Assignment 2

Writing high-level test cases and designing the system

Now, when you have described the requirements, it is high time to write test cases for them, estimate them and plan for their implementation. In this assignment, you will do the following:

- Write test cases for your requirements using the template in Figure 1. Please keep in mind that not always all these fields need to be filled in.
  - The template may be used for testing requirements, code, user manuals and other.
  - It specifies the fields that are important for running tests.
  - All test cases must have a unique identifier (Test Case ID).
  - With one test case, you test at least one requirement/class/function/module/method. You may also test several integrated requirements/classes/functions/modules/methods. At this moment, what you test is only requirements. So, your modules will be either one requirement or a set of them.
  - All that is tested is tested against some input and output. Please specify the input and the expected output with the input provided.
  - Some requirements must be tested in special environments, such as test labs, special hardware or software platform and the like. Therefore, you must specify environmental needs for testing those requirements.
  - Sometimes you have to run several tests in sequence. This must be specified. For instance, you may specify that test case 1 is to be run after test case 2.
  - A sequence of tests required corresponds to testing procedures. In addition, you might have special procedural needs for executing the test cases. For instance, you may have to execute tests 1, 2 and 3 once and then iterate tests 4,5, 6 several times, and finally, to test 7 only ones. This applies more when you have code.

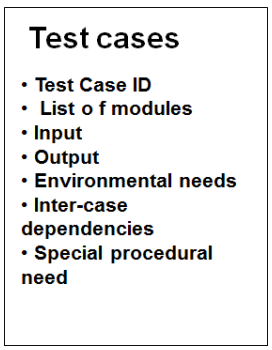

Figure 1. A test case template

- When formulating the test cases, you may realize that your description of requirements is not easily testable. Probably, your requirements are not clearly described. It is ok then to rewrite

requirements so that they match the test cases and vice versa. Make sure that you document the changes and deliver them together with the results of this assignment.

- After having created requirements and test cases, estimate the effort required for implementing and testing it. Do it in the following sequence:
  - **Step 1:** Make the estimation of the requirement described in an agile way. Do it the following:
    - Play the poker game, the same game as demonstrated on Lecture 3. The only difference is that you will use real time (in person hours) instead of story points. Create your own poker cards.
    - Document the effort to implement the requirement.
  - **Step 2:** Do the estimation of the requirement described in a traditional way, do it in the following way:
    - Divide the requirement into low-level requirements. Do it till you come to the lowest requirement level.
    - Estimate the requirements size in LOC (lines of code).
      - I suggest that you consider all the lines, blank and non-blank lines.
    - Estimate the effort using person-hours.
    - Document the effort to implement the requirements.
    - You probably wonder why you have to guess the size of your implemented requirements in LOC. You probably feel that it is like shooting yourself in the foot. Well, that is what may industries do. What they usually have is a historical database showing the size of similar systems. But many times they will have to guess. So, you will have to guess here as well. It is difficult, isn't it.
  - Compare the two approaches: traditional versus agile.
- Design your system using the design method that you feel comfortable with.
  - Out of what I have heard, some students attending this course have not gone any design course. Therefore, you use whatever design method you feel comfortable with. In its simplest, you may have boxes representing functional units and arrows representing communication between the units.
- Go through the states of the Software System alpha and identify its status.
- Go through the states of the Team alpha and identify its status by evaluating the status of your team.
- Also, check the progress of the status states of the *Requirements, Requirements Item* and *Opportunity* alphas. Have they progressed?
- Document the results of Assignment 2 and merge it with the results of Assignment 1. Wait with delivery till you are done with all the assignments.