CSL 740: Software Engineering
Theory Assignment No 1
Chapter 1

Date of submission : 4th February 2020
Maximum marks:100

Date uploaded :30th December 2019

Instructions:

- **Be brief and precise in your answers.**
- This assignment is to be submitted in the form of a .Tex source and the associated .pdf - both archived to a single folder by the name "YourIDNo-Tut1" and uploaded to the LMS.
- The assignment would be graded not only based on what you have written and uploaded but based on what you answer and present during the flip mode classroom discussions on this topic in the class.
- The total marks of this assignment are 100, including those for flip mode class discussions.

1. *Information determinacy refers to the predictability of the order and timing of information. An engineering analysis program accepts data that have a predefined order, executes the analysis algorithm(s) without interruption, and produces resultant data in report or graphical format. Such applications are determinate. A multi-user operating system, on the other hand, accepts inputs that have varied content and arbitrary timing, executes algorithms that can be interrupted by external conditions, and produces output that varies as a function of environment and time. Applications with these characteristics are indeterminate.* [Src: Stephen Schach].

   (a) Use information determinacy and information content as the two parameters to classify the software. Enlist various types of software based on this.

   (b) In addition, note that Software can be classified primarily as the *System software* and the *Application Software*. The *Application Software* can further be classified into the following:

   - Based on property and use rights e.g. open source or otherwise
   - Based on programming language e.g. .c, .cpp etc.
   - Based on output and purpose. These can also be either horizontal or vertical. General-purpose applications are more popular and widespread, in the category of horizontal just like word processors or databases. Vertical applications designed for a particular type of industry or business, or department within an organization. For example, a software suite is a group of software applications with related functionality such as Railways Reservation Software.

   In this context, briefly explain how *information content* and *information determinacy* can help in classifying software as it is in the above.

2. Download Brooks *No Silver Bullet* paper as referred to in the class and comment on whether does it represent a pessimistic or optimistic outlook of the Software Engineering. Why ?

3. This question is again on Brooks *No Silver Bullet* paper. Give an example each to justify each the four attributes viz. *conformity, changeability, complexity and invisibility* due to which Brook's argued that there are inherent issues with the way we develop the software.

4. In the class we discussed a few of the real life catastrophes that are attributed to some fault in the software. Now, it is your turn to research the common failures/catastrophes in different fields (e.g. space expeditions, wars, industrial process control disasters etc.) as well as the others on the web, identify and find out precisely what software fault caused these missions to fail.

5. Compare an Industrial Strength Software (ISS) vis-a-vis a Student Software development effort in terms of whether phased development is pursued, whether testing in every phase is done or not, what is the percentage of the testing (30% to 50% of the total effort). Assume that the ISS costs 10 times more than the student software. Suppose a program for solving a problem costs C and an industrial strength software for solving the same problem costs 10C. Estimate where this extra cost 9C is spent ? Suggest a possible breakdown of this extra cost.

6. Enlist and briefly discuss two main causes for the complexity of the software.

7. Enlist and briefly discuss different types of activities done in the maintenance phase.

8. Suppose the cost of fixing a fault detected in a particular phase A of SDLC is x units. Then, the same for the faults detected in phase B of SDLC is 6.5x, the same for the faults detected in phase C is 15x and that for the faults detected in phase D is 100x. If the phases A, B, C, D are drawn from the domain Design, Implementation, Testing, Maintenance, identify what exactly are A, B, C, D of these four phases.

9. Enlist specific reason(s) as to why the cost of fixing the faults increase steeply with the progression of each phase in conventional SDLC?

10. Suppose one wishes to keep the overall costs of maintaining the software low to the extent possible. Suggest some of the measures to do so.