Software Testing Tutorial LO 1

There are *five* tutorial topics scheduled for the Software Testing course. Each of these will be discussed over two weeks. In the first week the group will work on a preparatory task without a tutor and in the second the group will work with the tutor. The goal is to use the software you are working with to exemplify what each section of the portfolio should look like.

Recall LO 1 is "*Analyze requirements to determine appropriate testing strategies*" so this section of your portfolio should provide a range of requirements and some analysis of what might be necessary adequately to test those requirements. You may develop a longer list of requirements, but the portfolio should mention a small number that are distinctively different from one another.

Preparation session, week 2 (25 Sept-29 Sept)

Usually, the preparation sessions will not have a tutor present but since this is the first prep meeting the Course Organiser will be present at all tutorials in this week.

A good preparation for this activity is to read Chapters 2 and 3 of Y&P.

- Before the meeting, write a short description of the software you intend to work on and consider:
    - Who are the stakeholders in your software (remember that as well as the primary users they may be other stakeholders who have different perspectives on the software, e.g. an app may have primary users but there are also other potential users e.g. people who deliver the service ordered on the app)
    - From each of the stakeholders' perspectives, identify a small number (1-3) requirements that are important to those stakeholders. These can be quite informal, stated in natural language and imprecise in some ways. Try to have a diversity of different kinds of requirement.
    - In thinking about diversity of requirements, recall:
        - Some requirements are *functional* in that they specify some characteristic of the activity of the system e.g.
            - Safety properties say the system can't do some things.
            - Correctness properties say that the result is correct.
            - Liveness properties say that the system always returns to some recognised "ready" state.
            - Security properties
        - Some requirements make statement about a measurable quality attribute of the system e.g.
            - Statements about resource use (memory, compute time, …)
            - Statements about an "-ility" reliability, usability, availability, …
            - Statements about timing e.g. response times
    - Try to ensure you have a good range.

- At the meeting together with your group consider two or three example requirements contributed by the group. These examples should also be diverse (e.g. three functional safety requirements is not very diverse). Then, attempt to do the following:
    - Make the statement of the requirement more precise without significant loss of the meaning of the requirement.
    - Work out how you think it might be possible to test a system to see it satisfies the requirement.
    - Discuss the extent to which you think you proposed test really guarantees the system meets the requirement. Are there strengths and weaknesses in your proposed tests?
    - Write down the more precise versions of the requirement and your proposed means of testing.
- After the meeting consider your software and select a small, diverse, collection of requirements and try to make them more precise and consider how you might go about testing for these requirements

Tutored session, week 3 (2 Oct-6 Oct)

In this session you will consider some of the collections of requirements and proposed tests with the tutor and consider their strengths and weaknesses. The goal is to consider what would be a good section on LO 1 for your portfolio. This will involve assessing potential sections against the grading scheme. To do this we will consider the marking scheme for the portfolio because that indicates what emphasis is expected in the portfolio. Here are the four criteria LO 1 is graded on (recall each is graded in the range 0-5) with the interpretation provided in the Marking Scheme:

1.1. *Range of requirements, functional requirements, measurable quality attributes, qualitative requirements, …* At this point the focus is on the identification of system level requirements. The emphasis is on the range of different types of requirement so in the tutorial we should consider what would make a good combination of requirements to point to in the portfolio. The evidence to support this criterion is a list structured by types of requirement that good coverage of different types of requirement. This should be a short document stored in your GitLab project that this section of the portfolio can refer to.

1.2. *Level of requirements, system, integration, unit.* At this stage it may only be possible to refer to system level requirements because you have yet to consider an implementation where it will be necessary to develop unit and integration tests. It is possible to indicate the sorts of unit and integration tests that will be likely to be necessary and these can be refined once you have started developing some code. For some systems there are clear sub-systems that must appear in any implementation and for such systems you will be able to discuss requirements for sub systems and how they integrate.

1.3. *Identifying test approach for chosen attributes.* As the course progresses you will be able to elaborate this section somewhat but even at this early stage you can develop some broad statements about what approaches to test would be appropriate and which are not appropriate.

1.4. *Assess the appropriateness of your chosen testing approach.* At this early stage you should be able to point out some of the limitations of your approach to testing some particular requirement. For example, it may be that lack of data makes it difficult to do performance testing under realistic loads.