At the backdrop of Ch.1 pre-read, we simulated an in-class activity to experience the nuances of SDLC concept. At a broad level, SDLC can be viewed as a 5-step developmental process, with iterations among the phases of Planning(What?), Analysis(How?), Design, Implementation and Maintenance. It typically originates as an Enterprise Idea trigger (typically at strategic/CxOs level) and flows down to Conceptual Design (Technology Agnostic at Analysts level), the Logical & physical level (DBMS logic, technology platform), the implementation and to the validation level (Expected vs Actual check). While it seemed to be simple and structured process theoretically, the in-class simulation of the SDLC cycle brought up the potential intricacies in the actual implementation of the same.

As an observer, the following are my findings:

Initially, the client and the analyst tried to agree on what the picture looked like. Confusion prevailed as to whether the picture resembled a "Pyramid" or "Prism". During the entire process, "Client Feedback" on the prototype was completely ignored without realizing that it was indeed very much a possibility. In fact, the pre-read does mention about RAD (Rapid application Development) as an agile approach which emphasises upon the multiple iterations, more frequent validations with clients in terms of progressive prototypes at each iteration of analysis and design building. In today's, ever-changing, dynamic project-development environment, embracing such a "prototyping" kind of approach emerges to be a clear choice over the conventional SDLC. Further, the "people" aspects like behavioural traits and personal aspirations within team too came into picture. This said, in the simulation, the "whole" team (and not only the analyst) was in direct connect with client which may not be a feasible in real scenario. Also, time management was not addressed well. While, the time appeared well enough given the scope of work, we ended up rushing towards the last 5 mins (with QA guy not even getting a minute to validate!).

In conclusion, what seemed to be a simple, smooth workflow theoretically, the SDLC actually emerged out to be a fairly complex one - demanding a well guided, persistent efforts throughout the actual execution process!