

1. Suggest the most appropriate software process model that could be used as a basis for managing the development of the following systems, justify your suggestion:
 - a. A system to control anti-lock braking in a car
 - b. A virtual reality system to support software maintenance
 - c. A university accounting system that replaces an existing system
 - d. An interactive travel planning system that helps users plan journeys with the lowest environmental impact

Answer

- 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 novel system where the requirements will change and there will be an extensive user interface component. 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. An incremental development approach is the most appropriate as the system requirements will change as real user experience with the system is gained.
2. 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?

Answer

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 incrementally deployed. Also real-time systems usually safety critical which needed be built based on well planned process.

3. Consider the reuse-based process model. Explain why it is essential to have two separate requirements engineering activities in the process.

Answer

In a reuse based process, you need two requirements engineering activities because it is essential to adapt the system requirements according to the capabilities of the system/components to be reused. These activities are:

1. An initial activity where you understand the function of the system and set out broad requirements for what the system should do. These should be expressed in sufficient detail that you can use them as a basis for deciding if a system/component satisfies some of the requirements and so can be reused.
 2. Once systems/components have been selected, you need a more detailed requirements engineering activity to check that the features of the reused software meet the business needs and to identify changes and additions that are required.
4. Suggest why it is important to make a distinction between developing the user requirements and developing system requirements in the requirements engineering process.

Answer

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

1. 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.
2. 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.

5. In waterfall software development process model, we complete one stage and proceed to the next stage. The engineer may revisit the previous stage if refinement is needed. However, it is important that the number of times the engineer revisits the previous stage should be limited. Explain why the number of revisits should be limited using your own statements.

Answer

Upon completing each stage, the software engineer must present a complete documentation for the stage which represents a lot of effort from the development team. The document must be reviewed, inspected, and finally approved by the relevant authority. If refinement is needed, the documentation must be modified to reflect the refinement. The scope of modification may be large because the refinement may affect other areas of the software. The review, inspection, and approval process must carry out again for the newly modified document, which will consume a lot of time and cost. Therefore, the refinement of the work produced by the previous stage should be limited to ensure the project is not over budget and will complete within schedule.