System Development Technology

Md. Mohsin Uddin

East West University

mmuddin@ewubd.edu

March 24, 2023

The Course Outline

Course Outline

- **Course Title:** Preparation course for FE examination
- **Intended Participants:** University Students who are going to take ITPEC examinations
- **Course Duration:** 60 hours

The Lecture Plan

Lecture Plan: Morning Exam, Sec 4-Development Technology, Chapter 1-System Development Technology

Time	Learning Points/Keywords	Explanation Points	Method	Level	Note
10 minutes	System requirements definition	System performance requirements (response time, throughput), System functional specifications	Verbal Explanation	Low	
		Database requirements, Security requirements, Migration requirements, Test requirements			
		Operational requirements, Maintenance requirements			
		Failure handling, Education, Training, Cost, Execution environment requirements			
		Peripheral interface requirements, Quality requirements			
10 minutes	Systems architecture design	Hardware configuration, Software configuration, Manual operation, Configuration item	Verbal Explanation	Low	
		User work scope, Hardware architecture, Software architecture, Application architecture			
		Database architecture, System integration test requirements, Evaluation of systems architecture			
20 minutes	Software requirements definition	Software configuration item, Interface design, Screen design, Security implementation method	TextBook Vol. 2	Medium	
		Business operations modeling, Form design, Code design, Slip design, Data modeling	Page No. 286		
		Maintainability, Evaluation of software requirements, Hearing, Use case, Prototype, DFD (activity, data store			
		data flow, process), E-R diagram, UML (use case diagram, class diagram			
		operation, attribute, role name, sequence diagram, communication diagram (collaboration diagram)			
		state machine diagram (statechart diagram))			
		Context diagram, Decision table, Mini spec, State transition diagram			

Lecture Plan: Morning Exam, Sec 4-Development Technology, Chapter 1-System Development Technology

10 minutes	Software architecture design and software detailed design	Software architecture design tasks (Software structure, Software component (program) partitioning,		**High**	
		Architecture design of components			
		Component-to-component interface design, Decision of component functional specification			
		Structuring, partitioning, and reuse of components, I/O design, Integration test requirements, Checklist)			
		Software detailed design tasks (Software component unit, Functional hierarchy diagram			
10 minutes	Software coding and testing	Coding, Programming languages, Algorithm, Data processing, Traceability, External consistency	Verbal Explanation	Medium	
		Internal consistency, Coverage of unit test, Appropriateness of coding techniques and conventions			
		Feasibility of software integration and tests, Feasibility of operations and maintenance, Code inspection			
		Debugging environment, Static analysis, Dynamic testing, Assertion, Debugger, Test data generator			
		Test design and management technique, Bug curve, Error removal			
		Bug control chart, Coverage, Experimental design, Test case, White box test			
		Statement coverage, Condition coverage, Decision condition coverage, Multiple-condition coverage,			
		Black box test, Boundary value analysis, Equivalence partitioning,			
		Cause-effect graph method, Error embedding method			

Objectives

1.1 System requirements definition

- Understand the outline of system requirements definition.

Keywords

1.1 System requirements definition

- System performance requirements (response time, throughput), System functional specifications, Database requirements, Security requirements, Migration requirements, Test requirements, Operational requirements, Maintenance requirements, Failure handling, Education, Training, Cost, Execution environment requirements, Peripheral interface requirements, Quality requirements

Objectives

1.2 Systems architecture design

- Understand the outline of systems architecture design.

Keywords

1.2 Systems architecture design

- Hardware configuration, Software configuration, Manual operation, Configuration item, User work scope, Hardware architecture, Software architecture, Application architecture, Database architecture, System integration test requirements, Evaluation of systems architecture

Objectives

1.3 Software requirements definition**

- Understand the techniques required for software requirements definition, and apply them to associated matters.

1.3 Software requirements definition**

- Software configuration item, Interface design, Screen design, Security implementation method, Business operations modeling, Form design, Code design, Slip design, Data modeling, Maintainability, Evaluation of software requirements, Hearing, Use case, Prototype, DFD (activity, data store, data flow, process), E-R diagram, UML (use case diagram, class diagram, operation, attribute, role name, sequence diagram, communication diagram (collaboration diagram), state machine diagram (statechart diagram)), Context diagram, Decision table, Mini spec, State transition diagram

Objectives

1.4 Software architecture design and software detailed design***

- Understand the techniques required for software architecture design, and apply them to the associated matters.
- Learn the techniques required for software detailed design, and apply them.

1.4 Software architecture design and software detailed design***

- Software architecture design tasks (Software structure, Software component (program) partitioning, Architecture design of components, Component-to-component interface design, Decision of component functional specification, Structuring, partitioning, and reuse of components, I/O design, Integration test requirements, Checklist) Software detailed design tasks (Software component unit, Functional hierarchy diagram

Objectives

1.5 Software coding and testing**

- Learn the techniques required for software coding and testing, and apply them.

1.5 Software coding and testing**

- Coding, Programming languages, Algorithm, Data processing, Traceability, External consistency, Internal consistency , Coverage of unit test, Appropriateness of coding techniques and conventions, Feasibility of software integration and tests, Feasibility of operations and maintenance, Code inspection, Debugging environment, Static analysis, Dynamic testing, Assertion, Debugger, Test data generator, Test design and management technique, Bug curve, Error removal, Bug control chart, Coverage, Experimental design, Test case, White box test, Statement coverage, Condition coverage, Decision condition coverage, Multiple-condition coverage, Black box test, Boundary value analysis, Equivalence partitioning, Cause-effect graph method, Error embedding method

Objectives

1.6 Software integration and software qualification tests

- Learn and apply the basic concept, procedures, and techniques of software integration and software qualification tests.

Keywords

1.6 Software integration and software qualification tests

- Test requirements, Test procedure, Test data, Software requirements, Audit, Test plan, Test preparation (test environment, test data, etc.), Software integration test report, Top-down testing, Bottom-up testing, Driver, Stub

Objectives

1.7 System integration and system qualification tests

- Understand the techniques required for software integration and system qualification tests, and apply them to the associated matters.

Keywords

1.7 System integration and system qualification tests

- Hardware configuration item, Software configuration item, Manual operation, System requirements, Test plan, Test preparation (test environment, test data, etc.)

Objectives

1.8 Software installation

- Understand the outline of software installation.

Keywords

1.8 Software installation

- Software installation requirements, System migration requirements, Criteria for determining whether or not software can be installed, Creation of installation plan, Installation operation, Software installation procedure, Software system installation framework, User department, System operations department

Objectives

1.9 Software acceptance

- Understand the outline of software acceptance.

