**Review 1**

1. Which one of the following is not a step of requirement engineering?
   1. elicitation
   2. design
   3. analysis
   4. documentation
2. Which of the following statements explains portability in non-functional requirements?
   1. It is a degree to which software running on one platform can easily be converted to run on another platform
   2. It cannot be enhanced by using languages, OS’ and tools that are universally available and standardized
   3. The ability of the system to behave consistently in a user-acceptable manner when operating within the environment for which the system was intended
   4. None of the mentioned
3. Which of the following property does not correspond to a good Software Requirements Specification (SRS) ?
   1. Verifiable
   2. Ambiguous
   3. Complete
   4. Traceable
4. Why is Requirements Management Important ? It is due to the changes
   1. to the environment
   2. in technology
   3. in customer’s expectations
   4. in all of the mentioned.
5. Which perspective in system modelling shows the system or data architecture.
   1. Structural perspective
   2. Behavioral perspective
   3. External perspective
   4. All of the mentioned
6. Which one of the following models is not suitable for accommodating any change?
   1. Build & Fix Model
   2. Prototyping Model
   3. RAD Model
   4. Waterfall Model
7. Which model can be selected if user is involved in all the phases of SDLC?
   1. Waterfall Model
   2. Prototyping Model
   3. RAD Model
   4. both Prototyping Model & RAD Model
8. What is the major advantage of using Incremental Model?
   1. Customer can respond to each increment
   2. Easier to test and debug
   3. It is used when there is a need to get a product to the market early
   4. Easier to test and debug & It is used when there is a need to get a product to the market early
9. How is Incremental Model different from Spiral Model?
   1. Progress can be measured for Incremental Model
   2. Changing requirements can be accommodated in Incremental Model
   3. Users can see the system early in Incremental Model
   4. All of the mentioned
10. Agile Software Development is based on
    1. Incremental Development
    2. Iterative Development
    3. Linear Development
    4. Both Incremental and Iterative Development
11. Debugging is:
    1. creating program code
    2. finding and correcting errors in the program code
    3. identifying the task to be computerized
    4. creating the algorithm
12. A software engineer must design the modules with the goal of high cohesion and low coupling.
    1. True
    2. False
13. Function-oriented design techniques starts with functional requirements specified in
    1. SDD
    2. SRS
    3. All of the mentioned
    4. None of the mentioned
14. Choose the incorrect statement in terms of Objects.
    1. Objects are abstractions of real-world
    2. Objects can’t manage themselves
    3. Objects encapsulate state and representation information
    4. All of the mentioned
15. Which of the following describes”Is-a-Relationship” ?
    1. Aggregation
    2. Inheritance
    3. Dependency
    4. All of the mentioned
16. A design description of an object is known as a class
    1. instance
    2. object
    3. case
    4. both instance and object
17. Which of the following is not project management goal?
    1. Keeping overall costs within budget
    2. Delivering the software to the customer at the agreed time
    3. Maintaining a happy and well-functioning development team
    4. Avoiding customer complaints
18. Which of the following is an important factor that can affect the accuracy and efficacy of estimates?
    1. Project size
    2. Planning process
    3. Project complexity
    4. Degree of structural uncertainty
19. Software Maintenance includes
    1. Error corrections
    2. Enhancements of capabilities
    3. Deletion of obsolete capabilities
    4. All of the mentioned
20. What type of software testing is generally used in Software Maintenance?
    1. Regression Testing
    2. System Testing
    3. Integration Testing
    4. Unit Testing
21. The UML was designed for describing
    1. object-oriented systems
    2. architectural design
    3. SRS
    4. Both object-oriented systems and Architectural design
22. Which of the following pattern is the basis of interaction management in many web-based systems?
    1. architecture
    2. repository pattern
    3. model-view-controller
    4. different operating system
23. Which of the following is/are commonly used architectural pattern(s)?
    1. Model-View-Controller
    2. Layered Architecture
    3. Client–server
    4. All of the mentioned
24. Which of the following is golden rule for interface design?
    1. Place the user in control
    2. Reduce the user’s memory load
    3. Make the interface consistent
    4. All of the mentioned
25. Which of the following option is not considered by the Interface design?
    1. the design of interfaces between software components
    2. the design of interfaces between the software and human producers and consumers of information
    3. the design of the interface between two computers
    4. all of the mentioned
26. Which pattern prevents one from creating more than one instance of a variable?
    1. Factory Method
    2. Singleton
    3. Observer
    4. None of the mentioned
27. Service Oriented Architecture (SOA) is
    1. Strongly Coupled
    2. Loosely Coupled
    3. Strongly Cohesive
    4. Loosely Cohesive
28. White Box techniques are also classified as
    1. Design based testing
    2. Structural testing
    3. Error guessing technique
    4. None of the mentioned
29. Alpha testing is done at
    1. Developer’s end
    2. User’s end
    3. Developer’s & User’s end
    4. None of the mentioned
30. Which of the following is non-functional testing?
    1. Black box testing
    2. Performance testing
    3. Unit testing
    4. None of the mentioned

