Software Quality Assurance: SOFT3302

Tutorial — Week 11

Objectives

A retrospective. Revision.

Prework

Make sure you are not leaving Assignment 3 until the last minute. Begin organising your revision of the course in preparation for the exam.

Labwork

Consider some of the questions from previous weeks' tutorials. Discuss them in small groups. Have any of your answers/attitudes changed during the course of the semester?

1. *Consider the statement 'A tester is not a failed programmer.' Does this fit with your understanding of a tester?* (Week 2, Question 4)

   Has an understanding of the structure and complexity necessary to ensure effective testing changed/reinforced your answer to this question?

   How has your understanding of the nature of a tester and the skills required to be a tester changed?

   If students thought that testers needed fewer skills and/or less comprehension than developers then hopefully they will have been disabused of this by now. If they already suspected/realised the skills necessary in a good tester, hopefully delving into the topics we have covered this semester will have reinforced those ideas.

2. *Discuss any perceived advantages and disadvantages of unstructured* versus *structured testing.* (Week 3, Questions 2 & 8)

   Opinion should come down firmly on the side of structured testing for this one. Make sure it is stressed that exploratory/*ad hoc* testing is not an effective test technique as there is no structure or repeatability.

3. *Do you think that testing needs to be taken into account during design? What does this say about the temporal ordering of testing and coding?* (Week 3, Question 7)

   Discuss this in the light of what you have learnt this semester and also in Assignment 2.

   Clearly assignment 2 shows that skipping testing or leaving testing until late in a project drives up the cost of defect removal dramatically. QA activities need to take place during the early design phases (comment on IBM's 'cleanroom' approach where the aim is to not let defects enter the process in the first place). We will be looking at review activities this week.

Use any remaining time to review last week's lecture (remember, it is examinable) and, indeed, the course so far since we are drawing close to the end of the semester. Ask questions of your tutor and discover areas in which you need to perform revision.

Actually make sure students do this. Encourage them not to just walk out of the tutorial thinking they get 'a week off.' Try and engage them in discussion.

Comments on previous weeks' tutorials

I noticed and was told of many students whose attitude seems to have been 'yeah, yeah, this is obvious' and either did not do the tutorial or left early; similarly with the previous tutorial on test procedures.

As with quite a few things in computer science and mathematics thinking one understands something and actually doing it so one *knows* one understands it may be two entirely different things.

For example, many students verbally communicated during the test procedures in the Week 9 tutorial—the procedure author 'looking over the shoulder' of the tester and correcting his actions—even though this was expressly forbidden by the meta-procedure. Doing this negates one of the major points of the exercise: writing clear, succinct and comprehensive procedures is *hard*, things get left out and knowledge gets assumed.

Similarly, in last week's tutorial, performing the risk analysis and ranking the risks, while time-consuming, is a necessary part of preparing a test strategy and is good revision to make sure you have understood and can bring together many of the things covered in the course.

If you have skimped with your efforts in the last two tutorials, take the time to rectify this situation this week and revisit those tutorials.

Dr James Farrow

Stress the two points above: 'doing' may lead to the uncovering of unwarranted assumptions of knowledge.