Reminder

• I drop the lowest mini-exam mark
• If you wrote the first three, and you’re satisfied with your average, you don’t have to show up
Likely topics/questions on mini-exam #3

• From Lecture 20 (formal inspections)
  – Know (roughly) the relative costs from “Cost of Fixing Errors” (Lecture 20 slide 26)
  – Know the roles of Moderator, Inspectors, Author:
    • I could ask you to match them up with “keeps paraphrasing at a reasonable pace”, “paraphrase the code”, “clarifies code when asked”, etc.
    • I could describe a flawed Bogosys inspection process and ask you what’s wrong with it
Likely topics/questions on mini-exam #3

• From Lecture 21 (formal inspections)
  – Know (roughly) the defect classification (Critical, Severe, Moderate, Minor)
Likely topics/questions on mini-exam #3

• From Lecture 22 (code inspections)
  – Know some examples of items on a code checklist
Likely topics/questions on mini-exam #3

• From Lecture 19–0 (XP code inspection, refactoring)
  – Know some examples of “code smells” and refactorings, especially “Don’t Repeat Yourself” and how to “factor out” duplicate code into a new method
Likely topics/questions on mini-exam #4

• From Lecture 25:
  – Basics of measurement
Likely topics/questions on mini-exam #4

• From Lecture 26:
  – difference between faults and failures
  – defect density
    • number of defects found, not the “real” number of defects (which is beyond our mortal knowledge)
    • relationship between faults and failures
  – software size: what’s wrong with using (S)LOC?
Likely topics/questions on mini-exam #4

• From Lecture 27:

  – COCOMO = Constructive Cost Model
  – Effort = $a \times (\text{Size})^b$

  – almost meaningless if Size is in LOC

  – different measure of size: function points

    • you don’t need to memorize specific elements (“logical master files”, etc.) or the coefficients on slide 19, but you should understand the basic idea and why function points are a better measure than LOC
Bonus question

• involves Bogosys