**Software Engineering Questions and Answers – Evolutionary Software Process Models**

1. Which one of the following is not an Evolutionary Process Model?

a) WINWIN Spiral Model

b) Incremental Model

c) Concurrent Development Model

d) All of the mentioned

1. The Incremental Model is a result of combination of elements of which two models?

a) Build & FIX Model & Waterfall Model

b) Linear Model & RAD Model

c) Linear Model & Prototyping Model

d) Waterfall Model & RAD Model

1. What is the major advantage of using Incremental Model?

a) Customer can respond to each increment

b) Easier to test and debug

c) It is used when there is a need to get a product to the market early

d) Easier to test and debug & It is used when there is a need to get a product to the market early

1. The spiral model was originally proposed by

a) IBM

b) Barry Boehm

c) Pressman

d) Royce

1. The spiral model has two dimensions namely \_\_\_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_

a) diagonal, angular

b) radial, perpendicular

c) radial, angular

d) diagonal, perpendicularThe spiral model was originally proposed by

1. How is WINWIN Spiral Model different from Spiral Model?

a) It defines tasks required to define resources, timelines, and other project related information

b) It defines a set of negotiation activities at the beginning of each pass around the spiral

c) It defines tasks required to assess both technical and management risks

d) It defines tasks required to construct, test, install, and provide user supportThe spiral model was originally proposed by

1. Identify the disadvantage of Spiral Model.

a) Doesn’t work well for smaller projects

b) High amount of risk analysis

c) Strong approval and documentation control

d) Additional Functionality can be added at a later date

1. Spiral Model has user involvement in all its phases.

a) True

b) False

1. How is Incremental Model different from Spiral Model?

a) Progress can be measured for Incremental Model

b) Changing requirements can be accommodated in Incremental Model

c) Users can see the system early in Incremental Model

d) All of the mentioned

1. If you were to create client/server applications, which model would you go for?

a) WINWIN Spiral Model

b) Spiral Model

c) Concurrent Model

d) Incremental Model

**Software Engineering Questions and Answers – User Interface Design**

1. Which of the following is golden rule for interface design?

a) Place the user in control

b) Reduce the user’s memory load

c) Make the interface consistent

d) All of the mentioned

1. Which of the following is not a design principle that allow the user to maintain control?

a) Provide for flexible interaction

b) Allow user interaction to be interrupt-able and undo-able

c) Show technical internals from the casual user

d) Design for direct interaction with objects that appear on the screen

1. Which of the following is not a user interface design process?

a) User, task, and environment analysis and modeling

b) Interface design

c) Knowledgeable, frequent users

d) Interface validation

1. When users are involved in complex tasks, the demand on \_\_\_\_\_\_\_\_\_\_ can be significant.

a) short-term memory

b) shortcuts

c) objects that appear on the screen

d) all of the mentioned

1. Which of the following option is not considered by the Interface design?

a) the design of interfaces between software components

b) the design of interfaces between the software and human producers and consumers of information

c) the design of the interface between two computers

d) all of the mentioned

1. A software might allow a user to interact via

a) keyboard commands

b) mouse movement

c) voice recognition commands

d) all of the mentioned

1. A software engineer designs the user interface by applying an iterative process that draws on predefined design principles.