Keywords

1.9 Software acceptance

- Delivery, Acceptance procedure, Acceptance criteria, Acceptance test execution, Receiving inspection, Receiving inspection criteria, Preparedness for acceptance, Operation manual, Operations regulations

Objectives

1.10 Software maintenance

- Understand the basic concept, style, and procedure of software maintenance, and apply them to the associated matters.

1.10 Software maintenance

- Maintenance procedure, Maintenance framework, Feasibility of maintenance, Maintenance test, Regression test, Hardware maintenance, Scheduled maintenance, Preventive maintenance, On-site maintenance, Remote maintenance, Life cycle evaluation, Change procedure establishment, Preparation of maintenance documentation, Analyses of problems and change requests, Reproduction or verification of problems, Function addition, Performance improvement, Correction of problems, Planning and implementation of the migration, Notification to the users, Parallel operations of the old and new environments, Verification of the migration, Evaluation of the migration, Disposal planning, Parallel operations of the old and new software products, Disposal notification, Data integrity

Analyzation

Analyzation

- Analyzed 28 questions
- Covered most recent years
  - 2021 Q1 Exam
  - 2021 Q2 Exam

Questions

Question 1

Q1. (q4-1) Which of the following should be clearly specified as a software requirement?

a. Disk capacity of the development environment
b. Objectives of computerization
c. Data definition
d. Highest level design of databases

Q1. (q4-1) Which of the following should be clearly specified as a software requirement?

Theme: System Development Technology, Category: FE

a. Disk capacity of the development environment
b. Objectives of computerization
c. Data definition
d. Highest level design of databases

Question 1: Answer Explanation

Software requirements (also called functional requirements) are for the fulfillment of business requirements. In other words, software requirements are for the implementation of business processes in a business system. The definition of data is an important software requirement. Therefore, c) is the correct answer.

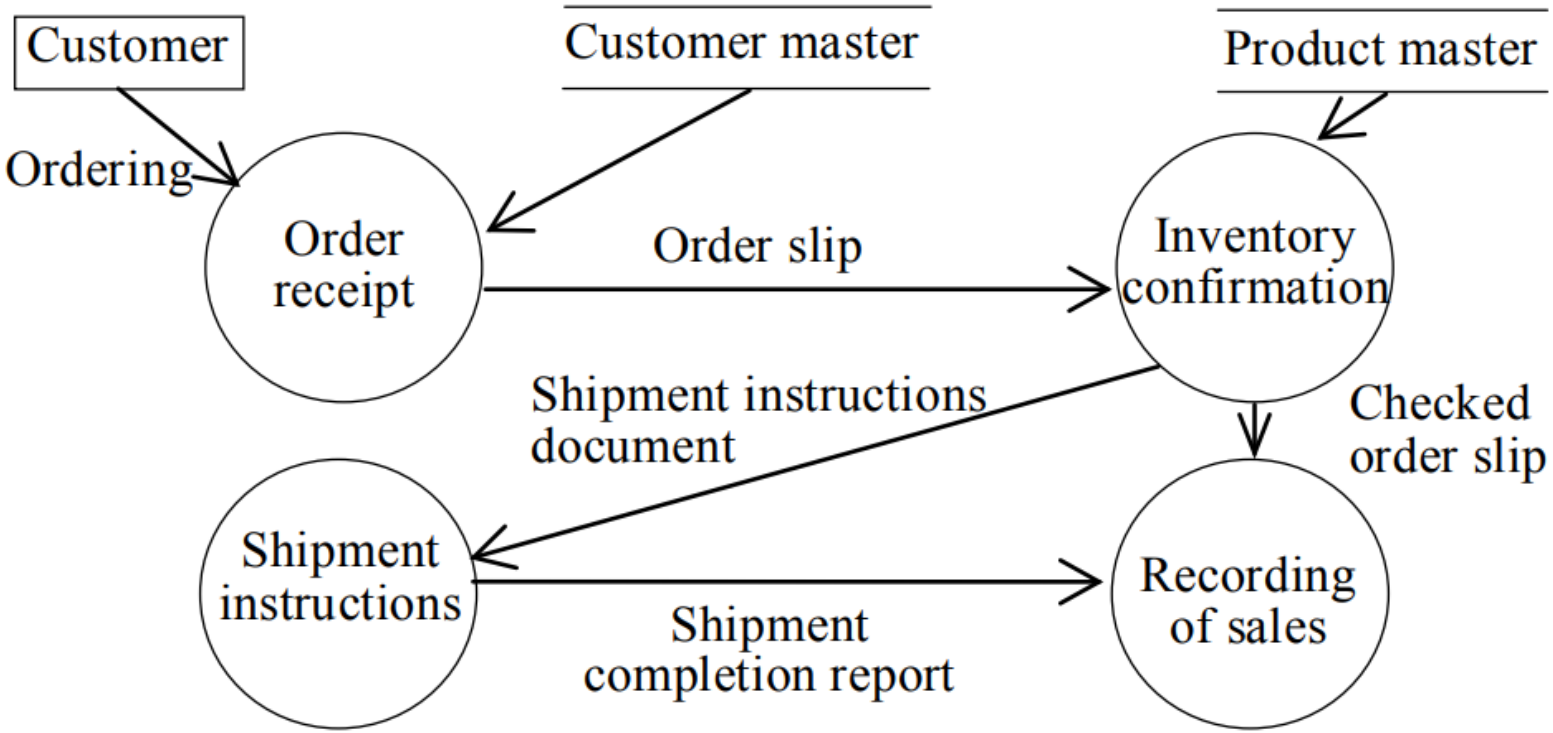
a) The disk capacity of the development environment is a necessary system requirement for system operation. It is not a software requirement, and is defined as a non-functional requirement.

b) The objectives of computerization are specified at the planning phase of the system.

d) Database design is performed at the external design phase.

