Software Quality

It doesn’t happen by accident.
Dimensions of Software Quality

- Software quality is measured in many ways:
  - Meets requirements/specifications
  - Absence of defects ("bugs")
  - Maintainability
  - Craftsmanship
  - Performance
  - Security
  - Size

- Objective and subjective measures
  - Code analysis vs code review
  - Automated testing vs demo
Many activities contribute to software quality:
  - Software engineering process
  - Architecture and design
  - Coding standards and practices
  - Version control
  - Testing
  - Code reviews

Quality does not happen by accident!
Agile ≠ Quality

- Agile provides a *process* framework
  - Individual activities largely not addressed
  - Customize for your quality standards

- Activities you (hopefully) already do:
  - Architecture and design
  - Good coding practices
  - Version control
  - Unit testing (one kind of testing)
  - Pair programming (sort-of continuous code review)

- Is it enough?
  - How will you know when your product is high quality?
Having a Plan

• For this course:
  • How will you validate that you have met the requirements?
  • How will you address project risks?
  • Will your code be high quality? maintainable?
  • Do you need to address performance, security, etc.?

• Make a plan
  • List activities you will do
  • Tie each activity to a quality aspect it addresses
  • E.g.,
    • Unit testing: verify basic functionality, provide repeatable detection of previously seen defects
“If you don’t like testing your product, most likely your customers won’t like to test it either.”

- Anonymous
(Some) Types of Testing

- Unit testing
  - These should mostly be developed in TDD
  - Adding after the fact is a pain and hard to get right
- Functional testing
  - Testing requirements/user stories
  - Generally requires interaction with UI
- Build, integration, deployment testing
  - Do all the parts work together?
  - Does it work in production?
- Load testing
- Security testing

There are tools for automating all of these!
What to Test

• User stories
  • The expected use cases

• Edge cases
  • What if the numbers are really big?

• Invalid inputs
  • Users can break anything...
Final Thoughts
“The bitterness of poor quality remains long after the sweetness of meeting the schedule has been forgotten.”

- Karl Wiegers