a) True

b) False

1. What incorporates data, architectural, interface, and procedural representations of the software?

a) design model

b) user’s model

c) mental image

d) system image

1. What establishes the profile of end-users of the system?

a) design model

b) user’s model

c) mental image

d) system image

1. What combines the outward manifestation of the computer-based system , coupled with all supporting information that describe system syntax and semantics?

a) mental image

b) interface design

c) system image

d) interface validation

**Software Engineering Questions and Answers – Requirement Management**

1. Which two requirements are given priority during Requirement Management of a product ?

a) User and Developer

b) Functional and Non-functional

c) Enduring and Volatile

d) All of the mentioned

1. Considering the example of issue/return of a book, cataloging etc. in a library management.What type of management requirement is being depicted here?

a) Enduring

b) Volatile

c) Both Enduring & Volatile

d) All of the mentioned

1. Why is Requirements Management Important ? It is due to the changes

a) to the environment

b) in technology

c) in customer’s expectations

d) in all of the mentioned.

1. Requirements Management is a prerequisite for Quality-Oriented Development.

a) True

b) False

1. Requirements traceability is one of the most important part requirement management. It may also be referred to as the heart of requirement management.

a) True

b) False

1. Requirements Management has a high initial start-up cost but does not need ongoing funding throughout a project.

a) True

b) False

1. Which of the following is not a Requirement Management workbench tool ?

a) RTM

b) DOORS

c) Rational Suite

d) RDD 100

1. Which of the following is a requirement management activity ?

a) Investigation

b) Design

c) Construction and Test

d) All of the mentioned

1. What functionality of Requirement Management Tool (RMT) is depicted by the statement: “the tool should be able to automatically detect relations between artifacts. For example information retrieval techniques, monitoring of change history, naming schemas or model transformations.”

a) Automatic Link Detection

b) Documentation Support

c) Graphical Representation

d) Automatic Link Creation and Change

1. According to a statistical report: “over 30% of all software projects are cancelled before completion and over 70% of the remainder fail to deliver expected features”. What must be the reason for such a situation ?

a) Poor change management

b) Poor requirements management

c) Poor quality control

d) All of the mentioned

**Software Engineering Questions and Answers – Requirement Engineering**

1. What are the types of requirements ?
   1. Availability
   2. Reliability
   3. Usability
   4. All of the mentioned
2. Select the developer-specific requirement ?
   1. Portability
   2. Maintainability
   3. Availability
   4. Both Portability and Maintainability
3. Which one of the following is not a step of requirement engineering?
   1. elicitation
   2. design
   3. analysis
   4. documentation
4. FAST stands for
   1. Functional Application Specification Technique
   2. Fast Application Specification Technique
   3. Facilitated Application Specification Technique
   4. None of the mentioned
5. QFD stands for
   1. quality function design
   2. quality function development
   3. quality function deployment
   4. none of the mentioned
6. A Use-case actor is always a person having a role that different people may play.
   1. True
   2. False
7. The user system requirements are the parts of which document ?
   1. SDD
   2. SRS
   3. DDD
   4. SRD
8. A stakeholder is anyone who will purchase the completed software system under development.
   1. True
   2. False
9. Conflicting requirements are common in Requirement Engineering, with each client proposing his or her version is the right one.
   1. True
   2. False
10. Which is one of the most important stakeholder from the following ?
    1. Entry level personnel
    2. Middle level stakeholder
    3. Managers
    4. Users of the software