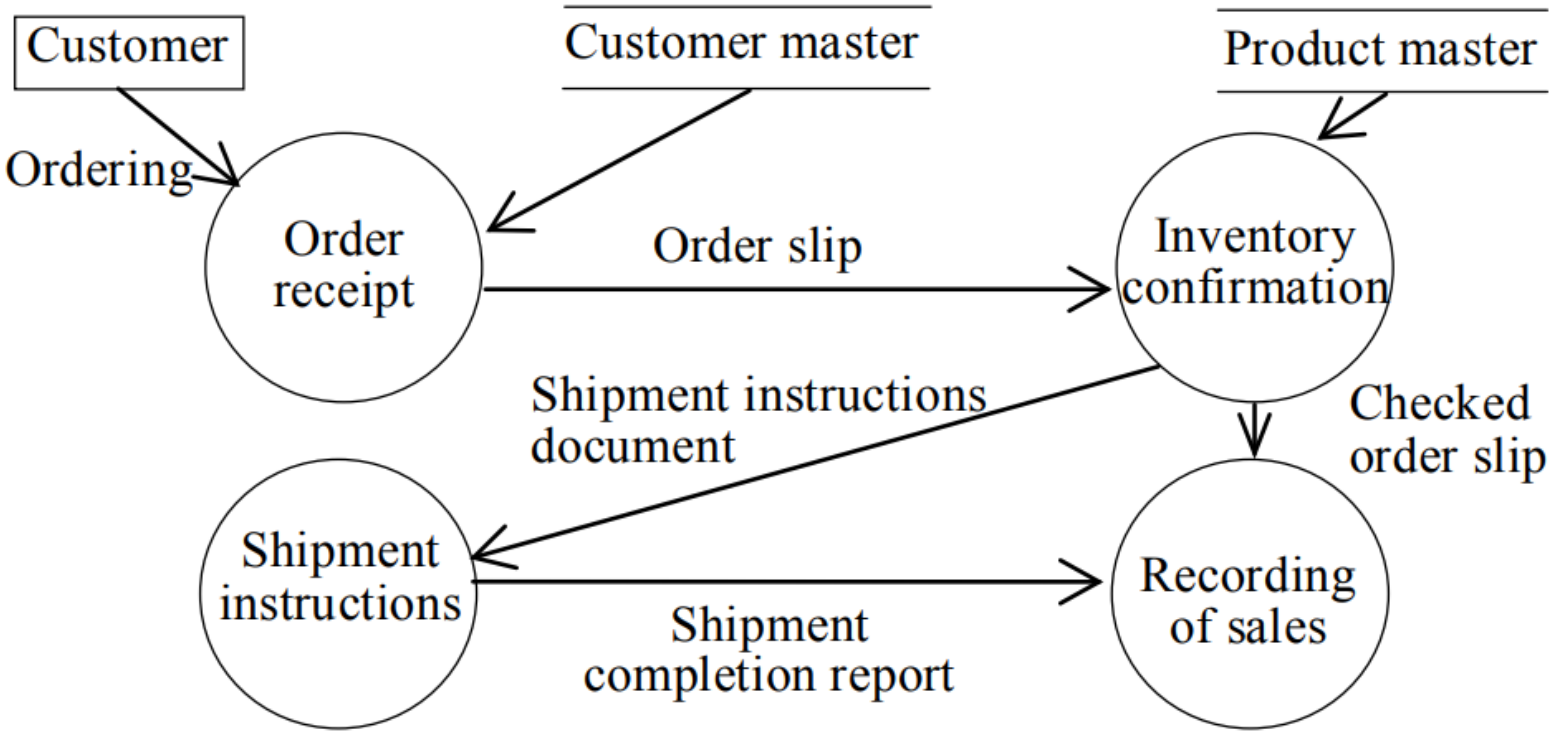
Question 2

Q2. (q4-2) Which of the following is the notation used in the diagram below?



a. DFD
b. State transition diagram
c. Flowchart
d. Petri net

Q2. (q4-2) Which of the following is the notation used in the diagram below?



Theme: System Development Technology, Category: FE

a. DFD
b. State transition diagram
c. Flowchart
d. Petri net

Question 3

Q3. (q4-3) Which of the following is the most appropriate description concerning code design for a business system?

a. Actual assignment of code should be performed by a system designer who is proficient in methods for code processing.

b. The attributes and number of digits in the code should be designed with importance placed on the internal processing efficiency of the computer.

c. When mistakes in code input are judged to have a significant effect on business, check characters, such as check digits, should be implemented.

d. Methods for code maintenance, such as additions, deletions, and changes, should be decided at the operational test phase.

Q3. (q4-3) Which of the following is the most appropriate description concerning code design for a business system?
Theme: System Development Technology, Category: FE

a. Actual assignment of code should be performed by a system designer who is proficient in methods for code processing.
b. The attributes and number of digits in the code should be designed with importance placed on the internal processing efficiency of the computer.
c. When mistakes in code input are judged to have a significant effect on business, check characters, such as check digits, should be implemented.
d. Methods for code maintenance, such as additions, deletions, and changes, should be decided at the operational test phase.

Question 3: Answer Explanation

Appending a check digit to code allows the strengthening of code input checking. Therefore, c) is an appropriate statement.
a) Actual assignment of code is generally performed by system users.
b) The attributes and number of digits in a code must be designed with consideration of not only the internal processing efficiency of the computer, but also other factors, such as ease of use from a user perspective, commonality, systematization, and expandability.
d) It is difficult to change the scheme of code once the code begins to be used. Therefore, it is necessary to fully consider maintainability and expandability at the time of design.

Question 4

Q4. (q4-4) Which of the following is UML that is standardized by the OMG?

a. A modeling language that is used in software development and has an object-oriented approach
b. An interface description language for the use of an object from another program
c. A manipulation language of a relational database for operations such as table definition and data manipulation
d. A markup language for describing the structure or meaning of a statement or data

Q4. (q4-4) Which of the following is UML that is standardized by the OMG?

Theme: System Development Technology, Category: FE

a. A modeling language that is used in software development and has an object-oriented approach

b. An interface description language for the use of an object from another program

c. A manipulation language of a relational database for operations such as table definition and data manipulation

d. A markup language for describing the structure or meaning of a statement or data

Question 4: Answer Explanation

UML (Unified Modeling Language) is a modeling language that is based on an object-oriented approach. It is used in software development and was standardized by a body called the OMG (Object Management Group). Therefore, a) is the correct answer.
In order to describe the details of analysis and design, UML has various diagrams such as use case diagrams, class diagrams, sequence diagrams, and state chart diagrams.
b) This is a description concerning IDL (Interface Definition Language).
c) This is a description concerning SQL (Structured Query Language).
d) This is a description concerning SGML (Standard Generalized Markup Language).

Question 5

Q5. (q4-5) A use case is a technique that clearly describes the functional requirements of a system and defines interactions between a user and the system. Which of the following is an appropriate scenario for modeling with use cases?

a. The planning of a new IT service that uses the Internet
b. Construction work on the air-conditioning equipment in a computer room
c. Use of a parallel processor to reduce processing time
d. The withdrawal of cash by a depositor from an ATM

Q5. (q4-5) A use case is a technique that clearly describes the functional requirements of a system and defines interactions between a user and the system. Which of the following is an appropriate scenario for modeling with use cases?

Theme: System Development Technology, Category: FE

a. The planning of a new IT service that uses the Internet
b. Construction work on the air-conditioning equipment in a computer room
c. Use of a parallel processor to reduce processing time
d. The withdrawal of cash by a depositor from an ATM

Question 5: Answer Explanation

Use cases are used for the specification of the functional requirements of a system, and if the option that is related to the function of a system is chosen, the obvious choice is d).
On the other hand, a) and b) are not implemented with a function of an information system, and c) is a processing method. Therefore, none of these choices are appropriate.

