1. **Which challenge is more common in IoT software than in traditional desktop apps?**
2. Managing memory limits on microcontrollers (correct)
3. Designing user interfaces (correct)
4. Requiring high-speed internet
5. Using object-oriented languages
6. **Which is more critical in a plant watering IoT system than in traditional software?**
7. High-res GUIs
8. Handling spotty connection and saving power (correct)
9. Less focus on security
10. No need for scalability
11. **At the middle of the V-Model for a plant watering IoT system, what is mainly done?**
12. Fixing bugs from deployment
13. Defining overall system goals
14. Implementing and integrating components
15. Writing acceptance test plans
16. **Which of the following is considered a major (minor, patch) architectural change in a plant watering IoT system?**
17. UI redesign to improve user experience
18. Changing the sync interval between the devices and the cloud
19. Table schema update to add new attributes
20. Updating the firmware on the plant sensors
21. **If a plant watering IoT system is built as one monolithic application, what team structure does it suggest?**
22. Multiple small, isolated teams
23. One large, centralized team
24. Randomly assigned individuals
25. Two cross-functional teams
26. **In the V-Model, if a user story validation fails after implementation, what is the correct next step?**
27. Escalate to design phase
28. Escalate to requirements verification phase
29. Escalate to system testing phase
30. Escalate to next development cycle
31. **In the V-Model, if architecture verification fails in a plant watering IoT system, what should happen next?**
32. **Move to coding anyway**
33. **Return to system design**
34. **Start integration testing**
35. **Update unit test cases**
36. **Why can’t the backlogs for Agile and V-Model be the same in a plant watering IoT system project?**
37. **Agile doesn’t define backlog items, but V-Model does**
38. **The V-Model backlog is static, while Agile requires flexibility**
39. **Agile backlogs are only for UI tasks, while the V-Model backlog focuses on testing**
40. **V-Model backlogs are predefined at the start, while Agile evolves over time**
41. **Which phase of the V-Model for a plant watering IoT system typically requires the most incremental steps?**
42. **Requirement’s analysis**
43. **System design**
44. **Testing and validation**
45. **Deployment**
46. **How does the architecture of the plant watering IoT system affect the scalability of development teams?**
47. **Microservices architecture allows teams to (horizontal) scale independently by focusing on individual services but not for IoT**
48. **A monolithic architecture limits (horizontal) scalability due to tightly coupled components**
49. **Layered design requires more teams for each layer, but it scales easily**
50. **Serverless architecture scales automatically, requiring no extra team effort**
51. **How does the V-Model address compliance with legal requirements?**
52. By ensuring legal compliance is checked only during the deployment phase
53. Through emphasize on verification, validation, and testing
54. By skipping testing and focusing solely on design and documentation
55. By relying on external audits after the project is completed
56. **What is the difference between verification, validation, and testing?**
57. Verification ensures user/customer needs, validation checks functionality, and testing finds bugs
58. Verification checks requirements, validation checks the right problem, and testing checks errors
59. Verification and validation are the same, testing is done at the end
60. Verification is during implementation, validation post-deployment, and no testing needed
61. **Which of the following best describes the "Arrange, Act, Assert" (AAA) pattern in testing?**
62. Arrange the function, assert expected behavior, and act to test
63. Act first, then arrange test data, and finally assert the expected result
64. Arrange test data, act on the system, and assert the outcome
65. Assert the behavior, act to modify it, and arrange the conditions
66. **Why is the AAA testing principle useful?**
67. It is only useful for unit tests
68. It limits tests to Arrange and Assert steps
69. It provides structured tests for all levels
70. It works only for low-level tests
71. **Which stage of the V-Model typically generates the most bug tickets in a plant watering IoT system?**
72. **System testing**
73. **Integration testing**
74. **Component testing**
75. **Unit testing**
76. **How do alpha, beta, and gamma tests relate to V-Model testing?**
77. **They are not required in V-Model, but nice to have**
78. **They extend it with real user testing**
79. **They test software hardware integration**
80. **They follow failed unit tests**
81. **By integrating Scrum into the V-Model, we can:**
82. Validate much earlier
83. Verify much earlier
84. Test much earlier
85. **Which level of testing in the V-Model is usually the most difficult to fix if issues are found?**
86. Unit testing
87. Component testing
88. Integration testing
89. System testing
90. **Which part of plant watering system fits user-based access control?**
91. Embedded
92. Backend
93. Database
94. Frontend
95. **How can you ensure that the plant water sensor is providing accurate data over time?**
96. **By running periodic calibration checks against a known reference**
97. **By assuming the sensor will maintain accuracy without testing**
98. **By testing the sensor only during the initial installation**
99. **By monitoring the system for sensor errors after deployment**
100. **How can you optimize battery life in embedded devices used for a plant watering IoT system?**
101. **By using high-power sensors for continuous data collection**
102. **By optimizing software to reduce processing and communication frequency**
103. **By increasing the sensor's transmission range to collect more data**
104. **By constantly sending data to the cloud, regardless of battery status**
105. **Why do IoT projects, like a plant watering system, often have a fixed completion date?**
106. Because software development phases are predictable
107. Because user feedback must be collected before starting
108. Because production lines are often third-party with fixed time slots
109. Because the cloud deployment process is fast
110. **A Plant Watering Application customer calls and says, "*Last week the application sent me regular notifications, but this week it is not working anymore.*" Please list 5 reasons why notifications may not be working.**
111. **The Plant Watering Application is required to work outdoors as well. Please list 5 changes that must be made to enable the application to run outdoors.**