**Software Engineering Questions and Answers – Architectural Design**

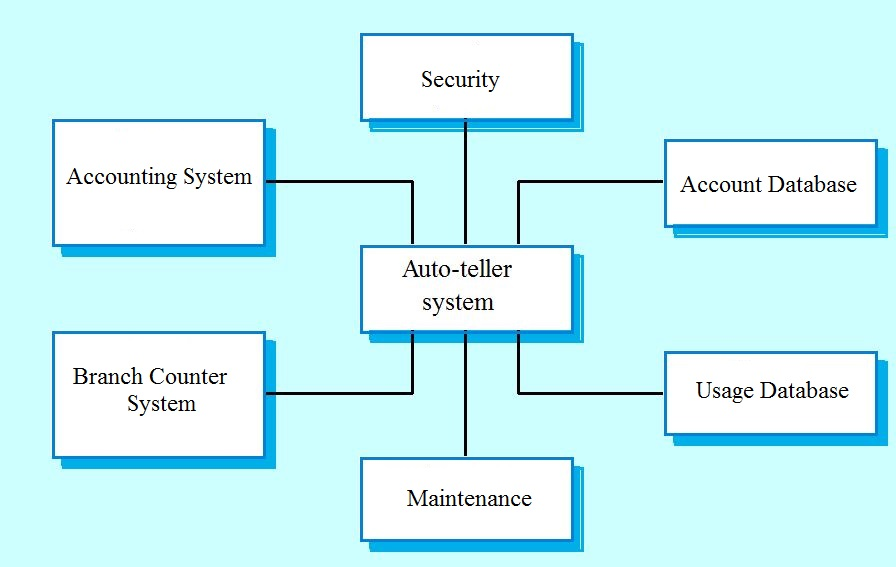
1. Architectural design is a creative process satisfying only functional-requirements of a system.
   1. True
   2. False
2. A \_\_\_\_\_\_\_\_ view shows the system hardware and how software components are distributed across the processors in the system.
   1. physical
   2. logical
   3. process
   4. all of the mentioned
3. The UML was designed for describing \_\_\_\_\_\_\_\_\_
   1. object-oriented systems
   2. architectural design
   3. SRS
   4. Both object-oriented systems and Architectural design
4. Which of the following view shows that the system is composed of interacting processes at run time?
   1. physical
   2. development
   3. logical
   4. process
5. Which of the following is an architectural conflict?
   1. Using large-grain components improves performance but reduces maintainability
   2. Introducing redundant data improves availability but makes security more difficult
   3. Localizing safety-related features usually means more communication so degraded performance
   4. All of the mentioned
6. Which of the following is not included in Architectural design decisions?
   1. type of application
   2. distribution of the system
   3. architectural styles
   4. testing the system
7. Architecture once established can be applied to other products as well.
   1. True
   2. False
8. Which of the following pattern is the basis of interaction management in many web-based systems?
   1. architecture
   2. repository pattern
   3. model-view-controller
   4. different operating system
9. What describes how a set of interacting components can share data?
   1. model-view-controller
   2. architecture pattern
   3. repository pattern
   4. none of the mentioned
10. Which view in architectural design shows the key abstractions in the system as objects or object classes?
    1. physical
    2. development
    3. logical
    4. process
11. Which of the following is a type of Architectural Model?
    1. Static structural model
    2. Dynamic process model
    3. Distribution model
    4. All of the mentioned

**Software Engineering Questions and Answers – Software Testing Techniques – 1**

1. Which of the following term describes testing?
   1. Finding broken code
   2. Evaluating deliverable to find errors
   3. A stage of all projects
   4. None of the mentioned
2. What is Cyclomatic complexity?
   1. Black box testing
   2. White box testing
   3. Yellow box testing
   4. Green box testing
3. Lower and upper limits are present in which chart?
   1. Run chart
   2. Bar chart
   3. Control chart
   4. None of the mentioned
4. Maintenance testing is performed using which methodology?
   1. Retesting
   2. Sanity testing
   3. Breadth test and depth test
   4. Confirmation testing
5. White Box techniques are also classified as
   1. Design based testing
   2. Structural testing
   3. Error guessing technique
   4. None of the mentioned
6. Exhaustive testing is
   1. always possible
   2. practically possible
   3. impractical but possible
   4. impractical and impossible
7. Which of the following is/are White box technique?
   1. Statement Testing
   2. Decision Testing
   3. Condition Coverage
   4. All of the mentioned
8. What are the various Testing Levels?
   1. Unit Testing
   2. System Testing
   3. Integration Testing
   4. All of the mentioned
9. Boundary value analysis belong to?
   1. White Box Testing
   2. Black Box Testing
   3. White Box & Black Box Testing
   4. None of the mentioned
10. Alpha testing is done at
    1. Developer’s end
    2. User’s end
    3. Developer’s & User’s end
    4. None of the mentioned