Question 6

Q6. (q4-6) Which of the following is an explanation of encapsulation in object-oriented design?

a. The abstraction and organization of multiple objects with similar characteristics
b. The inheritance of the characteristics of a base class by a subclass
c. The abstraction of characteristics that are common between classes, and the creation of a base class
d. The unification of data and the procedures that manipulate this data, and its concealment inside an object

Q6. (q4-6) Which of the following is an explanation of encapsulation in object-oriented design?

Theme: System Development Technology, Category: FE

a. The abstraction and organization of multiple objects with similar characteristics
b. The inheritance of the characteristics of a base class by a subclass
c. The abstraction of characteristics that are common between classes, and the creation of a base class
d. The unification of data and the procedures that manipulate this data, and its concealment inside an object

Question 7

Q7. (q4-7) In order to increase the independence of modules, it is necessary to weaken module coupling. Which of the following is the method for transferring information that has the weakest module coupling?

a. Data defined in a common area is referenced by the relevant modules.
b. Control parameters are passed as arguments, and the execution sequence of modules is controlled.
c. Only data items are passed as arguments between modules.
d. Required data is shared through external declaration.

Q7. (q4-7) In order to increase the independence of modules, it is necessary to weaken module coupling. Which of the following is the method for transferring information that has the weakest module coupling?
Theme: System Development Technology, Category: FE

a. Data defined in a common area is referenced by the relevant modules.
b. Control parameters are passed as arguments, and the execution sequence of modules is controlled.
c. Only data items are passed as arguments between modules.
d. Required data is shared through external declaration.

Question 7: Answer Explanation: Slide I

Module coupling is a measure of interdependence between modules, and it is categorized into six types. Generally, the weaker the coupling, the better it is considered to be. Arranging in the sequence from strong to weak gives the following result. "No direct coupling" (7) means that there is no interdependence between modules, and it is the ideal form of relationship. However, this is very rare in practice, and therefore coupling is usually categorized into the following six types from (1) to (6).

- (1) Content coupling
- (2) Common coupling
- (3) External coupling
- (4) Control coupling
- (5) Stamp coupling
- (6) Data coupling
- (7) No direct coupling

Question 7: Answer Explanation: Slide II

Among these, in "data coupling" (6), which has the weakest module coupling, "only data items are passed as arguments between modules," so the correct answer is c).
In "stamp coupling" (5), when the pointer of a structure is passed as an argument, the receiving side will become aware of this data structure and use the required data.
a) "Data defined in a common area is referenced by the relevant modules" corresponds to "common coupling" (2).
b) "Control parameters are passed as arguments, and the execution sequence of modules is controlled" corresponds to "control coupling" (4). In this form, awareness of the internal logic of other modules is necessary, so module coupling strengthens.
d) "Required data is shared through external declaration" corresponds to "external coupling" (3).

Question 8

Q8. (q4-8) Which one of the following has the highest module strength?

a. A module that executes multiple functions in sequence for certain data
b. A module that processes data from different input media
c. A module that executes a single function
d. A module that contains all tasks considered necessary at a specific point in time

Q8. (q4-8) Which one of the following has the highest module strength?
Theme: System Development Technology, Category: FE

a. A module that executes multiple functions in sequence for certain data
b. A module that processes data from different input media
c. A module that executes a single function
d. A module that contains all tasks considered necessary at a specific point in time

Question 8: Answer Explanation: Slide I

Module strength is a measure of the strength of the connection between instructions in a module, and the stronger the connection (high strength), the better it is considered to be.
"Functional strength" is required for a module to perform its unique function. In other words, all instructions are required to perform this one function. As the module with the highest "functional strength" is considered to be the best, c) is the correct answer.

As mentioned above, module strength is a measure of strength of the connection between instructions in a module. The objective of considering this measure is to increase the maintainability of a module. When one module has multiple functions, and if the connection between these functions is not strong, maintenance of the module becomes difficult (for example, the modification of one instruction may affect multiple functions). Therefore, modules that have a "single function" (functional

Question 8: Answer Explanation: Slide II

strength) with a high strength are considered to be better. With regard to other options in the answer group,
a) includes multiple functions for identical data, so it is called "communicational strength." As for d), it includes multiple functions performed simultaneously at a specific point in time, and it is called "classical strength."
Moreover, with regard to b), it is difficult to determine the strength from the description. However, data processing from different media means that the module has multiple functions, and therefore it cannot be said that its strength is higher than module c), which has a single function.
Incidentally, in addition to three types of strengths (functional strength, communicational strength, and classical strength) described above, there are four other types of module strengths. Each of them corresponds to the following modules.

Question 8: Answer Explanation: Slide III

- Coincidental strength: The function of a module cannot be defined. A large program is simply partitioned into modules with number of lines or another parameter as a guide.
- Logical strength: A module has multiple functions, and a function is executed in a selective manner according to the details of the function parameters.
- Procedural strength: A module that contains multiple functions defined in functional specifications.
- Informational strength: A module into which multiple functions that handle the same data are combined, and which has separate entry point names for each function (module name is separate for each function). These seven strengths can be arranged in the decreasing order of strength (strong, desirable) as follows: functional strength, informational strength, communicational strength, procedural strength, classical strength, logical strength, and coincidental

Question 8: Answer Explanation: Slide IV

strength. Therefore, module partitioning that results in modules with functional strength and informational strength is desired.

Question 9

Q9. (q4-9) Which of the following is a method where the author and multiple involved persons review a design document at the completion of the design for the early detection of design errors?

a. Walkthrough
b. Desk checking
c. Top-down testing
d. Parallel simulation

Q9. (q4-9) Which of the following is a method where the author and multiple involved persons review a design document at the completion of the design for the early detection of design errors?
Theme: System Development Technology, Category: FE

a. Walkthrough
b. Desk checking
c. Top-down testing
d. Parallel simulation

Question 9: Answer Explanation: Slide I

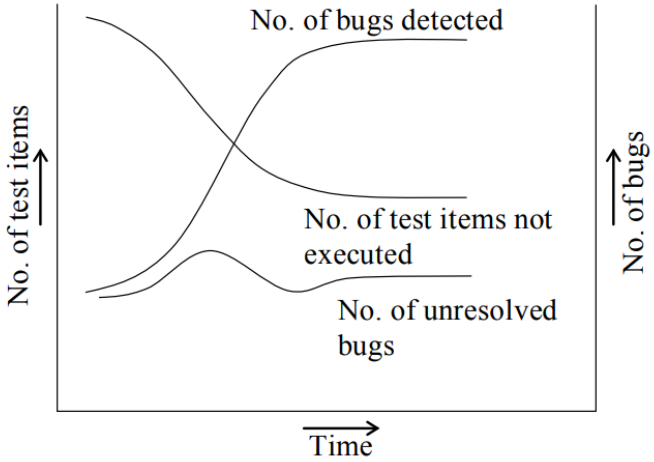
In system development, work is divided and assigned to multiple people, and these people work in parallel. If work is performed by the person in-charge only, assumptions or mistakes may cause incorrect results or deliverables. In order to prevent this, upon completion of each process, the person in-charge and other relevant persons get together and conduct a review to check the content of the work. A series of activities conducted by relevant persons reviewing the document or work in order to detect design errors as early as possible is called a) walkthrough. Furthermore, a review conducted under the guidance of a dedicated facilitator (called a moderator) to identify and analyze problems and consider measures for such problems is called an inspection.
b) Desk checking: A debugging method where the developer himself traces instructions one by one with the assumed test data without running the program. Errors may be overlooked due to perceived notions.

