

**Question 1:** What is the evolving perspective on testing in software engineering, and how does it differ from the traditional approach?

**Answer:** The evolving perspective on testing in software engineering emphasizes that testing is not limited to debugging programs; rather, it should be an integral part of the entire development process. In contrast to the traditional approach, where testing was considered a separate step, modern software engineers advocate incorporating testing into various development phases. This approach aims to ensure accuracy at each intermediate step and considers testing as part of the broader quality assurance struggle throughout the software life cycle.

**Question 2:** How have software engineering methodologies evolved from the traditional waterfall model, and what are the key features of incremental and iterative models?

**Answer:** Software engineering methodologies have shifted from the strict sequential waterfall model to more flexible approaches like the incremental and iterative models. In incremental development, the software system is constructed in increments, with each version adding functionality. The iterative model involves refining each version rather than just extending it. These models allow for adaptability and creative problem-solving, addressing the limitations of the waterfall model's rigid structure.

**Question 3:** What role does prototyping play in software development, and how do throwaway prototyping and evolutionary prototyping differ?

**Answer:** Prototyping is utilized in incremental and iterative models, with evolutionary prototyping involving the evolution of prototypes into the final system. Throwaway prototyping, on the other hand, discards prototypes in favor of fresh implementations of the final design. Rapid prototyping, a throwaway category, quickly constructs a simple example to facilitate

VU APEX CAMPUS	<a href="http://vuapex.com.pk">vuapex.com.pk</a>	<a href="http://vuapex.pk">vuapex.pk</a>
Contact Us:	0322-8877744	

communication and clarify system requirements during early development stages.

**Question 4:** Why is modularity crucial in software development, and how does it impact the design process in imperative and object-oriented paradigms?

**Answer:** Modularity is essential in software development as it facilitates manageability and future modifications by dividing software into manageable units or modules. In the imperative paradigm, modularity is represented by functions, whereas in the object-oriented paradigm, objects serve as modular constituents. The imperative paradigm designs modules by considering activities that must be performed, while the object-oriented paradigm identifies and understands how objects interact, providing different perspectives on the initial software design process.

VU APEX CAMPUS	<a href="http://vuapex.com.pk">vuapex.com.pk</a>	<a href="http://vuapex.pk">vuapex.pk</a>
Contact Us:	0322-8877744	