**Software Engineering Questions and Answers – System Modelling – 1**

1. The Unified Modeling Language (UML) has become an effective standard for software modelling.How many different notations does it have ?
   1. Three
   2. Four
   3. Six
   4. Nine
2. Which model in system modelling depicts the dynamic behaviour of the system ?
   1. Context Model
   2. Behavioral Model
   3. Data Model
   4. Object Model
3. Which model in system modelling depicts the static nature of the system ?
   1. Behavioral Model
   2. Context Model
   3. Data Model
   4. Structural Model
4. Which perspective in system modelling shows the system or data architecture.
   1. Structural perspective
   2. Behavioral perspective
   3. External perspective
   4. All of the mentioned
5. Which system model is being depicted by the ATM operations shown below:



* 1. Structural model
  2. Context model
  3. Behavioral model
  4. Interaction model

1. Activity diagrams are used to model the processing of data.
   1. True
   2. False
2. Model-driven engineering is just a theoretical concept. It cannot be converted into a working/executable code.
   1. True
   2. False
3. The UML supports event-based modeling using \_\_\_\_\_\_\_\_\_\_\_\_ diagrams.
   1. Deployment
   2. Collaboration
   3. State chart
   4. All of the mentioned

**Software Engineering Questions and Answers – Object Oriented Software Design – 1**

1. Choose the incorrect statement in terms of Objects.
   1. Objects are abstractions of real-world
   2. Objects can’t manage themselves
   3. Objects encapsulate state and representation information
   4. All of the mentioned
2. What encapsulates both data and data manipulation functions ?
   1. Object
   2. Class
   3. Super Class
   4. Sub Class
3. Which of the following is a mechanism that allows several objects in an class hierarchy to have different methods with the same name?
   1. Aggregation
   2. Polymorphism
   3. Inheritance
   4. All of the mentioned
4. Inherited object classes are self-contained.
   1. True
   2. False
5. Which of the following points related to Object-oriented development (OOD) is true?
   1. OOA is concerned with developing an object model of the application domain
   2. OOD is concerned with developing an object-oriented system model to implement requirements
   3. All of the mentioned
   4. None of the mentioned
6. How is generalization implemented in Object Oriented programming languages?
   1. Inheritance
   2. Polymorphism
   3. Encapsulation
   4. Abstract Classes
7. 7. Which of the following is a disadvantage of OOD ?
   1. Easier maintenance
   2. Objects may be   
      understood as stand-alone entities
   3. Objects are potentially reusable components
   4. None of the mentioned
8. Which of the following describes”Is-a-Relationship” ?
   1. Aggregation
   2. Inheritance
   3. Dependency
   4. All of the mentioned
9. Object that collects data on request rather than autonomously is known as
   1. Active Object
   2. Passive Object
   3. Multiple instance
   4. None of the mentioned
10. Objects are executed
    1. sequentially
    2. in Parallel
    3. sequentially & Parallel
    4. none of the mentioned

**Software Engineering Questions and Answers – Software Maintenance – 1**

1. Software Maintenance includes
   1. Error corrections
   2. Enhancements of capabilities
   3. Deletion of obsolete capabilities
   4. All of the mentioned
2. Maintenance is classified into how many categories ?
   1. two
   2. three
   3. four
   4. five
3. The modification of the software to match changes in the ever changing environment, falls under which category of software maintenance?
   1. Corrective
   2. Adaptive
   3. Perfective
   4. Preventive
4. How many phases are there in Taute Maintenance Model?
   1. six
   2. seven
   3. eight
   4. nine
5. What type of software testing is generally used in Software Maintenance?
   1. Regression Testing
   2. System Testing
   3. Integration Testing
   4. Unit Testing
6. Regression testing is a very expensive activity.
   1. True
   2. False
7. Selective retest techniques may be more economical than the “retest-all”technique.How many selective retest techniques are there?
   1. two
   2. three
   3. four
   4. five
8. Which selective retest technique selects every test case that causes a modified program to produce a different output than its original version?
   1. Coverage
   2. Minimization
   3. Safe
   4. Maximization
9. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ measures the ability of a regression test selection technique to handle realistic applications.
   1. Efficiency
   2. Precision
   3. Generality
   4. Inclusiveness
10. Which regression test selection technique exposes faults caused by modifications?
    1. Efficiency
    2. Precision
    3. Generality
    4. Inclusiveness