Question 9: Answer Explanation: Slide II

c) Top-down testing: A method for an integration test, where modules are integrated and tested in sequence from the upper level to the lower level.
d) Parallel simulation: In a system audit conducted using a computer, an auditor prepares a test program for the audit with respect to a specific application, and performs verifications by comparing the simulation results obtained using the actual input data with the actual results generated by the production program.

Question 10

Q10. (q4-10) In a bug control chart, if all the lines level out as shown in the figure below, which of the following is an appropriate assumption that can be made?



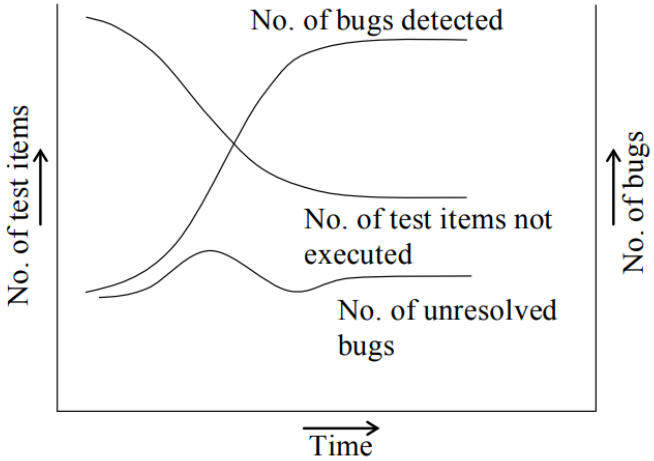
a. Due to a bug that is difficult to resolve, subsequent testing is not progressing.

b. Many bugs were detected, so the number of test items conducted is not increasing.

c. The detection of bugs is halted, and the tests are close to being completed.

d. The number of bugs detected is equal to that of test items conducted, and the number of unresolved bugs remains unchanged.

Q10. (q4-10) In a bug control chart, if all the lines level out as shown in the figure below, which of the following is an appropriate assumption that can be made?



Theme: System Development Technology, Category: FE

a. Due to a bug that is difficult to resolve, subsequent testing is not progressing.
b. Many bugs were detected, so the number of test items conducted is not increasing.
c. The detection of bugs is halted, and the tests are close to being completed.
d. The number of bugs detected is equal to that of test items conducted, and the number of unresolved bugs remains unchanged.

Question 10: Answer Explanation: Slide I

As all three curves in the bug control chart given in the question remained at the same levels, the chart indicates that all three activities of test execution (number of test items not executed), bug detection (number of bugs detected), and bug resolution (number of unresolved bugs) have halted, and the test is not progressing. There could be multiple reasons for this, such as all members of the test team caught a cold or went on a company outing.
However, if a selection is made from the answer group, as mentioned in a), it can be thought that the test team have discovered a bug that is difficult to resolve. For example, a functional bug that is related to all test cases not yet executed (remaining) has occurred, and unless this bug is resolved, testing cannot move forward, which leads to the situation described in the question. In the typical form of the software reliability growth curve, the rate of bug detection is low immediately after the test starts. This is

Question 10: Answer Explanation: Slide II

usually because tests have come across a critical bug that affects other tests. As is the case in this question, testing cannot move forward.
b) Rather than the detection of many bugs, the reason that the execution of test items is not processed further could be that the number of unresolved bugs is not decreasing.
c) The number of test items not executed is flat but it is not approaching zero. Therefore, it cannot be said that test is close to being completed.
d) The fact that the number of bugs detected becomes equal to that of test items executed has no relation with the resolution of bugs.

Question 11

Q11. (q4-11) Which of the following is an appropriate description of the black box test?

a. The coverage ratio of test cases is used as a standard for the preparation of test data.

b. Even if there is redundant code in the test program, it cannot be detected.

c. Whether the required portion is executed or not is verified with the focus on the internal structure of a program.

d. An increase in the number of branch instructions or modules results in a sharp increase in test data.

Q11. (q4-11) Which of the following is an appropriate description of the black box test?

Theme: System Development Technology, Category: FE

a. The coverage ratio of test cases is used as a standard for the preparation of test data.

b. Even if there is redundant code in the test program, it cannot be detected.

c. Whether the required portion is executed or not is verified with the focus on the internal structure of a program.

d. An increase in the number of branch instructions or modules results in a sharp increase in test data.

Question 11: Answer Explanation

The method for the design of test cases is divided into a white box test and a black box test.
A black box test is a test to check whether the program satisfies the functions in the design document without focusing on the internal logic of the program. It can also be said that this test is conducted from the viewpoint of a user. Equivalence partitioning and boundary value analysis are used for preparing test data. In this case, since focus is not on the internal logic of the program, even if there is unnecessary description (redundant code) in the program, it cannot be detected unless processing results are abnormal. Therefore, b) is the appropriate description.
On the other hand, a white box test is a test conducted based on the internal specifications of a program, and test data is prepared in consideration of the coverage of instructions, conditional decisions/branches, path combinations, and other factors. Description of a), c), and d) corresponds to the white box test.

Question 13

Q13. (q4-13) When modifications are made to a part of a system, which of the following is a test that verifies the modified part is not adversely affected and the correct result is obtained?

a. Function test
b. Integration test
c. Regression test
d. Exception test

Q13. (q4-13) When modifications are made to a part of a system, which of the following is a test that verifies the modified part is not adversely affected and the correct result is obtained?
Theme: System Development Technology, Category: FE

a. Function test
b. Integration test
c. Regression test
d. Exception test

Question 13: Answer Explanation: Slide I

In system modification during the maintenance phase, modifications that are made often adversely affect other functioning sections, and functions that were working normally before modification stop working correctly. Therefore, for the testing conducted after system modification, in addition to testing the modified section, it is necessary to check that functions that were working normally before the modification still work correctly. A test that verifies that such modifications do not lead to new errors in functions that were working correctly is called c) regression test.

- a) A function test is a test for verification that a system satisfies all the functions required by its specifications.
- b) An integration test is conducted after a unit test of each module is completed, and it verifies that modules are processed correctly when they are integrated.

