1. Which of the following is a test activity?
2. **Test planning**
3. Writing methods
4. Debugging
5. Using reflection
6. In order to avoid conflict with developers, what skills should a QA possess?
7. Good test planning skills
8. Accept that developers are always right
9. **Good communication skills**
10. Excellence in test execution
11. Which of the statements below is the best assessment of how the test principles apply across the test life cycle?
12. Test principles only affect the preparation for testing
13. **Test principles affect activities throughout the test life cycle**
14. Test principles only affect test execution activities
15. Test principles affect the early test activities such a review
16. Which of the following is an agile methodology?
17. Scrum
18. Kanban
19. **All of the above**
20. Lean
21. Which of the following is NOT an actual reason for an incident to occur?
22. **Office environment**
23. Corrupted test data
24. Inefficient communication
25. Poor training of tester
26. What does the SDLS describe?
27. Software defects
28. Software functionality
29. **Stages in the development of a software**
30. Software as a service
31. Which of the following is NOT a testing principle?
32. Exhaustive testing is impossible
33. Defects cluster together
34. Beware of the pesticide paradox
35. **Every test should hold single responsibility**
36. What does the STLC describe?
37. Test planning
38. Reporting
39. Test implementation and test execution
40. **Testing phases throughout the development of a project**
41. Which opinion best describes objectives for test levels with a life cycle model?
42. The objectives of a test level don’t need to be defined in advance.
43. Objectives should be generic for any test level.
44. **Each level objectives specific to that level.**
45. Objectives are the same for each test level.
46. Acceptance testing:
47. Considers legal or regulatory requirements
48. Meets the system actual behavior with the client expectations.
49. Validates the end-to-end business flow.
50. **All of the above.**
51. Component testing includes:
52. Testing the whole system
53. **Testing separate components of the software**
54. Testing a few components together.
55. Testing usability of the system
56. Which of the following is a non-functional quality characteristic?
57. Maintenance
58. Regression
59. **Usability**
60. Feasibility
61. Non-functional testing includes:
62. Gaining user approval for the system
63. Testing to see where the system does not function correctly
64. **Testing the quality attributes of the system including reliability and usability**
65. Testing a system feature using only the software required for that function
66. Which test level applies to the following statement: “Testing must confirm that all components collaborate correctly”?
67. **Integration testing**
68. Functional testing
69. Non-functional testing
70. Acceptance testing
71. Which of the following is a test type?
72. Acceptance testing
73. Component testing
74. **Functional testing**
75. System testing
76. An incident logging system:
77. **Is a valuable source of project information during testing if it contains all incidents**
78. Only records defects
79. Should be used only by the test team
80. Is of limited value
81. The correct order of states in bug life cycle are:
82. **Open, Assigned, Fixed, Closed**
83. Assigned, Open, Closed, Fixed
84. Assigned, Open, Fixed, Closed
85. Open, Fixed, Assigned, Closed
86. Expected results are:
87. **Most useful when specified in advance**
88. Derived from the code
89. Only important in system testing
90. Only used in component testing
91. When reporting faults found to developers, testers should be:
92. Diplomatic, sensitive to the way they may react to criticism
93. **All of the above**
94. Firm about insisting that a bug is not a “feature” if it should be fixed
95. As polite, constructive and helpful as possible
96. Which of the following characterizes the cost of faults?
97. They are easiest to find during system testing but the most expensive to fix then
98. **They are cheapest to find in the early development phases and the most expensive to fix in the latest test phases.**
99. Although faults are the most expensive to find during early development phases, they are cheapest to fix then
100. Faults are cheapest to find in the early development phases but the most expensive to fix then
101. Which of the following is NOT true of incidents?
102. Incidents require investigation and/or correction.
103. Incidents are raised when expected and actual results differ.
104. Incidents may be raised against requirements.
105. **Incidents resolution is the responsibility of the author of the software under test.**
106. Static code analysis typically identifies all but one of the following problems. Which is it?
107. **Faults in the requirements**
108. Too few comments
109. Undeclared variables
110. Unreachable code
111. Which is not a type of review?
112. Inspection
113. Informal review
114. **Management approval**
115. Walkthrough
116. Which of the following artifacts can be examined by using review techniques?
117. **All of the above**
118. Software
119. Test design
120. Requirement specification
121. What statement about review is true?
122. Technical reviews are led by a trained leader, inspections are not
123. Participants for a walkthrough always need to be thoroughly trained
124. **Inspections are led by a trained moderator, whereas technical review are not necessarily**
125. In a walkthrough the author does not attend
126. What is a key characteristic of structure-based testing techniques?
127. They are mainly used to assess the structure of a specification
128. They use a formal or informal model of the software or component
129. **They are used both to measure coverage and to design tests to increase coverage**
