|  |  |
| --- | --- |
| QN=1 (1) | One reason to involve everyone on the software team in the planning activity is to |
| a. | adjust the granularity of the plan |
| b. | control feature creep |
| c. | get all team members to “sign up” to the plan |
| d. | understand the problem scope |

|  |  |
| --- | --- |
| QN=2 (6) | Refactoring keeps the code simple and \_\_\_\_\_ . |
| a. | maintainable |
| b. | readable |
| c. | executable |
| d. | efficient |

|  |  |
| --- | --- |
| QN=3 (11) | Verification is |
| a. | Checking that we are building the right system |
| b. | Checking that we are building the system right |
| c. | Performed by an independent test team |
| d. | Making sure that it is what the user really wants |

|  |  |
| --- | --- |
| QN=4 (16) | The Agile Manifesto states the following values |
| a. | People are more important than contracts |
| b. | Working software should have priority over comprehensive documentation |
| c. | Plans should have priority over ability to respond |
| d. | Contracts should be negotiated which allow control over the people |

|  |  |
| --- | --- |
| QN=5 (21) | Which one below is not the attributes of good software? |
| a. | Maintainable |
| b. | Dependable |
| c. | Usable |
| d. | Executable |

|  |  |
| --- | --- |
| QN=6 (26) | What should not be included in scenarios? |
| a. | A description of the starting situation |
| b. | A description of the normal flow of events |
| c. | A description of the state when the scenario finishes |
| d. | A description of the software system performance |

|  |  |
| --- | --- |
| QN=7 (31) | Software is both a \_\_\_\_\_\_\_\_ and a vehicle that delivers a \_\_\_\_\_\_\_\_\_. |
| a. | Product |
| b. | Process |
| c. | Design |
| d. | Concept |

|  |  |
| --- | --- |
| QN=8 (36) | Which one below is not incremental development benefits? |
| a. | The cost of accommodating changing customer requirements is reduced. |
| b. | It is easier to get customer feedback on the development work that has been done. |
| c. | It can be easily applied in large scale project. |
| d. | More rapid delivery and deployment of useful software to the customer is possible. |

|  |  |
| --- | --- |
| QN=9 (41) | The reason for software bugs and failures is due to |
| a. | Software companies |
| b. | Software Developers |
| c. | Both Software companies and Developers |
| d. | All of the mentioned above |

|  |  |
| --- | --- |
| QN=10 (46) | In principle, requirements should be both \_\_\_\_\_ and \_\_\_\_\_. |
| a. | complete; general |
| b. | simple; consistent |
| c. | complete; consistent |
| d. | long; detailed |

|  |  |
| --- | --- |
| QN=11 (51) | What is the Agile approach to doing design early in a project? |
| a. | A big design up front is always a good idea |
| b. | Just enough design up front gives a good foundation to start from and helps to mitigate risk, without wasting unnecessarily time |
| c. | No design up front is the best approach as most of the fun of a project is in the discovery of the unexpected |
| d. | Design has no place in an Agile project |

|  |  |
| --- | --- |
| QN=12 (56) | Which one below is not a stage of requirement engineering? |
| a. | Requirements discovery |
| b. | Requirements classification and organization |
| c. | Requirements prioritization and negotiation |
| d. | Requirements explanation |

|  |  |
| --- | --- |
| QN=13 (61) | Prototyping aims at |
| a. | end user understanding and approval |
| b. | program logic |
| c. | planning of dataflow organization |
| d. | none of the mentioned above |

|  |  |
| --- | --- |
| QN=14 (66) | For business modeling of a human organization or the workflow of a system, which one of the following OOAD artifacts is the MOST useful? |
| a. | Interaction Diagrams |
| b. | Activity Diagrams |
| c. | Package Diagrams |
| d. | State Diagrams |
| e. | Class Diagrams |

|  |  |
| --- | --- |
| QN=15 (72) | A software process model is a/an \_\_\_\_\_ representation of a process. |
| a. | concrete |
| b. | sample |
| c. | abstract |
| d. | limited |

|  |  |
| --- | --- |
| QN=16 (77) | Every software engineer must recognize that |
| a. | They are the ultimate authority on software design |
| b. | Change is natural |
| c. | Clients know little about what they need |
| d. | Collaboration is unimportant  e..  f. |

|  |  |
| --- | --- |
| QN=17 (82) | What do all Agile approaches have in common? |
| a. | A prescribed, fixed iteration (Timebox) length |
| b. | Iterative development and incremental delivery |
| c. | A strict focus on on-time delivery |
| d. | A large set of clearly defined roles |