Question 13: Answer Explanation: Slide II

- d) An exception test is a test for verification that when data other than the predefined data is processed, such data is processed as per specifications without abnormal termination.

Question 14

Q14. (q4-14) Which of the following is a test that is conducted to verify interfaces between modules and between subsystems?

a. Operational test
b. Integration test
c. System test
d. Unit test

Q14. (q4-14) Which of the following is a test that is conducted to verify interfaces between modules and between subsystems?
Theme: System Development Technology, Category: FE

a. Operational test
b. Integration test
c. System test
d. Unit test

Question 15

Q15. (q4-15) An integration test of software that is composed of module groups with a hierarchical structuring is conducted from higher level modules. In such a test, which of the following is the test module that is used as a substitute for lower level modules?

a. Emulator
b. Simulator
c. Stub
d. Driver

Q15. (q4-15) An integration test of software that is composed of module groups with a hierarchical structuring is conducted from higher level modules. In such a test, which of the following is the test module that is used as a substitute for lower level modules?
Theme: System Development Technology, Category: FE

a. Emulator
b. Simulator
c. Stub
d. Driver

Question 16

Q16. (q4-16) Which of the following is a test that is conducted in the system test phase when a new system is developed?

a. Load test
b. Interface test between modules
c. Operation check test based on module specifications
d. Regression test

Q16. (q4-16) Which of the following is a test that is conducted in the system test phase when a new system is developed?
Theme: System Development Technology, Category: FE

a. Load test
b. Interface test between modules
c. Operation check test based on module specifications
d. Regression test

Question 16: Answer Explanation

In software development based on the waterfall model, testing is conducted in the sequence of unit (module) test, integration test, system test, and operational test.

- a) Load test is conducted in the system test phase. In a load test, a load that is equivalent to or higher than the data processing volume at the peak time of operation is applied in order to verify that a system operates normally.
- b) Conducted in the integration test phase.
- c) Conducted in the unit test phase.
- d) A regression test is a test for checking whether modification of a program adversely affects any other parts than the corresponding function, and it is conducted in the maintenance phase.

Question 17

Q17. (q4-17) Which of the following is the test data that should be prepared when a system qualification test is conducted?

a. Data that is actually used in operations and data that is processed as an exception during operations
b. Data to detect errors related to interfaces between software units
c. Data that passes through all branches of a software unit once or more
d. Data that executes all instructions inside a software unit once or more

Q17. (q4-17) Which of the following is the test data that should be prepared when a system qualification test is conducted?
Theme: System Development Technology, Category: FE

a. Data that is actually used in operations and data that is processed as an exception during operations
b. Data to detect errors related to interfaces between software units
c. Data that passes through all branches of a software unit once or more
d. Data that executes all instructions inside a software unit once or more

Question 17: Answer Explanation: Slide I

A system qualification test, sometimes simply called as a system test, is conducted after a system developer has conducted a unit test and an integration test.
Therefore, when it is required, users also participate in this test to check that the entire system is working as per the specifications.

- A system qualification test includes a function test to check whether all the required functions are included or not, a performance test to check the processing capacity (throughput) and response time, an exception processing test for error and data failure, and a load test to check that the system can handle large volume of data processing. The data used in actual operations and the data handled by exception processing are also required, so a) is the correct answer. In addition, a system qualification test also includes an operability test to check usability for users. Also, the term "system qualification test" is one of

Question 17: Answer Explanation: Slide II

the processes of Common Frame 2007 (SLCP-JCF2007), and was previously called a "system test."

- b) A test concerning the interface between software units is called an integration test, and is conducted before a system test. A system is tested for errors using data not defined in the interfaces between software units.
- c) and d) Data that covers all branches and executes all instructions within a software unit one or more times is used in a unit test conducted as per the white box test method. A unit test verifies whether each software unit satisfies the relevant functions in the program specifications, and it checks if there are any logical errors inside a software unit. Moreover, in the terminology of testing techniques, c) is the test data that is used in a decision condition (or branch) coverage, while d) is the test data that is used in statement coverage.

Question 18

Q18. (q4-18) Which of the following is the primary objective for a migration test of a system?

a. To check the procedure for the switch over from the existing system to the new system and the problems that accompany the changeover, from the viewpoint of safety and efficiency
b. By using a copy of database of the existing system, to check that sufficient performance is obtained in the new system as well
c. To check the consistency of interfaces between the existing program and the newly developed program
d. To check that the new system satisfies all the required functions

Q18. (q4-18) Which of the following is the primary objective for a migration test of a system?
Theme: System Development Technology, Category: FE

a. To check the procedure for the switch over from the existing system to the new system and the problems that accompany the changeover, from the viewpoint of safety and efficiency
b. By using a copy of database of the existing system, to check that sufficient performance is obtained in the new system as well
c. To check the consistency of interfaces between the existing program and the newly developed program
d. To check that the new system satisfies all the required functions

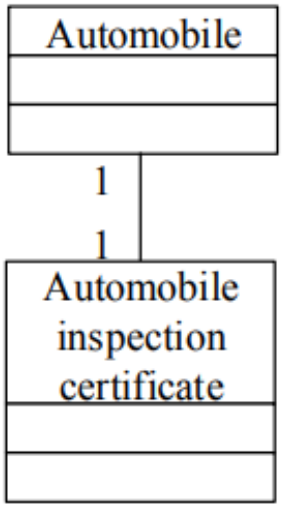
Question 19: Answer Explanation

A migration test is a test to check that after a new system is developed, changeover (migration) from the old system to the new system can be performed smoothly. A migration test is not a test of the new system itself. Since it is also conducted to check the process for system changeover in advance from the viewpoint of safety and efficiency, a) is the correct answer. b), c), and d) are correct in terms of description concerning the test and performance verification conducted prior to migration test. However, these are not the main objectives of migration test.

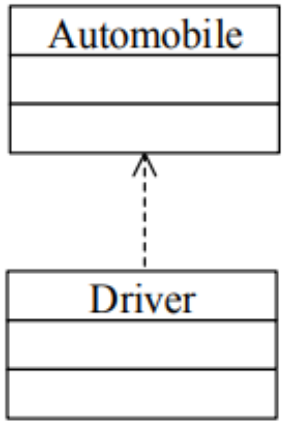
Question 19

Q19. (2021 S FE AM-q45) Which of the following is the UML class diagram that shows a generalization relation?

