CISC327 - Software Quality Assurance

Lecture 9a

Review for Mini-Exam #1
Announcements

• We have 4 TAs
• Assignment #0 handin should be open again
• If you need exam accommodations, please wait until tomorrow (Thursday) morning to contact me
• Today: More about testing (not on exam), review of exam material
Likely topics on mini-exam

• From Lecture 1:
  – Quality parameters: **technical** and **user**
    • correctness, reliability, capability, ...
    • usability, user documentation, availability, ...
  – The 3 general principles of quality assurance
  – Formal methods, testing, inspection, metrics
  – Know what a safety-critical system is
Likely topics on mini-exam

• From Lecture 2: 4 fundamental process activities
  – If it helps, remember “SDEV”, like “Software DEVELOper” (but also remember that the order is actually S, D, V, E)
  – Think about **why** the order is S, D, V, E. Say you do remember “Development”. Development needs some kind of specification. So now you have “Specification; Development”. Who cares about the software? The customer ⇒ Validation. Software has version numbers ⇒ Evolution.
  – You should be able to briefly describe what happens in one stage, not just the 4 words in that order
Likely topics on mini-exam

• From Lecture 2: Process models
  – Waterfall model
    • Drawback: “Early freezing” at various stages. Example: Freezing requirements risks doing lots of work against the wrong requirements.
    • Could work well when requirements are very stable
  – Prototyping model
    • Do a “quick and dirty hack” prototype, poor quality in most respects—**but** it gives the customer something to try out, identifying issues with requirements.
    • Drawback: wastes work—the prototype gets thrown away
Likely topics on mini-exam

• From Lecture 3:
  – Spiral model:
    • Continuously document and evaluate potential risks
    • Needs an experienced team
  – Iterative development:
    • Develop a subset of the software based on a subset of the requirements
    • Develop bigger subsets
    • Architecture is chosen early & needs a small team
Likely topics on mini-exam

• From Lecture 3 (continued):
  – Object-oriented development process
    • Analyze “essential system”, define “essential classes”
    • Derive additional classes
    • Define interfaces
    • Complete design and implement classes
    • Drawbacks: no testing until late, architectural inflexibility (maybe the essential classes aren’t complete, maybe there are issues with requirements)
Likely topics on mini-exam

• From Lecture 4:
  – Defects Prevention Process
    • You should have a rough idea of the steps involved
  – {CMM, SPR, Baldrige, ISO...} ←THESE WILL NOT BE ON THE MINI-EXAM
Likely topics on mini-exam

• From Lecture 5: XP
  – Know the XP practices, especially:
    • simplicity
    • test first, test all the time
    • pair programming
    • on-site customer
  – Know some drawbacks of XP
  – Would XP be suitable for safety-critical software? (this is a “deeper” question than I would ask on an exam, but thinking about it may solidify your knowledge)
Rest of XP

• The chapters in the coursepack on XP are pretty readable, and will help solidify what you get from the slides

• Think about XP and how it might apply to the course project so far