

# 389COM: Code Quality

Dr Carey Pridgeon, DR Nazaraf Shah

Coventry University

<http://www.coventry.ac.uk>

2016-08-06 Tue

# Readability

- Style guides exist for a reason.
- Well written, non obscure code is more likely to be re-used.
- Potential employers are more likely to be impressed by well structured code.
- Comments!!!!
- 'I can understand it so other people can too' has *never* been a valid viewpoint.

# Appropriate design choices

- In open source coding, especially at the start of a project, there is no rush.
- Choose the right Language.
- Choose the right programming methodology *OOP*, *Procedural*, *Functional*.
- It matters more that the chosen methodology matches the problem than whether or not it is in vogue.
- Have some design ideas, and make them visible to others, write some User Stories, put these online in your repository.

# Documentation 1

- This is probably the biggest problem in open source development.
- Why is it worse for open source than it is for the commercial world?
- Why should avoiding this problem be a priority for you?
- Automatic Documentation

# Documentation 2

- Wiki pages (gitlab/github et al provide these)
- Avoid static documentation formats (hand coded web pages)
- Documentation can be auto-generated from your source, provided you provide rich comments in the right format  
[Doxygen](#)

# Testing

- What do we mean by testing code.
- When do you start testing.
- What testing approaches are there?
- Black Box testing.
- Edge cases determination.
- Unit testing.

# Bug Tracking/Fixing 1

- If you don't use an online git repository, your code will essentially be invisible.
- 'Many Eyes' Principle is only true in theory.
- All Bugs need to be reported in your online documentation gitlab/github has 'issues' functionality.
- All work to fix them, successful or not, must be reported in 'issues', because of the potential employer visibility aspect.

## Bug Tracking/Fixing 2

- Bugs aren't really significant externally till you get to tentative release level of development, although for shared development.
- Non trivial bugs still go in issues.
- For shared development, planning and timeline management should also go online
- [Trello](#) is a good place to do this.



# Obligatory XKCD

KEEP IN MIND THAT I'M  
SELF-TAUGHT, SO MY CODE  
MAY BE A LITTLE MESSY.

LEMME SEE-  
I'M SURE  
IT'S FINE.



...WOW.

THIS IS LIKE BEING IN  
A HOUSE BUILT BY A  
CHILD USING NOTHING  
BUT A HATCHET AND A  
PICTURE OF A HOUSE.



IT'S LIKE A SALAD RECIPE  
WRITTEN BY A CORPORATE

IT'S LIKE SOMEONE TOOK A  
TRANSCRIPT OF A COUPLE