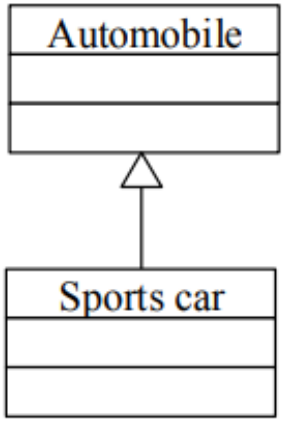
a)



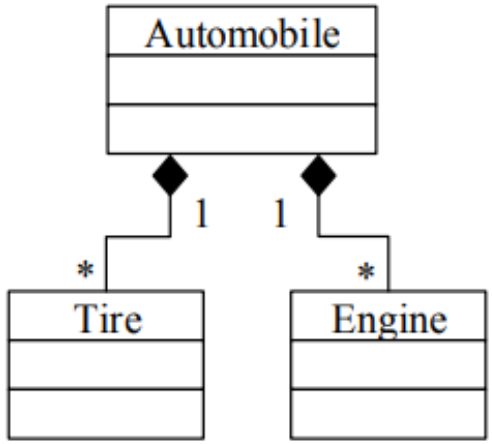
b)



c)

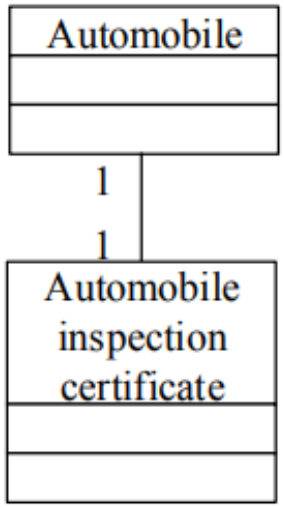


d)

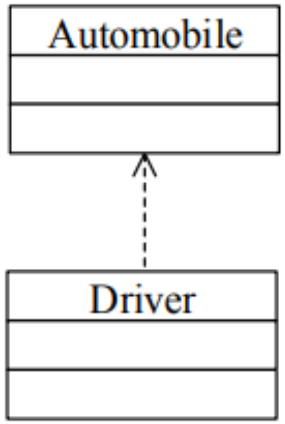


Q19. (2021 S FE AM-q45) Which of the following is the UML class diagram that shows a generalization relation?

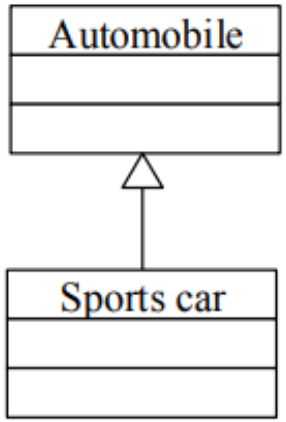
a)



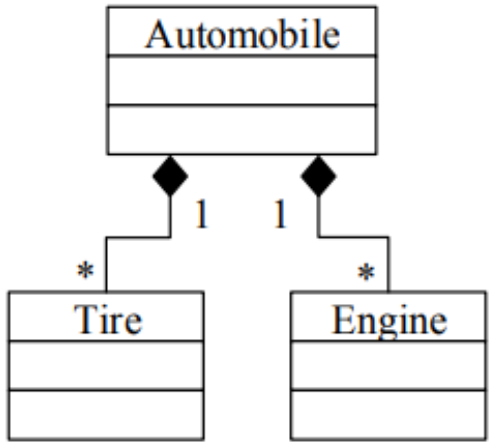
b)



c)



d)

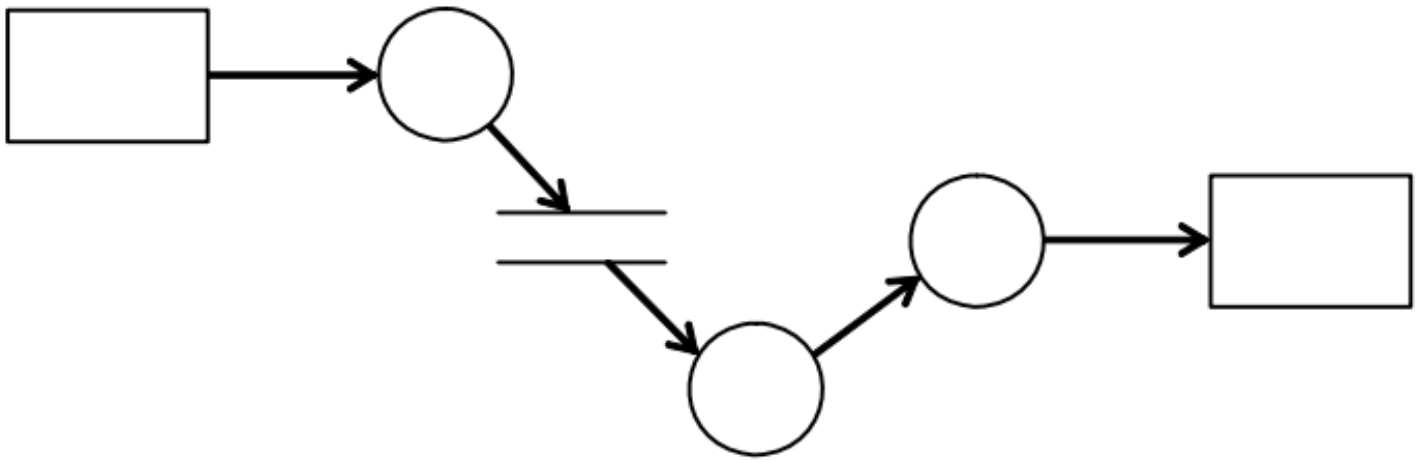


Theme: System Development Technology, Category: FE

- Option C

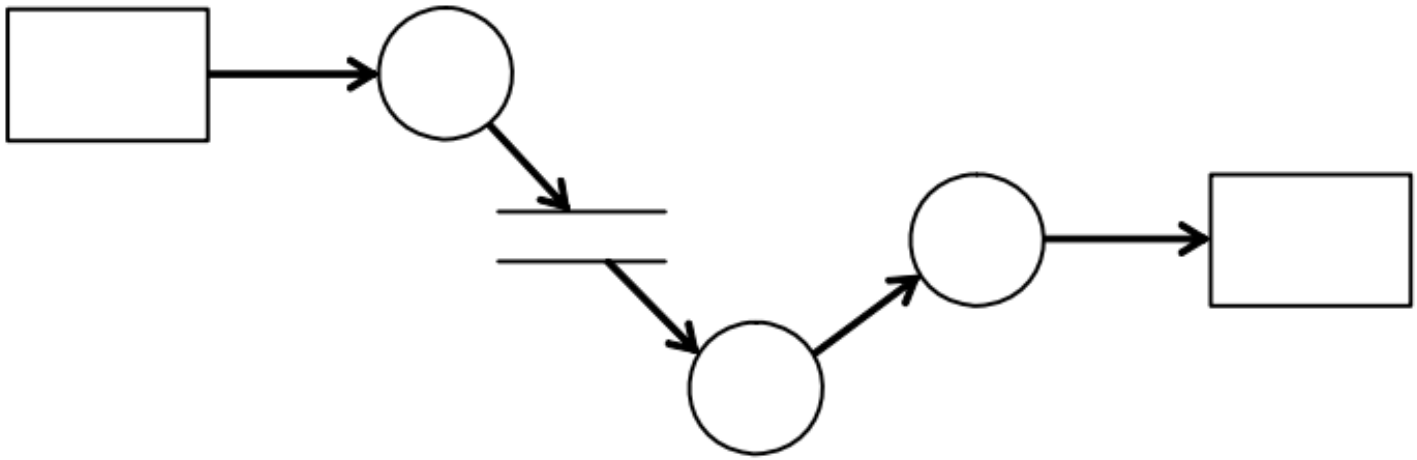
Question 20

Q20. (2021 S FE AM-q46) The figure below shows an example of a DFD that uses one of the DFD notations and represents a flow of data in a system. Texts in the figure are not shown. Which of the following is represented by the circle "O" in the DFD notation used in the example?