|  |  |
| --- | --- |
| QN=18 (87) | Which of the following is not one of core principles of software engineering practice? |
| a. | All design should be as simple as possible, but no simpler. |
| b. | A software system exists only to provide value to its users. |
| c. | Pareto principle (20% of any product requires 80% of the effort). |
| d. | Remember that you produce others will consume |

|  |  |
| --- | --- |
| QN=19 (92) | Which testing phase tests individual software modules combined together as a group? |
| a. | Module testing |
| b. | Integration testing |
| c. | White box testing |
| d. | Unit testing |

|  |  |
| --- | --- |
| QN=20 (97) | Essential software product attributes are \_\_\_\_\_\_\_, dependability and security, efficiency and acceptability |
| a. | performance |
| b. | easy to use |
| c. | maintainability |
| d. | reusability |

|  |  |
| --- | --- |
| QN=21 (102) | The result of the requirements engineering task is an analysis model that defines which of the following problem domain(s)? |
| a. | informational |
| b. | functional |
| c. | behavioral |
| d. | all of the mentioned above |

|  |  |
| --- | --- |
| QN=22 (107) | Which one below is not a software process model? |
| a. | waterfall |
| b. | pair programming |
| c. | incremental development |
| d. | reuse-oriented |

|  |  |
| --- | --- |
| QN=23 (112) | Which one below is not the fundamental software engineering activities? |
| a. | Software specification |
| b. | software development |
| c. | software validation |
| d. | software deployment |

|  |  |
| --- | --- |
| QN=24 (117) | In practice, most large systems are |
| a. | developed using a process that incorporates elements from all of software process models. |
| b. | developed using a process that incorporates elements from one of software process models. |
| c. | developed using a special software process model. |
| d. | developed using a process that incorporates elements from some of software process models. |

|  |  |
| --- | --- |
| QN=25 (122) | System requirement document is a/an \_\_\_\_\_ document setting out detailed descriptions of the system’s functions, services and operational constraints. |
| a. | structured |
| b. | unstructured |
| c. | detailed |
| d. | general |

|  |  |
| --- | --- |
| QN=26 (127) | Use-cases are a scenario based technique in the UML which identify the \_\_\_\_\_ in an interaction and which describe the interaction itself. |
| a. | actors |
| b. | customers |
| c. | validators |
| d. | testers |

|  |  |
| --- | --- |
| QN=27 (132) | The implementation of cloud computing requires the development of an architecture that encompasses \_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_ services |
| a. | Good; bad |
| b. | Complete; incomplete |
| c. | Collaborative; combative |
| d. | Front-end; back-end |

|  |  |
| --- | --- |
| QN=28 (137) | The requirements engineering process is an iterative process including requirements elicitation, specification and \_\_\_\_\_. |
| a. | validation |
| b. | extraction |
| c. | evolvement |
| d. | maintenance |

|  |  |
| --- | --- |
| QN=29 (142) | Both quality and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ are an outgrowth of good design |
| a. | maintainability |
| b. | communication |
| c. | review |
| d. | criticism |

|  |  |
| --- | --- |
| QN=30 (147) | Which of these does not account for software failure? |
| a. | Increasing Demand |
| b. | Low expectation |
| c. | Increasing Supply |
| d. | Less reliable and expensive |

|  |  |
| --- | --- |
| QN=31 (152) | Which of the items listed below is not one of the software engineering layers? |
| a. | Process |
| b. | Manufacturing |
| c. | Methods |
| d. | Tools |

|  |  |
| --- | --- |
| QN=32 (157) | Which OOAD artifact is the MOST useful in situations where asynchronous events occur? |
| a. | State diagrams |
| b. | Activity diagrams |
| c. | Package Diagrams |
| d. | Class Diagrams |

|  |  |
| --- | --- |
| QN=33 (162) | Which of the following is not one of the principles of good coding? |
| a. | Allows developers to make changes to the delivered increment |
| b. | Delivery schedule can be revised to reflect changes Law of unintended consequences |
| c. | Developers can identify changes to incorporate into next increment |
| d. | All of the mentioned above |

|  |  |
| --- | --- |
| QN=34 (167) | What is a Software? |
| a. | Software is set of programs |
| b. | Software is documentation and configuration of data |
| c. | Software is set of programs, documentation & configuration of data |
| d. | None of the mentioned above |

|  |  |
| --- | --- |
| QN=35 (172) | The intent of agile requirements engineering is to transfer \_\_\_\_\_\_\_\_\_\_\_ from stakeholders to the software team |
| a. | questions |
| b. | tasks |
| c. | behavior |
| d. | ideas |

