Testing of HPC Scientific Software Introduction

Presented at
Better Scientific Software tutorial

ECP 2nd Annual Meeting, Knoxville, Tennessee

Jared O'Neal
Argonne National Laboratory

EXASCALE COMPUTING PROJECT

License, citation and acknowledgements

Acknowledgements

- Alicia Klinvex
- 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.

Why is testing important?

Therac-25 computer-controlled radiation therapy machine

- Minimal software testing
- System administers radiation in improper setup
- Unlucky patients were struck with approximately 100 times the intended dose
- Recalled after six accidents resulting in serious injury
- Race condition in the code went undetected

Leveson & Turner, "An Investigation of the Therac-25 Accidents". IEEE (1993).

Why is testing important?

Ariane 5 Launch Vehicle

- Launched to lift heavy payloads into low Earth orbit
- Initial rocket driven off course, started to disintegrate, and destroyed with 40 seconds of launch
- SW shutdown when 64bit fixed-point to 16bit int overflowed
- Used directly SW from Ariane 4 - overflow physically impossible
- Conversion never tested

Bashar Nuseibah, "Ariane 5: who dunnit?". IEEE Software (1997).
http://iansommerville.com/software-engineering-book/case-studies/ariane5/

Why is testing important?

Protein structures in scientific software

- Inherited data analysis code interchanged two data columns
  - inverted electron-density map
  - incorrect protein structure
- Retracted 5 publications
  - One was cited 364 times
- Many papers and grant applications conflicting with his results were rejected

Greg Miller, "A Scientist's Nightmare: Software Problem Leads to Five Retractions". Science (2006).

How common are bugs?

Programs do not acquire bugs as people acquire germs, by hanging around other buggy programs. Programmers must insert them.
- Harlan Mills

Industry average for delivered software

- 1-25 errors per 1000 lines of code

Microsoft Applications Division

- 10-20 defects per 1000 lines of code during in-house testing
- 1 defect per 2000 lines of code in released product

Steven McConnell, "Code Complete". Second Edition (2004).

Avoid debugging

Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it.
- Brian Kernighan

- Upfront effort can lead to clean code
- Integrated tests encode result of time and effort
- Integrated tests can ease debugging

Benefits of testing

- Promotes high-quality software that delivers correct results and improves confidence
- Increases quality and speed of development, reducing development and maintenance costs
- Maintains portability to a variety of systems and compilers
- Automated testing helps create code that isn't brittle

Definitions

Classes of Tests by Granularity / Hierarchy

- Unit tests
  - Test individual functions or classes
- Component or Integration tests
  - Test linkage of functions and classes into higher-functioning unit
- System-level tests
  - At the user interaction level

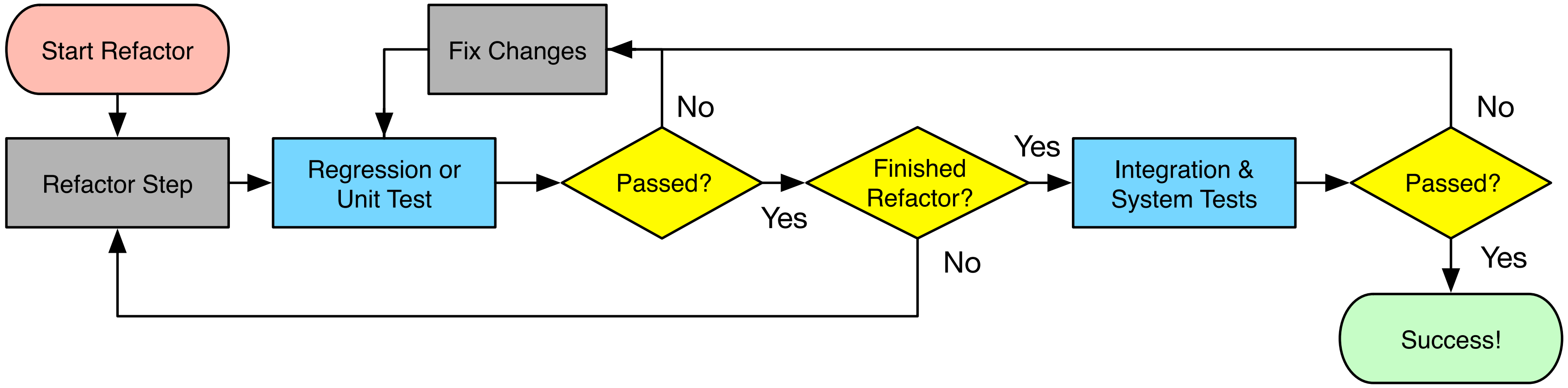
Definitions

Classes of Tests by Intent

- General verification
  - Compare against analytic results, independently-derived result, *etc.*
- Regression tests
  - Compare current observable output to a gold standard
- Performance tests
  - Focus on the runtime and resource utilization
- Restart tests
  - Code starts transparently from a checkpoint

Refactoring

Toy workflow with testing



If using version control, when do you commit?

Automated Test Frameworks

Software packages that

- provide routines that reduce tedium of writing test conditions,
- Simplify maintaining and building test suite,
- automates execution of a collection of client test routines, and
- report test results.

Examples:

- C++
  - Boost.Test, Catch, GoogleTest
- Fortran
  - Fruit, PFunit
- python - nose, pytest, unittest
- Java - JUnit
- MATLAB - built in
- Custom - FlashTest

Test Servers

Servers that
- automate the execution of a test suite or a subset of a test suite,
- allow for running tests on different environments,
- host an interface for viewing results, and
- allows for configuring when the tests are run.

Examples:
- CTest/CDash
- Jenkins
- Travis CI and GitLab CI

Testing Policies

- Testing regime is only useful if it is maintained and monitored
- When to test and what to test?
- When and how to update baseline data and tolerances?
- Must have consistent policy on dealing with failed tests and undetected bugs

Agenda

Time	Topic	Speaker
1:30pm-2:15pm	Why effective software practices are essential for CSE projects	Anshu Dubey, ANL
2:15pm-2:45pm	Better (small) scientific software teams	Michael A. Heroux, SNL
2:45pm-3:00pm	Improving Reproducibility Through Better Software Practices	Michael A. Heroux, SNL
3:00pm-3:30pm	Break	
3:30pm-4:15pm	Testing HPC Scientific Software: Introduction	Jared O'Neal, ANL
4:15pm-4:45pm	Verification, and Evaluating Project Testing Needs	Anshu Dubey, ANL
4:45am-5:00pm	Code Coverage and CI	Jared O'Neal, ANL