Thank you!

Simon Hettrick
Deputy Director
ORCID: 0000-0002-6809-5195

Software Sustainability Institute

Future and feedback

- Use your new knowledge!
- Visit Software Carpentry website for more
- Six-monthly Software Carpentry bootcamps coming soon

- Feedback:

  http://bit.ly/NGCMFeedback

Five things to consider when writing code

1. Get your requirements right

Most errors are introduced during requirements analysis and design

The later they are removed, the more expensive it is to take them out (1/10/100?)

2. Write code assuming others will see it

It's harder to read code than to write it

Include code comments that explain the *why*

Don't optimise unless necessary

- "Premature optimisation is the root of all evil" (D. Knuth, attributed to C. Hoare 1974)

Three rules of optimisation:

1. Don't
2. Don't yet
3. Profile your code before optimising

3. Write code assuming others will run it

Include ‘Gold’ test data

- ‘This is what you should see’
- Ensure tests are well-documented
- Test rationale, what does it give you

Documentation

- In-code documentation e.g. model breakdown rationale by module, by function
- Model documentation – how does implementation relate to the science?

“Half the errors are found in 15% of the modules”
– Davis (1995) quoting Endres (1975)

“About 80% of the defects come from 20% of the modules”
– Boehm and Basili (2001)

4. Get others to look at your code

- Rigorous inspections can remove 60-90% of errors before first test is run (Fagan, 1975)
- A colleague!
- The first review & hour matter most (Cohen, 2006)
- Just explaining your code helps you to better understand your intent – and improve it!

- Focus on critical sections

5. Use version control

- Complete code development history – helps reproducibility
- Avoid ‘dead laptop, lost software’ syndrome
- Collaborative development becomes easy
- What should you version control?
    - Code, documentation, unit tests, test data, custom scripts…

“If you're not using version control, whatever else you might be doing with a computer, it's not science”

– Greg Wilson, Software Carpentry