|  |  |
| --- | --- |
| QN=36 (177) | For showing how several objects collaborate in single use case, which one of the following OOAD artifacts is the MOST userful? |
| a. | Interaction Diagrams |
| b. | Activity Diagrams |
| c. | Package Diagrams |
| d. | State Diagrams |
| e. | Class Diagrams |

|  |  |
| --- | --- |
| QN=37 (183) | Every \_\_\_\_\_\_\_\_\_\_\_\_ indicates an error in design or in the process through which design was translated into machine executable code |
| a. | Hardware failure |
| b. | Miscalculation |
| c. | Software failure |
| d. | Engineering mistake |

|  |  |
| --- | --- |
| QN=38 (188) | Which of the following pattern involves a single class which is responsible to create an object while making sure that only single object gets created |
| a. | Factory Pattern |
| b. | Abstract Factory Pattern |
| c. | Singleton Pattern |
| d. | Transfer Object Pattern |

|  |  |
| --- | --- |
| QN=39 (193) | For purposes of behavior modeling an event occurs whenever |
| a. | consumer or producer of data. |
| b. | data object hierarchy. |
| c. | observable mode of behavior. |
| d. | well defined process. |

|  |  |  |  |
| --- | --- | --- | --- |
| QN=40 (198) | In agile development, specification, design, implementation and testing are \_\_\_\_\_. | | |
| a. | Consecutive | | |
| b. | inter-leaved | | |
| c. | Separated | | |
| d. | unrelated | | |
| QN=41 (203) | | For showing scenarios, which one of the following OOAD artifacts is the MOST useful? |
| a. | | Interaction Diagrams |
| b. | | Activity Diagrams |
| c. | | Use Cases |
| d. | | State Diagrams |
| e. | | Class Diagrams |

|  |  |
| --- | --- |
| QN=42 (209) | Which layer is missing in the sociotechnical system stack as shown below |
| a. | organizational layer |
| b. | application layer |
| c. | physical layer |
| d. | transport layer |

|  |  |
| --- | --- |
| QN=43 (214) | Which of the following best represents the Agile approach to planning? |
| a. | Planning is not part of an Agile approach, because Agile is exploratory |
| b. | Planning should be done in detail at the outset of a project and not revisited |
| c. | Planning should involve the whole team, not just the Project Manager |
| d. | Planning should all be done by the Project Manager |

|  |  |
| --- | --- |
| QN=44 (219) | When you want to look at the behavior across many use cases or many threads, which one of the following OOAD artifacts is the MOST useful? |
| a. | Activity Diagrams |
| b. | Package Diagrams |
| c. | State Diagrams |
| d. | Class Diagrams |
| e. | Sequence Diagrams |

|  |  |
| --- | --- |
| QN=45 (225) | For showing detailed design of procedures, which one of the following OOAD artifacts is the MOST useful? |
| a. | Interaction Diagrams |
| b. | Activity Diagrams |
| c. | Package Diagrams |
| d. | State Diagrams |
| e. | Class Diagrams |

|  |  |
| --- | --- |
| QN=46 (231) | Activity diagrams CANNOT be used in the following situation: |
| a. | Analyzing a use case |
| b. | Understanding workflow |
| c. | Describing a complicated sequential algorithm |
| d. | Dealing with multithreaded applications |
| e. | Procedural flow of control |
| f. | Representing complex conditional logic |

|  |  |
| --- | --- |
| QN=47 (238) | Roughly \_\_\_\_ of software costs are development costs, \_\_\_\_ are testing costs. |
| a. | 40%; 60% |
| b. | 50%; 50% |
| c. | 60%; 40% |
| d. | 70%; 30% |

|  |  |
| --- | --- |
| QN=48 (243) | The high-level activities of \_\_\_\_\_, development, validation and evolution are part of all software processes |
| a. | specification |
| b. | configuration |
| c. | requirement |
| d. | collaboration |

|  |  |
| --- | --- |
| QN=49 (248) | Software Maintenance includes |
| a. | Error corrections |
| b. | Enhancements of capabilities |
| c. | Deletion of obsolete capabilities |
| d. | All of the mentioned above |

|  |  |
| --- | --- |
| QN=50 (253) | When you want to look at the behavior of a single object across many use cases, which one of the following OOAD artifacts is the MOST useful? |
| a. | Activity Diagrams |
| b. | Package Diagrams |
| c. | State Diagrams |
| d. | Class Diagrams |
| e. | Sequence Diagrams |

|  |  |
| --- | --- |
| QN=51 (259) | Requirements engineering is the process of developing a software \_\_\_\_\_\_\_\_\_. |
| a. | specification |
| b. | design |
| c. | implementation |
| d. | validation |

