foundation

Questions?

NCAR is sponsored by the National Science Foundation

Science is King

- ***My missions statement is: CLM Science is premier here.***
- My purpose is to support CLM science with good quality software to enable the science of CLM to happen: robustly, easily, flexibly, and quickly (with no sacrifice of quality).
- BUT BAD SOFTWARE CAN GET IN THE WAY OF SCIENCE ADVANACEMENT.
- And it continually does.
- The relatively small number of software engineers to scientists – means scientists MUST also practice the good software engineering to advance the science.

NCAR is sponsored by the National Science Foundation

Scripts/XML files now a Significant Part of the Code

- For CLM 20%
- Common problems are in XML filenames either filename is incorrect or doesn't match metadata describing it.

NCAR is sponsored by the National Science Foundation

Outline

- Summary for the impatient
- Science trumps software
- Scientific method meets Software (vice versa).
- The end product is software so software methods and testing have to be done by all

NCAR is sponsored by the National Science Foundation

Code is NOT Static!

- All of CESM 1M
- CLM core code: 100k
- bld: 15k
- test/system: 5k
- drv/datm: 30k
- utils: 100k

NCAR is sponsored by the National Science Foundation