130. They are based on the skills and experience of the tester
131. What is the main difference between a walkthrough and an inspection?
132. Authors are not present during inspections, whilst they are during walkthrough
133. **A walkthrough is led by the author, whilst an inspection is led by a trained moderator**
134. An inspection is led by the authors, whilst a walkthrough is led by a trained moderator
135. An inspection has a trained leader, whilst a walkthrough has no leader
136. Why are error guessing and exploratory testing good to do?
137. They will ensure that all of the code or system is tested
138. They don’t require any training to be as effective as formal techniques
139. **They can find defects missed by specification-based and structure-based techniques**
140. They can be used most effectively when there are good specifications
141. What statement about static analysis is true?
142. When properly performed, static analysis makes functional testing redundant
143. **With static analysis, defects can be found that are difficult to find with dynamic testing**
144. Compiling is not a form of static analysis
145. Static analysis finds all faults
146. Postal rates for ‘light letters’ are 25p up to 10g, 35p up to 50g plus extra 10p for each additional 25g up to 100. Which test inputs (in grams) would be selected using equivalence partitioning?
147. 5, 20, 40, 60, 80
148. 8, 42, 82, 102
149. 10, 50, 75, 100
150. **4, 15, 65, 92, 159**
151. How do experience-based techniques differ from specification-based techniques?
152. They depend on tester’s understanding of the way the system is structured rather than on a document record pf what the system should do.
153. **They depend on an individual’s personal view rather than on a documented record of what the system should do**
154. They depend on having alder testers rather than younger testers
155. They depend on a documented record of what the system should do rather than on an individual’s personal view.
156. What do we mean when we call someone a test manager?
157. A test manager manages a collection of test leaders
158. A test manager reports to a test leader
159. **A test manager is the leader or a test team or teams**
160. A test manager gets paid more than a test leader
161. Which of the following elements of the test plan, while specified during test planning, is assessed during test execution?
162. Test team training
163. Test tasks
164. **Exit criteria**
165. Environmental needs
166. A product risk analysis meeting is held during the project planning period.
167. **The harm that might result to the user**
168. The price for which the software is hold
169. Difficulty of fixing related problems in code
170. The technical staff in the meeting
171. Why is independent testing is important?
172. Independent testing is usually cheaper than testing your own work
173. **Independent testing is more effective at finding defects**
174. Independent testers are dispassionate about whether the project succeeds or fails
175. Independent testers should determine the processes and methodologies used.
176. You are working as a tester on a project to develop a point-of-sales system for grocery stores and other similar retail outlets. Which of the following is a product risk for such a project?
177. An excessively high number of defect fixes fail during re-testing
178. **Failure to accept allowed credit cards.**
179. Th arrival of a more-reliable competing product on the market
180. Delivery of an incomplete test release to the first cycle of system test
181. Which of the following is among the typical tasks of a test leader?
182. **Gather and report test progress metrics**
183. Keep tests and test coverage hidden from the programmers
184. Develop system requirements, design specifications and usage models
185. Handle all test automation duties.
186. Project risk can be divided into categories:
187. **Technical issues, organizational factors, supplier issues**
188. Data standarts, test planning, organizational factors
189. Configuration management, test planning, risk testing
190. Risk testing, technical issues, supplier issues
191. Test coverage can be calculated by:
192. Dividing the percentage of test environment completion by the total amount of preparation required
193. Dividing total number of defects by the number of modules
194. Dividing number of test cases executed by the total number of planned test cases to be executed
195. **Dividing number of requirements tested by the total number of requirements**
196. A product risk is related to which of the following?
197. A potential negative outcome
198. **The test object**
199. A single test item
200. Control of the test project
201. Which of the following is NOT a test progress monitoring task?
202. Measuring progress against milestones
203. Monitoring test execution progress
204. **Paying attention to the work schedule of the test team**
205. Monitoring status of a test project
206. Which of the following items would not come under Configuration Management?
207. Test documentation
208. Operating system
209. **Live data**
210. User requirement documents
211. The status of test activities is defined by:
212. Test environment preparation, defect information, test case competition, tested object
213. **Test case competition, tested object, quantity of incomplete tests, number of open defects, regression testing and retesting required**
214. defect density, test coverage, percentage of test case preparation
215. test case failure, test executed, failures per unit of time, ratio of the number of failures
216. Percentage of test case execution can be calculated by:
217. **Dividing number of test cases executed by the total number of planned test cases to be executed**
218. Dividing total number of defects by the number of modules
219. Dividing the percentage of test environment completion by the total amount of preparation required
220. Dividing number of requirements tested by the total number of requirements
221. Which of the following is NOT an objective of Configuration Management?
222. Providing accurate information on time
223. Supporting incident and change management processes
224. **Optimizing the monitoring and control activities**
225. Elimination of data and effort duplication