CS-389

Homework 1

(submission date: September 17th, during class or before the class start)

2.1 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.

1. To design a system to control anti-lock braking in a car the most appropriate generic software process model would most likely be the waterfall design method. This is the best method because the five steps of the design process will create a perfect system for the specified car. The first step in the design process “Requirements Analysis and Definition” is necessary to begin development of this system because you must learn about what the car already uses. For example, the car will most likely use an embedded system requiring you to use either C or assembly to build the software. Another crucial step in the process model is the “Implementation and unit testing” step. Using unit tests in our software is crucial to ensure small bits of our code all throughout our software are working correctly. Step four is another important step as well because we will have to ensure all of our unit tests work and that when the program is running together there is no failures because a failure may actually cause an accident.
2. I believe that the correct generic software process model for developing a virtual reality system to support software maintenance would be the Integration and configuration though I did consider Incremental Development. The reason I thought against Incremental development was to my belief that you may not need to have a lot of versions. My main reason for choosing this model would be due to the specific use cases of the software and the users. This will be a very unique and quite lucky group of developers to use this maintenance software, so the software itself will be developed and must be configured to adhere to their specific requirements. While importantly the software must also have reusable components for future development.
3. The correct generic software process model for developing a university accounting system that replaces an existing system would also be best as Incremental Development. This software process is very rapid which would be perfect for the development of this software during possible one semester or over the summer to allow it to be used as soon as possible. Also due to its early release to customers you can change the feedback to improve the accounting software for the university depending on there needs. And as new needs arise or new functions are developed by the developers they can add the features as they see fit.
4. Lastly, an interactive travel planning system that helps users plan journeys with the lowest environmental impact would be best developed utilizing Incremental Development. This system similar to the accounting system may require many changes due to the users tending to not know what they want right away. This will require many updates to bring those changes and features to the user. But in the end the user will be much happier with an evolving product that will fit there needs and perform all the duties they require.

2.4 Suggest why it is important to make a distinction between developing the

user requirements and developing system requirements in the requirements

engineering process.

It is important to distinguish between developing the user requirements and developing the system requirements because they are completely different things. First to note system requirements after they have been established may not change often. While on the other hand user requirements may change rapidly and they may differ drastically. If you choose allow these two sets of requirements to overlap, then you may run into a problem with both. For example, if the users decide that want something different them you may have to change the system entirely. But if you decide on a system and what it requires you will save large amounts of time and money and can give the user most of the requirements while not all of them may be met.