## Today
- Exam Overview
- Testing (cont)

## Homework
- Final (Signed) Plan & Feature Checklist due next Wed
- HW8 out (please read carefully)

## Announcements
- Exam 1 Tues
Exam 1 Overview

Basics

- Closed book, notes, etc.
- Worth 10% of final grade
- Mix of questions (short answer, terminology, etc)
- Everything we’ve discussed and readings are fair game
- Up through estimation, planning, sprints (but not testing)

Possible Topics:

- **Basics**: Project, Product, Process, Scope, Budget, Time, Quality
- **Requirements**: Function, Feature, Non-Functional, Business Objectives
- **Development Processes**: Code-and-fix, Waterfall, Spiral, Iterative, Incremental
- **Agile/Scrum Approaches**: User stories (epics, themes, stories, spikes), Backlogs, Sprints (and meetings: planning, review, retrospectives), Task Boards
- **Planning**: Estimates, Targets, Commitments, Estimation approaches and metrics, Velocity, Milestones, Release Planning
- **Design Tools**: Context, Architecture, UI Sketches/Storyboards
Testing Terminology: Basics

Verification: Are you building it correctly (to spec)?

Validation: Are you building the correct thing (user)?

Independent: Testing by a “third party”
- E.g., independent verification (e.g., system testing by 3rd party)
- And independent validation (e.g., usability testing for “ease of use”)

White Box: Test with knowledge of the code
- Tests based on how system design/coded (e.g., unit and integration tests)

Black Box: Test without knowledge of the code
- Tests based on input/output (e.g., system tests)
- How user interacts with or “sees” the system

Gray Box: Mix of both
- Testing connections to subsystems
- Performance testing another example
Small Test: Usually automated, single function or module test
- functional issues like error conditions, data corruption, off-by-one mistakes
- Answer question: “Does this code do what it is supposed to do?”

Medium Test: Usually automated, two or more interacting features
- Answer question: “Does a set of functions interoperate with each other the way they are supposed to?”

Large Test: Three or more features and represent real user scenarios
- Check that software satisfies user needs
- Answer question: “Does the product operator the way a user would expect and produce the desired results?”

Enormous Test: End-to-end tests
- “Whole” system tests from UI to backend storage
Terminology: Manifestations

Mistake: incorrect human action (select bad loop condition)

Fault: the manifestation of the mistake (coded condition)

Failure: the result of the fault (wrong value calculated)

Error: expected vs unexpected (off by 20%)

Bug: usually same as fault ... sometimes all of these!

Defect: usually same as fault
Terminology: Metrics

LOC: lines of code (issues?)

Defect rate: defects found / hour

Defect density: defect found / LOC

Inspection rate: LOC / hour