|  |  |
| --- | --- |
| QN=52 (264) | If software has value, it will change over its useful life. For that reason, software must be built to be \_\_\_\_\_\_\_\_\_\_\_\_ |
| a. | maintainable |
| b. | disposable |
| c. | broken |
| d. | tested |

|  |  |
| --- | --- |
| QN=53 (269) | Software process \_\_\_\_\_\_\_\_\_\_\_\_ is essential for project success. |
| a. | communication |
| b. | planning |
| c. | adaptation |
| d. | modeling |

|  |  |
| --- | --- |
| QN=54 (274) | Which one of the following is not a phase of Prototyping Model? |
| a. | Quick Design |
| b. | Coding |
| c. | Prototype Refinement |
| d. | Engineer Product |

|  |  |
| --- | --- |
| QN=55 (279) | Development testing includes |
| a. | unit testing |
| b. | component testing |
| c. | system testing |
| d. | all of the mentioned above |

|  |  |
| --- | --- |
| QN=56 (284) | Before launching a software which testing is to be done in-house? |
| a. | Beta |
| b. | Gamma |
| c. | Alpha |
| d. | None of the mentioned above |

|  |  |
| --- | --- |
| QN=57 (289) | A good OO development is: |
| a. | Use case driven |
| b. | Data driven |
| c. | Responsibility driven |
| d. | All of the mentioned above |

|  |  |
| --- | --- |
| QN=58 (294) | The Agile way is |
| a. | To produce working product of the right quality, early and incrementally |
| b. | To produce working product after documentation has been signed off |
| c. | To produce simple prototypes early, but no finished product until the end of the project |
| d. | To produce products without technical integrity, but re-engineer later |

|  |  |
| --- | --- |
| QN=59 (299) | Inspections and testing are \_\_\_\_\_ and not opposing verification techniques. |
| a. | complementary |
| b. | separated |
| c. | similar |
| d. | contrary |

|  |  |
| --- | --- |
| QN=60 (304) | Which of these are the 5 generic software engineering framework activities? |
| a. | communication, planning, modeling, construction, deployment |
| b. | communication, risk management, measurement, production, reviewing |
| c. | analysis, designing, programming, debugging, maintenance |
| d. | analysis, planning, designing, programming, testing |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MULTIPLE CHOICES QUESTIONS:(Mark=60)

[id=1, Mark=1]1. C

[id=6, Mark=1]2. A

[id=11, Mark=1]3. B

[id=16, Mark=1]4. B

[id=21, Mark=1]5. D

[id=26, Mark=1]6. D

[id=31, Mark=1]7. A

[id=36, Mark=1]8. C

[id=41, Mark=1]9. C

[id=46, Mark=1]10. C

[id=51, Mark=1]11. B

[id=56, Mark=1]12. D

[id=61, Mark=1]13. A

[id=66, Mark=1]14. B

[id=72, Mark=1]15. C

[id=77, Mark=1]16. B

[id=82, Mark=1]17. B

[id=87, Mark=1]18. C

[id=92, Mark=1]19. B

[id=97, Mark=1]20. C

[id=102, Mark=1]21. D

[id=107, Mark=1]22. B

[id=112, Mark=1]23. D

[id=117, Mark=1]24. A

[id=122, Mark=1]25. A

[id=127, Mark=1]26. A

[id=132, Mark=1]27. D

[id=137, Mark=1]28. A

[id=142, Mark=1]29. A

[id=147, Mark=1]30. C

[id=152, Mark=1]31. B

[id=157, Mark=1]32. A

[id=162, Mark=1]33. D

[id=167, Mark=1]34. C

[id=172, Mark=1]35. D

[id=177, Mark=1]36. A

[id=183, Mark=1]37. C

[id=188, Mark=1]38. C

[id=193, Mark=1]39. C

[id=198, Mark=1]40. B

[id=203, Mark=1]41. C

[id=209, Mark=1]42. B

[id=214, Mark=1]43. C

[id=219, Mark=1]44. A

[id=225, Mark=1]45. A

[id=231, Mark=1]46. F

[id=238, Mark=1]47. C

[id=243, Mark=1]48. A

[id=248, Mark=1]49. D

[id=253, Mark=1]50. C

[id=259, Mark=1]51. A

[id=264, Mark=1]52. A

[id=269, Mark=1]53. C

[id=274, Mark=1]54. B

[id=279, Mark=1]55. D

[id=284, Mark=1]56. C

[id=289, Mark=1]57. D

[id=294, Mark=1]58. A

[id=299, Mark=1]59. A

[id=304, Mark=1]60. A

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*