CS 430 MidTerm Practice Problems on Text Ch 6

Not for handing in - Solutions will be posted on Friday, October 24

1. State the similarities and differences between the two review processes: **walkthroughs** and **inspections**.
   Solution:
   Similarities:
   - Both are chaired by an SQA professional.
   - Both are review processes performed by teams with the aim of finding faults.
   - In both cases, the material is produced by individual team members, and then the material is reviewed by the team as a group.

   Differences:
   - An inspection is a formal five-step process whereas a walkthrough has only two informal steps.
   - Previously acquired fault statistics play an important role in the inspection process, not in the walkthrough process.
   - There is a formal component of the inspection process for ensuring that all faults noted are later corrected. In a walkthrough faults are only found, not fixed.
   - If more than a certain proportion of the material is changed then it must be submitted for re-inspection.
2. In class, we discussed five things that should be tested for a software product. List the five things, and provide one example for each.
   Solution: The five things to test are
   (a) **Utility** ✓
   Examples (any of these):
      i. Is the S/W product easy to use?
      ii. Does the S/W product perform useful functions?
      iii. Is the S/W product cost effective?

   (b) **Reliability** ✓
   Examples (any of these):
      i. **mean time between failures** Long times → more reliable.
      ii. **mean time to repair failures** Long times to fix → less reliable.
         A. **Also important (often overlooked):** time required to fix the **effects** of the failure. Long times to fix → less reliable.

   (c) **Robustness** ✓
   Examples (any of these):
      i. range of operating conditions (permissible by the specifications or not)
         A. A robust product has a wide range of operating conditions, including some outside the specifications.
      ii. possibility of unacceptable results given acceptable input
         A. A robust product produces acceptable results given acceptable input.
      iii. acceptability of results given unacceptable input
         A. A robust product produces acceptable results (e.g. a helpful error message

instead of a crash) even given unacceptable input.

(d) **Performance** ✓
Examples (any of these):
   i. It is crucial to verify that a S/W product meets its constraints with respect to response times or time/space requirements.
   ii. Space constraints can be critical in miniature applications, e.g.
      A. missile guidance systems as in the text, or
      B. smart phone apps.
   iii. Time constraints can be critical in **real time** applications, e.g.
      A. measuring core temperature in a nuclear reactor as in the text, or
      B. controlling signals on a railroad network.

(e) **Correctness** ✓
Example: Does the S/W product satisfy its output specifications, without regard for the computing resources used, when operated under permissible (pre-)conditions?

3. In class, we discussed one particular situation in which correctness proving can be justified via cost-benefit analysis. What was it?
Solution: In class, we discussed the fact that it is easy to justify correctness proving via cost-benefit analysis, when human lives depend on software correctness (e.g. the software to control autonomous vehicles).