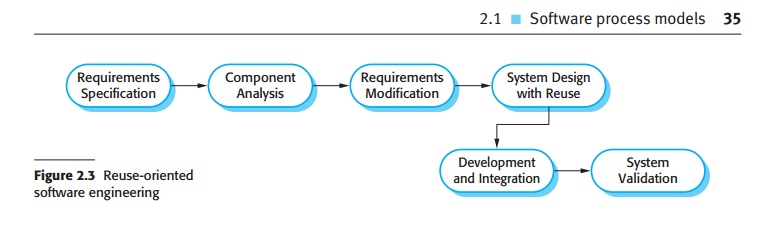
2.2: Explain why incremental development is the most effective approach for developing business

software systems. Why is this model less appropriate for real-time systems engineering?

Answer :

Incremental is most effective approach for bussies software because Bussiness software usually change requirement, change bussines purpose ,add more functionality so incremental development is matched.

Incremental model is not appropriate for real-time systems engineering .Real-time systems include many of hardware, it's difficult to change and impossible to incremental.Real-time system need to be carefully planned like water fall model.

2.3:Consider the reuse-based process model shown in Figure 2.3. Explain why it is essential to have two separate requirements engineering activities in the process?

Answer:

When requirements specification is done. Component Analysis would search component to implement that specification .If not separate requirement engineering activities Requirement Specification and Requirement Modification , the component used to modify might be not satisfy for specification or not enough for it ,then the result will not match to specification.

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.

Answer:

There are two main concept between user requirement and System Requirement that both of them need to be considered separately:

User Requirement is the requirement, functions and system features from user perspective and it essential that user understand these requirement. Using natural language will make them understand better.

System Requirement based User Requirement but deeper and more detail from developer ,coder, who design software perspective.It have to be a precise specification of the system.

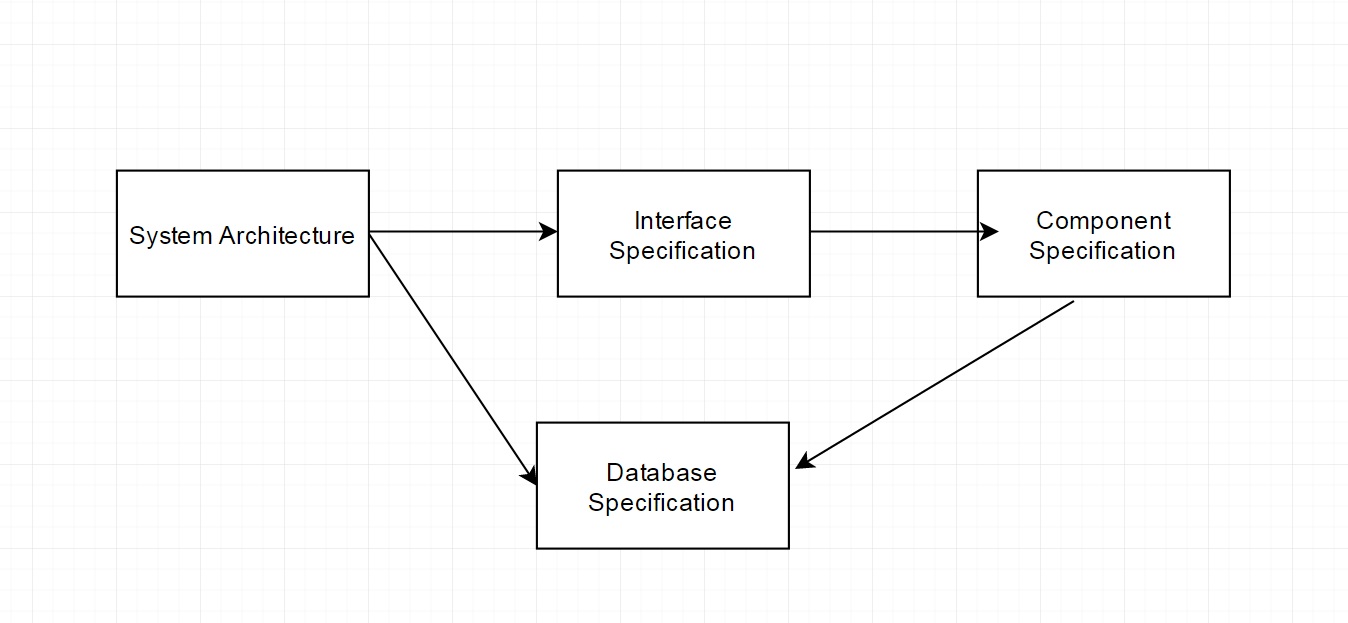
2.5: Describe the main activities in the software design process and the outputs of these activities. Using a diagram, show possible relationships between the outputs of these activities.

Answer:

Architectural Design : Identity overall system structure,principle component,their relationship, and how they distributed.

Interface Design : Interface between system component.

Component Design : take each system component and design how it operate.

Database Design : Design system data structure and how they are presented in database.

2.6. Explain why change is inevitable in complex systems and give examples (apart from prototyping and incremental delivery) of software process activities that help predict changes and make the software being developed more resilient to change.

Answer:

Change is inevitable:

1.The business and technical environment of the system always changes. New hardware always introduced every year, have to change to improve and make it adapt with new hardware.

2.Large System serves many community, whom having different requirement and behavior. New function always add and change.

**2.7.** Explain why systems developed as prototypes should not normally be used as production  
systems

Answer:

1.Prototype isn’t handling code and error is not detected yet.

2.GUI not good design, not intuitive.

3.It look like bad quality software, bad production , cause bad impression to end-user.

2.8. Explain why Boehm’s spiral model is an adaptable model that can support both change avoidance and change tolerance activities. In practice, this model has not been widely used. Suggest why this might be the case.

Answer:

Boehm’s spiral is deal with risk every loop of spiral