a. Activity
b. Data flow
c. Data store
d. Process

Q20. (2021 S FE AM-q46) The figure below shows an example of a DFD that uses one of the DFD notations and represents a flow of data in a system. Texts in the figure are not shown. Which of the following is represented by the circle "O" in the DFD notation used in the example?



Theme: System Development Technology, Category: FE

a. Activity
b. Data flow
c. Data store
d. Process

Question 21

Q21. (2021 S FE AM-q47) Which of the following is an appropriate relationship between classes and instances in object orientation?

a. An instance defines the specifications of a class.
b. An instance is created based on the definition of a class.
c. Multiple classes correspond to a single instance.
d. Only one (1) instance exists for a single class.

Q21. (2021 S FE AM-q47) Which of the following is an appropriate relationship between classes and instances in object orientation?
Theme: System Development Technology, Category: FE

a. An instance defines the specifications of a class.
b. An instance is created based on the definition of a class.
c. Multiple classes correspond to a single instance.
d. Only one (1) instance exists for a single class.

Question 22

Q22. (2021 S FE AM-q48) Which of the following is the weakest module coupling?

a. To implement as many functions as possible with a single module
b. To pass arguments that control another module's logic when the module is called
c. To pass only required data items as arguments between two (2) modules
d. To use a global area to share data items with other modules

Q22. (2021 S FE AM-q48) Which of the following is the weakest module coupling?

Theme: System Development Technology, Category: FE

a. To implement as many functions as possible with a single module
b. To pass arguments that control another module's logic when the module is called
c. To pass only required data items as arguments between two (2) modules
d. To use a global area to share data items with other modules

Question 23

Q23. (2021 S FE AM-q49) Which of the following tests verifies the interfaces between two (2) modules or subsystems?

a. Acceptance test
b. Integration test
c. Operational test
d. Unit test

Q23. (2021 S FE AM-q49) Which of the following tests verifies the interfaces between two (2) modules or subsystems?
Theme: System Development Technology, Category: FE

a. Acceptance test
b. Integration test
c. Operational test
d. Unit test

Question 24

Q24. (2021 A FE AM-q45) Which of the following UML diagrams is used to represent the user interactions with the system and the functions of the system?

a. Activity diagram
b. Class diagram
c. Sequence diagram
d. Use case diagram

Q24. (2021 A FE AM-q45) Which of the following UML diagrams is used to represent the user interactions with the system and the functions of the system?

Theme: System Development Technology, Category: FE

a. Activity diagram
b. Class diagram
c. Sequence diagram
d. Use case diagram

Question 25

Q25. (2021 A FE AM-q46) Which of the following is an appropriate description of an acceptance test?

a. It is conducted by developers to verify response time and other performance items.

b. It is conducted by testers to ensure that the interfaces and linkages between different software parts work properly.

c. It is conducted by the project manager to verify whether users' functional requirements are met or not.

d. It is conducted by users to confirm that the software is complete and meets the business needs that prompted the software to be developed.

Q25. (2021 A FE AM-q46) Which of the following is an appropriate description of an acceptance test?
Theme: System Development Technology, Category: FE

a. It is conducted by developers to verify response time and other performance items.
b. It is conducted by testers to ensure that the interfaces and linkages between different software parts work properly.
c. It is conducted by the project manager to verify whether users' functional requirements are met or not.
d. It is conducted by users to confirm that the software is complete and meets the business needs that prompted the software to be developed.

Question 26

Q26. (2021 A FE AM-q47) In the development process, which of the following is an activity that should be performed in software architectural design?

a. Analyzing the requirements in incremental steps by arranging them in the form of a chart
b. Describing the specifications so that the program is clarified on a line-by-line basis
c. Obtaining the opinion of the customer and deciding the specifications
d. Transforming the requirements for the software into a design that describes its top-level structure and identifies the software components

Q26. (2021 A FE AM-q47) In the development process, which of the following is an activity that should be performed in software architectural design?

Theme: System Development Technology, Category: FE

a. Analyzing the requirements in incremental steps by arranging them in the form of a chart
b. Describing the specifications so that the program is clarified on a line-by-line basis
c. Obtaining the opinion of the customer and deciding the specifications
d. Transforming the requirements for the software into a design that describes its top-level structure and identifies the software components

Question 27

Q27. (2021 A FE AM-q48) Which of the following is an appropriate description concerning test data for black box testing?

a. Test data is created based on branch coverage.
b. Test data is created based on condition coverage.
c. Test data is created based on external specifications.
d. Test data is created based on program structures.

Q27. (2021 A FE AM-q48) Which of the following is an appropriate description concerning test data for black box testing?
Theme: System Development Technology, Category: FE

a. Test data is created based on branch coverage.
b. Test data is created based on condition coverage.
c. Test data is created based on external specifications.
d. Test data is created based on program structures.

Question 28

Q28. (2021 A FE AM-q49) In an agile software development project, when can the customer review the first piece of working software?

a. After the completion of the first epic
b. After the completion of the first few releases
c. After the completion of the first iteration
d. After the completion of the whole project

Q28. (2021 A FE AM-q49) In an agile software development project, when can the customer review the first piece of working software?
Theme: System Development Technology, Category: FE

a. After the completion of the first epic
b. After the completion of the first few releases
c. After the completion of the first iteration
d. After the completion of the whole project

Question 29

Q29. (2021 A FE AM-q50) According to the Capability Maturity Model Integration, which of the following is the final and highest level of maturity?

a. Auditable
b. Custom
c. Optimizing
d. Quantitatively Managed

Q29. (2021 A FE AM-q50) According to the Capability Maturity Model Integration, which of the following is the final and highest level of maturity?

Theme: System Development Technology, Category: FE

a. Auditable
b. Custom
c. Optimizing
d. Quantitatively Managed

Any Questions?

References

 IT Fundamentals (New FE Textbook Vol. 2)