Q1. Based on your own knowledge of some of the application types discussed in class (lecture 1), explain, with examples, why different application types require specialized software engineering techniques to support their design and development.

Ans :

We are living in an age where technology is just growing rapidly day by day and we have such a large variety of applications which do different work. We have so many different types of systems like Embedded Systems, Information Systems, Weather Systems, Sensor Based Connection Systems, School System, Health Care Management Systems and Weather Systems. Each of these different systems enlisted above have different function to be performed by them. A School System will generally have all the information of all the students, faculty, support staff, parents information, trustees of the school and a lot of other things which will be useful to run a School. In the same way all the systems listed above have different functions which are performed by them for the user.

As each application type systems have a particular function that they need to perform each of them will require specialized software engineering techniques to support their design and development. The following are the reasons why they require specialized software engineering techniques:

1. Costs and Frequency of Change:
   There are different types of software models such as the Waterfall Model, Incremental Model and Re-Used Model. So every application has different type of software model depending upon the functionality of the system.

For Example: An Employee Management System is a system which can be developed using Waterfall Model in which the customer tells the developer all the requirements and the kind of software that they are expecting. Following the waterfall model it's a one-time investment and the customer gets the desired software in one go and then there are not any major changes in the software.

Similarly any Mobile Game can be developed using Incremental Model where the developers make a prototype send it out for testing and incorporate the different feedback received from the testing and then these changes are included into the game and we get a new version with improvements. This continues even after the game has been launched for the public. As more number of people engage in the game, the developers understand things which people would like to see in the game and then they try to incorporate those changes in our game. For Example : The game of Fruit Ninja which allowed a single player to play at a time followed by Multiplayer on the same device and extending to online multiplayer mode. These different changes were not included overnight, as per the requirements these changes were introduced tested and then incorporated into the game.

2. Different Non Functional Requirements:

   As described above we have so many different types of software systems and each of these systems have different nonfunctional requirements.

   For Example the Bank of America Mobile Application has the top priority to keep the user data safe and ensure that all the information of the user is safe and doesn't not fall into the wrong hands. So the top priority is to ensure safety of user information. This can be seen in the BOFA app as it logs the user out of the system after two minutes of inactivity and thus keeps all the user information safe. So the top priority of this application is Safety of user Information primarily.

   Similarly in FIFA responsiveness and usability is the top priority. The user should be able to use all the feature present in the game and the game should be responsive to the commands given by the user. Following this the extensive UI design, graphics, are the next in line priorities for gaming software.

   As we can see above the nonfunctional requirements for the BOFA mobile app and FIFA are very different as safety of user information is not the top priority in a software game and extensive UI design in not a top priority for the BOFA mobile application.

3. Lifetime of Software and Delivery Scheduling

There are some software which have a short life time where as there are software systems which have lifetime of tens of years. There are certain software which have to be delivered quickly only then they would be of any use. The techniques used for developing short lifetime systems are obviously inappropriate for developing long lifetime systems.

Q2. Discuss whether professional engineers should be certified in the same way as doctors or lawyers.

Ans :

Professional Doctors are required to give Licensing exams and clear the exam to get a certification and he/she must be licensed by the medical licensing board in the state where they plan to practice. Only after the doctors receive this license are they allowed to practice in a particular state. This way doctors are certified for practicing medicine in any particular state.

Similarly Lawyers are required to clear the Bar Exam in order to get into the Bar and get a license to practice Law in a particular state.

In my opinion even professional engineers should be certified in the same way as doctors and lawyers. As the job of an engineer requires a lot of skills and the ability to apply concepts learned in academics in real life situations in order to solve problems in an efficient and optimized manner. It is not fair that doctors and lawyers have to be certified to practice their skillset and engineers are able to do the same without the need of any certification.

Q3. Explain why incremental development is the most effective approach for developing business software systems. Why is this model less appropriate for real-time systems?

Ans:

Incremental Development is the most effective for developing business software systems as the requirements of a business system change as per the requirement of the business. So a traditional plan driven approach will not be helpful because we will not be able to make changes which will be required in the future. In incremental development in each iteration there will be updates as per the new updated requirements of the business. So the incremental development is the most effective approach for business software system.

Real Time Systems are systems which are having hardware components which are cannot be updated in each increment and thus cannot be incremental. Also real time systems have to ensure that they are safety critical. This cannot be done in a number of increments and requires a well-defined plan so that the hardware components are setup in a proper manner and they are safe. So incremental model is not appropriate for real time systems.

Therefore incremental model is appropriate for a business software model and it is less appropriate for a real time system.