**Software Engineering Questions and Answers – Software Life Cycle Models**

1. Build & Fix Model is suitable for programming exercises of \_\_\_\_\_\_\_\_\_\_\_ LOC (Line of Code).
   1. 100-200
   2. 200-400
   3. 400-1000
   4. above 1000
2. RAD stands for
   1. Relative Application Development
   2. Rapid Application Development
   3. Rapid Application Document
   4. None of the mentioned
3. Which one of the following models is not suitable for accommodating any change?
   1. Build & Fix Model
   2. Prototyping Model
   3. RAD Model
   4. Waterfall Model
4. Which is not one of the types of prototype of Prototyping Model?
   1. Horizontal Prototype
   2. Vertical Prototype
   3. Diagonal Prototype
   4. Domain Prototype
5. Which one of the following is not a phase of Prototyping Model?
   1. Quick Design
   2. Coding
   3. Prototype Refinement
   4. Engineer Product
6. Which of the following statements regarding Build & Fix Model is wrong?
   1. No room for structured design
   2. Code soon becomes unfixable & unchangeable
   3. Maintenance is practically not possible
   4. It scales up well to large projects
7. RAD Model has
   1. 2 phases
   2. 3 phase
   3. 5 phases
   4. 6 phases
8. What is the major drawback of using RAD Model?
   1. Highly specialized & skilled developers/designers are required
   2. Increases reusability of components
   3. Encourages customer/client feedback
   4. Increases reusability of components, Highly specialized & skilled developers/designers are required
9. SDLC stands for
   1. Software Development Life Cycle
   2. System Development Life cycle
   3. Software Design Life Cycle
   4. System Design Life Cycle
10. Which model can be selected if user is involved in all the phases of SDLC?
    1. Waterfall Model
    2. Prototyping Model
    3. RAD Model
    4. both Prototyping Model & RAD Model

**Software Engineering Questions and Answers – Functional and Non-Functional Requirements**

1. Which one of the following is a functional requirement ?
   1. Maintainability
   2. Portability
   3. Robustness
   4. none of the mentioned
2. Which one of the following is a requirement that fits in a developer’s module ?
   1. Availability
   2. Testability
   3. Usability
   4. Flexibility
3. “Consider a system where, a heat sensor detects an intrusion and alerts the security company.” What kind of a requirement the system is providing ?
   1. Functional
   2. Non functional
   3. Know requirement
   4. None of the mentioned
4. Which of the following statements explains portability in non-functional requirements?
   1. It is a degree to which software running on one platform can easily be converted to run on another platform
   2. It cannot be enhanced by using languages, OS’ and tools that are universally available and standardized
   3. The ability of the system to behave consistently in a user-acceptable manner when operating within the environment for which the system was intended
   4. None of the mentioned
5. Functional requirements capture the intended behavior of the system.
   1. true
   2. false
6. Choose the incorrect statement with respect to Non-Functional Requirement(NFR)
   1. Product-oriented Approach – Focus on system (or software) quality
   2. Process-oriented Approach – Focus on how NFRs can be used in the design process
   3. Quantitative Approach – Find measurable scales for the functionality attributes
   4. Qualitative Approach – Study various relationships between quality goals
7. How many classification schemes have been developed for NFRs ?
   1. 2
   2. 3
   3. 4
   4. 5
8. According to components of FURPS+, which of the following does not belong to S ?
   1. Testability
   2. Speed Efficiency
   3. Serviceability
   4. Installability
9. Does software wear & tear by decomposition ?
   1. Yes
   2. No
10. What are the four dimensions of Dependability ?
    1. Usability, Reliability, Security, Flexibility
    2. Availability, Reliability, Maintainability, Security
    3. Availability, Reliability, Security, Safety
    4. Security, Safety, Testability, Usability

