UNIVERSITY OF BIRMINGHAM

# Good Coding Practices


Andrew Edmondson
Research Software Group
Advanced Research Computing
May 2018

# The Hacker Within

To subscribe to the Hacker Within mailing list please email Majordomo@lists.bham.ac.uk with **subscribe hacker-within** in the main body of the email (please include a new line at the end).

- http://www.thehackerwithin.org/UoB/
- https://intranet.birmingham.ac.uk/thw

# BEAR Software

BEAR Software is a range of services provided to researchers, and research groups, with the aim of improving the research software written and used by the researchers. BEAR Software's mission is summed up in the words of the Software Sustainability Institute:

'*better software, better research*'.

- www.birmingham.ac.uk/bear-software
- bear-software@contacts.bham.ac.uk
- https://bham-rse.slack.com/

# Reproducible Research

It's two years later… and you need to reproduce those results… but your software doesn't produce the same results today…

# Reproducible Research

- **VERSION CONTROL**
  - Data
  - Software
  - Environment
- Does the same version of the software using the same version of the data running in the same environment produce the same results?

# Reproducible Research

- If your software is not under control, then you cannot have confidence in your results. Or hope to reproduce them.

(or hope that anyone else could reproduce them either)

Chaos

# Collaboration


Time has passed, you are now a PI. And you've got some RFs working with you now…

And they need to work on your code.

# Collaboration (1/3)

- Version control (again!)
  - Branching, merging etc.
- Issue tracking
- Testing
  - Unit tests with continuous integration
- Building
  - Straightforward documentation
  - Use standard tools

Untested code

# Collaboration (2/3)

- Comments – not too much, not too little
- Naming conventions
  - Use meaningful names
- Coding standards
  - E.g. PEP8 for Python
  - Static code analysis (while editing if possible)
- Make it readable

Misleading names/comments

# Collaboration (3/3)

- Structure
  - Functions should do one thing
  - Functions should fit on one page
  - Files should contain related things
  - Folders should contain related things
- Simple is better than complex
  - Don’t write code you don’t need

Complexity

# Editing your own code

It’s 5 years later and you need to edit you own code… but you have no idea how it works.

# Editing your own code

- See “Collaboration”

Thinking “I’ll never need to…”

# Design

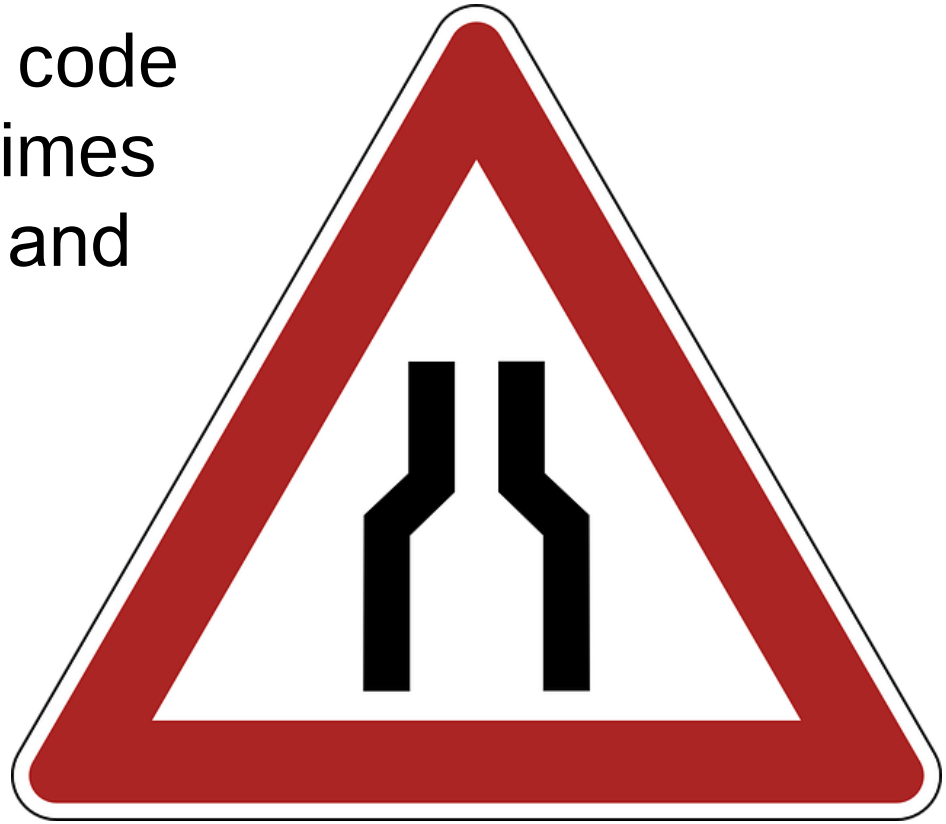
You need to write some new software, but you're not sure what to use…

# Design

- Language / tools
  - Do you use your favourite?
  - Do you use the best for the job?
  - (That's not as simple as it seems)
- Patterns
  - There may be a standard way of doing it
- Review
  - Ask a friend to review the design/code

# Optimisation

You need to run your code with data 1,000,000 times bigger than before… and it's really slow.



# Optimisation

- Profiling
  - Language/system specific
- DRY
- Docstrings
- architecture documentation
- Then optimise the slow, frequently used bits
  - Optimising usually adds complexity

Complexity

# Discussion