DEPARTMENT OF COMPUTER & INFORMATION SCIENCES

University of Strathclyde Science

Software Engineering

Lecture 10: Quality

Plus Other Stuff

Billy Wallace
w.wallace@strath.ac.uk

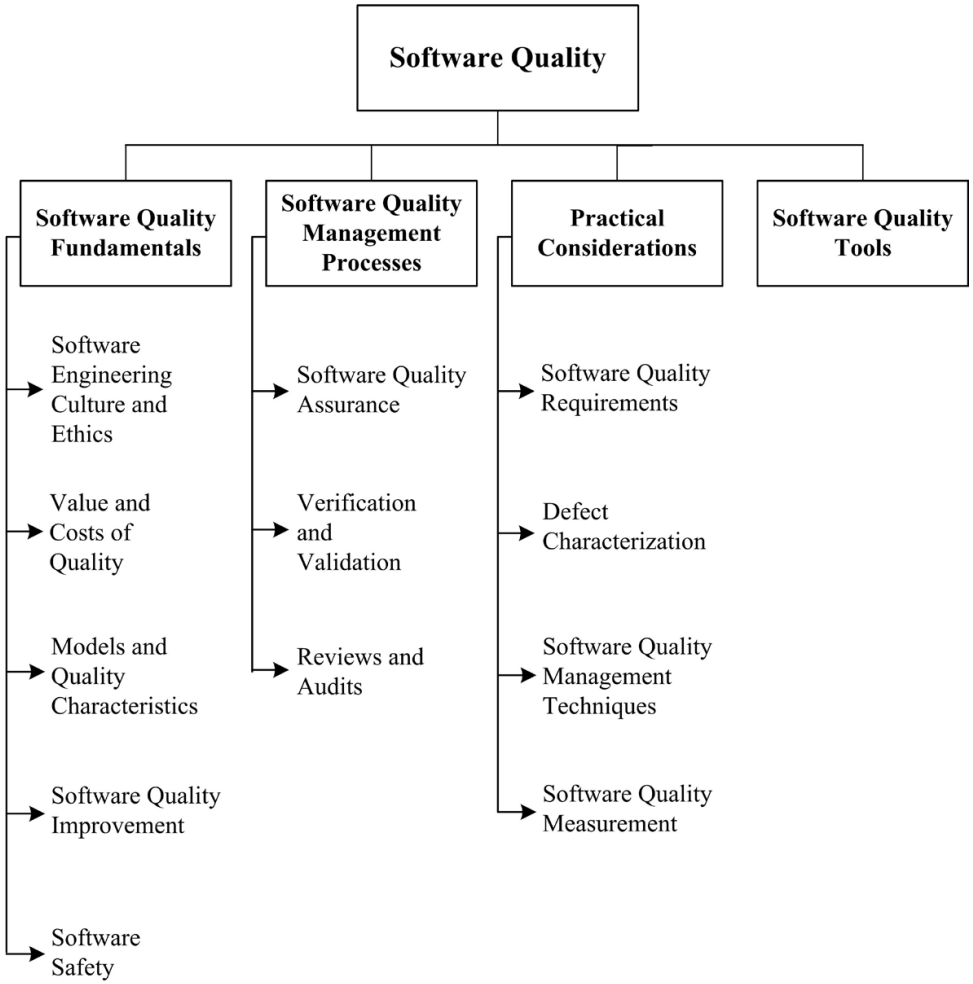
CS993

Lecture Outline

- What is quality?
- How do we encourage it?

- Let's talk about the reports

Quality


Software Quality
Software Quality Fundamentals
Software Quality Management Processes
Practical Considerations
Software Quality Tools
Software Engineering Culture and Ethics
Value and Costs of Quality
Models and Quality Characteristics
Software Quality Improvement
Software Safety
Software Quality Assurance
Verification and Validation
Reviews and Audits
Software Quality Requirements
Defect Characterization
Software Quality Management Techniques
Software Quality Measurement

What is Quality?

- Could be several things
  - Does our software meet requirements?
  - Is our software buggy?
  - Is it anything to do with ISO9000?
  - Is it more intangible?
  - Does it solve the problem?

Things we've already seen

- Lifecycle model
- Processes
- Testing
  - Verification
  - Validation
- Code reviews
- Etc

If you can't measure it …

- We already spoke about:
  - Velocity
  - Numbers of bug reports
  - Etc
- We can also use metrics to measure "quality"
  - Simple things like length of methods can point us to things to improve.
  - Averages of such measures can be used as metrics.
  - NB the lab last week was supposed to show you how you might implement such a thing.
- Not a metric, but tools like FindBugs can help you identify areas for improvement.
- Code reviews
- Etc

Culture & Ethics

- People like to work on things that are “good”.
- This includes a good quality product, but also things that do good.
- As well as being the right thing to do, building a culture like this makes people more productive.
- If you ever have influence over culture at an organisation or just a software team, strive to be good.

Individual Reports

- You won't know what your project will look like yet, so how do you write a report on it?
- Basically do as much as you can, but if there are things you don't know yet, make assumptions.
- Let's say you're going to write a mobile app and don't know yet whether it's iOS or Android. Pick one and write about what you'll need to do for that.
- It would be good to be able to show a wee bit of requirements capture, but the best you might be able to do is some background reading in the general area.
- Try to get to the point where you have an architecture in mind and write about a process and tools that will support that.

Summary

- Quality is what we're aiming for, but it's hard to define.
- For your individual report, you might need to make assumptions and write about things based on your assumptions.

Moving on from here …

- The final lab is a bit different. It takes a new programming language (Rust) and gets you to look at the tools to support the software lifecycle in that language.

University of Strathclyde Glasgow