

NAME = AHMAD ULLAH KHAN

ID = 6958

SUBJECT = Software Engineering

Q.1: What are the four important attributes that all professional software should have? Suggest four other attributes that may sometimes be significant

Ans:

The four most important characteristics of professional software are Acceptability, Dependability and Security, Efficiency, and Maintainability. Other examples of quality attributes are the software's Performance, Scalability, Reuseability, Testability, Reliability, and Availability

Q.2: Explain why professional software is not just the programs that are developed for a customer.

Ans:

Professional software is not just the programs developed for a customer because the software is almost always packaged with associated documentation such as requirements, design models, and user manuals. A good or professional software goes beyond software developed solely for a customer.

Q.3: Giving reasons for your answer based on the type of system being developed, suggest the most appropriate generic software process model that might be used as a

basis for managing the development of the following systems:

- **A system to control anti-lock braking in a car**
- **A virtual reality system to support software maintenance**
- **A university accounting system that replaces an existing system**
- **An interactive travel planning system that helps users plan journeys with the lowest environmental impact**

Ans:

a) Anti-lock braking system This is a safety-critical system so requires a lot of up-front analysis before implementation. It certainly needs a plan-driven approach to development with the requirements carefully analysed. A waterfall model is therefore the most appropriate approach to use, perhaps with formal transformations between the different development stages.

b) Virtual reality system This is a system where the requirements will change and there will be an extensive user interface components. Incremental development with, perhaps, some UI prototyping is the most appropriate model. An agile process may be used.

c) University accounting system This is a system whose requirements are fairly well-known and which will be used in an environment in conjunction with lots of other systems such as a research grant management system. Therefore, a reuse-based approach is likely to be appropriate for this.

d) Interactive travel planning system System with a complex user interface but which must be stable and reliable. An incremental development approach is the most appropriate as the system requirements will change as real user experience with the system is gained.

Q.4: Explain why incremental development is the most effective approach for developing business software systems. Why is this model less appropriate for real-time systems engineering?

Ans:

Business software systems usually complex, software intensive, and frequently being changes when business goals or processes are changed. So incremental development is better.

Real-time systems usually involve many hardware components which are not easy to change and cannot be incremental. Also real-time systems usually safety critical which needed be built based on well planned process.

Q.5: Suggest why it is important to make a distinction between developing the user requirements and developing system requirements in the requirements engineering process.

Ans:

There is a fundamental difference between the user and the system requirements that mean they should be considered separately.

a) The user requirements are intended to describe the system's functions and features from a user perspective and it is essential that users understand these requirements. They should be expressed in natural language and may not be expressed in great detail, to allow some implementation flexibility. The people involved in the process must be able to understand the user's environment and application domain.

b) The system requirements are much more detailed than the user requirements and are intended to be a precise specification of the system that may be part of a system contract. They may also be used in situations where development is outsourced and the development team need a complete specification of what should be developed. The system requirements are developed after user requirements have been established.

