

1. (True/False) Prerequisites are not that important. A high quality, high risk project can be created while ignoring prerequisites.

Answer : False

2. What is the overarching goal of using prerequisites?

- A. Spending more time on work
- B. Risk Reduction
- C. Remaking code from previous projects

Answer : B

3. What are some common risks in software development?

Answer : Poor project requirements and poor project planning

4. What is a benefit of being able to find defects or errors in our code earlier on?

Answer : Because leaving defects for later discovery can be costly, discovering defects or errors in the code earlier can fix them at a lower cost, avoiding risks and avoiding increased costs.

5. (True/False) There is not that much variation in software projects.

Answer : False

6. Please describe the differences between an iterative approach and a sequential approach to software development. Please list out some scenarios where one approach might be better to use than the other.

Answer : In general, Iterative applies to projects with rapidly changing needs, while parallel Approaches apply to projects with relatively stable needs. Business system projects benefit from a highly Iterative approach, where planning, requirements, and architecture are put together with construction, system testing, and so on. Life-critical systems benefit from a sequential approach that allows project requirements to be stable and brings a very high degree of reliability to the project. For example, Sequential Approaches are better than Iterative when the following scenarios occur:

- Stable demand
- The design is easy to understand
- The development team is familiar with the application
- Low project risk
- Long-term predictability is important
- Changing requirements, design, and code is costly

7. Please describe in your own words what a problem definition prerequisite is.

Answer : Problem definitions define what the problem is without reference to possible solutions. It should be a simple statement, and it should sound like a problem. The statement "user cannot query history orders" sounds like a problem, but it's a good problem definition. Saying "we need to add a button to display the history order so that the user can view the history order directly" is a bad definition of a problem. That doesn't sound like a problem; That sounds like a solution.

8. In a problem-definition, do you specify the solution to your problem? Why or why not?

Answer : A problem definition is a definition of what we want to solve before we code it. It only

describes the problem itself and does not provide any possible solutions. Because the solution can be misleading and cause us to waste time trying to solve the wrong problem.

9. Please describe in your own words what requirement prerequisites are.

Answer : First of all, we must first clarify the definition of the problem, and then make sure to describe the requirements. The requirements describe in detail what the software system should do. It includes:

Requirements development, requirements analysis, analysis, requirements definition, software requirements, specifications, specifications, functional specifications, etc.

It is necessary to clarify the official requirements, which can determine the minimum scope of development, which allows users to drive requirements, rather than programmers.

Avoid guessing what the user wants, but let the user make the request.

10. We saw that official requirements can be beneficial in this chapter, what are some benefits of having official requirements?

Answer : Clear official requirements help us define a minimum development scope and allow users to drive requirements, avoiding programmers guessing what users want. When you encounter confusion during development, you can look at the requirements that the user wants. Moreover, the official requirements can minimize changes to the system after development, because the late change costs can be very high.

11. (True/False) When working on a project, you should assume that the requirements will never change.

Answer : False

12. Please describe in your own words what Architecture Prerequisites are

Answer : The quality of the architecture determines the conceptual integrity of the system, which in turn determines the overall quality of the software. Architecture can provide top-down guidance to programmers as a blueprint for the development process, Good software architecture makes building software easy, while bad software architecture makes building software impossible. Late architectural changes can be costly.

13. In your own words, what is the difference between coding in a language and coding into a language.

Answer : "Programming in a language" limits the programmer's mind to the structure of the language, getting caught up in the principles and techniques generated by the Programming language features, and neglecting the actual requirements that need to be implemented.

"Programming into a Language" allows you to think independently of the development environment. This allows you to use the programming language to express the requirements and designs that you actually want to implement. "Programming into a Language" is not specific to a language; they can be used in a variety of areas.

14. (True/False) Java is the best language to code in

Answer : False

15. Can you list out some examples of Major Construction practices we saw in Chapter 4?

Answer : pair programming, test-first development.

16. What is the first stage of starting when starting a software development project?

- A. Construction
- B. System Testing
- C. Requirements
- D. Problem Definition

Answer : D

17. If we focus on using high-quality practices at the start of our project, what type of practices will we be emphasized?

- A. System testing
- B. Defining a problem, determining the solution, and designing that solution
- C. Construction and coding practices

Answer : B

18. (True/False) Fixing a defect in our project during the architecture phase costs roughly the same amount as during the construction phase.

Answer : False

19. Which of the following is not a general category that software projects fall under?

- A. Simplistic Systems
- B. Business Systems
- C. Embedded Life-Critical Systems
- D. Mission-Critical Systems

Answer : A

20. In the architectural components we saw that Buy vs. Build decisions should be laid out in the architectural blueprints of a software project. Please describe in your own words what buy vs. build decisions are?

Answer : When there is a software product that perfectly meets our needs, it should be a priority to buy it rather than rebuild it. Because there is a huge difference in cost and it takes more time and resources to build the software.