

- 12.** The development of distributed systems present the software developer with a number of other considerations not present in other types of solutions. One of the most evident of these is process coordination. Explain.
- 13.** How can one increase overall reliability of a distributed software solution?
- 14.** In your own words, explain the following statement: "There are two aspects to consider when developing software solutions, one is building the thing right, and the other is building the right thing." To what do these two aspects refer?
- 15.** Do the two aspects in the previous question necessarily imply each other? Explain.

- 16.** Explain in your own words why and how formal descriptions and representations, and reasoning (based on such descriptions and representations), is used to ascertain the correctness of software solutions.
- 17.** Sometimes there is confusion when software developers need to deal with reliability and robustness requirements of a software solution. Can you clearly differentiate between these two important quality attributes?
- 18.** Efficiency is something that at first glance seems to relate to the product, i.e. the solution being constructed. However, it pertains to both product and process. Can you explain in what way it pertains to the latter. Give one specific example to substantiate your reasoning.
- 19.** Quite often, user friendliness of a solution is exclusively equated to GUI properties. Is this a fair assumption? Explain.

20. Explain how the quality attribute of Understandability and that of Maintainability are related.

21. The quality attribute of Reusability is considered as a relatively "modern" attribute. Explain why this is so.

22. Reusability is something that requires considerable investment before paying off any dividends. Explain.

23. How does the quality attribute of Portability translate into a software solution of higher quality?

- 24.** What are the issues, in terms of process management, that a "monolithic" development process may incur?
- 25.** The introduction of phased software development over monolithic software development offers an immediate set of advantages. List these.
- 26.** What is meant by the statement "a development phase cannot be open-ended", and why is this the case? How must subsequent software development phases treat abstraction? Why?
- 27.** Explain what you understand a milestone to be in phased development.

28. Both the analysis and implementation phases are unique - differently in their own ways. Explain the two cases.

29. Why is it important to distinguish and separate testing into "component" and "integration"?

30. What would you consider the main drawback of the traditional Waterfall SDLC to be? Explain.

31. How does the V-shaped model help address the issue of late product maturity?

- 32.** How does the Incremental model help address the issue of late stakeholder feedback?
- 33.** What is the difference between evolutionary and throw-away prototyping.
- 34.** Prototyping is a central tool used by software developers, especially in an RAD context. It nevertheless exhibits a definite inadequacy. What is this inadequacy, and how can it be addressed?
- 35.** Explain the notion of a rigid timebox in RAD, and how, while being rigid, it can nevertheless cater for realistic progression of development.

- 36.** List and outline *two* roles of a JAD, or similar grouping, in the context of RAD.
- 37.** Outline and justify the notion and use of "reversibility" in RAD.
- 38.** Explain how the effort from the developers' side relating to requirements establishment is alleviated in the context of RAD.
- 39.** What is the difference between the notion of a "method" and that of a "technique"?

40. What is the difference between RAD and DSDM?

41. What does the "DSDM Filter" attempt to achieve?

42. Solutions where user groups are easily identifiable and requirements can be prioritised are suited for DSDM-based development. Explain why this is so.

43. List and explain two pros and two cons of formally specifying system behaviour.

44. Why is the notion that formal specifications are shunned by stakeholders not true?