**Software Engineering Questions and Answers – Modularity in Software Design**

1. Java packages and Fortran subroutine are examples of\_\_\_\_\_\_\_\_\_\_
   1. Functions
   2. Modules
   3. Classes
   4. Sub procedures
2. Which of the property of software modularity is incorrect with respect to benefits software modularity?
   1. Modules are robust
   2. Module can use other modules
   3. Modules Can be separately compiled and stored in a library
   4. Modules are mostly dependent
3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a measure of the degree of interdependence between modules.
   1. Cohesion
   2. Coupling
   3. None of the mentioned
   4. All of the mentioned
4. Which of the following is the best type of module coupling?
   1. Control Coupling
   2. Stamp Coupling
   3. Data Coupling
   4. Content Coupling
5. Which of the following is the worst type of module coupling?
   1. Control Coupling
   2. Stamp Coupling
   3. External Coupling
   4. Content Coupling
6. Which of the following is the worst type of module cohesion?
   1. Logical Cohesion
   2. Temporal Cohesion
   3. Functional Cohesion
   4. Coincidental Cohesion
7. Which of the following is the best type of module cohesion?
   1. Functional Cohesion
   2. Temporal Cohesion
   3. Functional Cohesion
   4. Sequential Cohesion
8. A software engineer must design the modules with the goal of high cohesion and low coupling.
   1. True
   2. False
9. In what type of coupling, the complete data structure is passed from one module to another?
   1. Control Coupling
   2. Stamp Coupling
   3. External Coupling
   4. Content Coupling
10. If all tasks must be executed in the same time-span, what type of cohesion is being exhibited?
    1. Functional Cohesion
    2. Temporal Cohesion
    3. Functional Cohesion
    4. Sequential Cohesion

**Software Engineering Questions and Answers – Software Design Pattern**

1. Which mechanism is applied to use a design pattern in an OO system?
   1. Inheritance
   2. Composition
   3. All of the mentioned
   4. None of the mentioned
2. Design patterns does not follow the concept of software reuse.
   1. True
   2. False
3. The use of design patterns for the development of object-oriented software has important implications for
   1. Component-based software engineering
   2. Reusability in general
   3. All of the mentioned
   4. None of the mentioned
4. Which of the following is a design pattern?
   1. Behavioral
   2. Structural
   3. Abstract Factory
   4. All of the mentioned
5. You want to minimize development cost by reusing methods? Which design pattern would you choose?
   1. Adapter Pattern
   2. Singleton Pattern
   3. Delegation pattern
   4. Immutable Pattern
6. You want to avoid multiple inheritance. Which design pattern would you choose?
   1. Abstraction-Occurrence Pattern
   2. Player-Role Pattern
   3. General Hierarchy Pattern
   4. Singleton Pattern
7. The recurring aspects of designs are called design
   1. patterns
   2. documents
   3. structures
   4. methods
8. Design pattern is a solution to a problem that occurs repeatedly in a variety of contexts.
   1. True
   2. False
9. Which pattern prevents one from creating more than one instance of a variable?
   1. Factory Method
   2. Singleton
   3. Observer
   4. None of the mentioned
10. Facade pattern promotes weak coupling between subsystem and its clients.
    1. True
    2. False
11. Which design pattern defines one-to-many dependency among objects?
    1. Singleton pattern
    2. Facade Pattern
    3. Observer pattern
    4. Factory method pattern
12. Facade pattern couples a subsystem from its clients.
    1. True
    2. False
13. In factory method pattern, the framework must instantiate classes but it only knows about the abstract classes, which it cannot initiate. How would one solve this problem?
    1. encapsulating the knowledge of which document subclass to is to be created and
    2. moving this knowledge out of the framework
    3. instantiating the application specific documents without knowing their class
    4. all of the mentioned

**Software Engineering Questions and Answers – Project Management**

1. Which of the following is not project management goal?
   1. Keeping overall costs within budget
   2. Delivering the software to the customer at the agreed time
   3. Maintaining a happy and well-functioning development team
   4. Avoiding customer complaints
2. Project managers have to assess the risks that may affect a project.
   1. True
   2. False
3. Which of the following is not considered as a risk in project management?
   1. Specification delays
   2. Product competition
   3. Testing
   4. Staff turnover
4. The process each manager follows during the life of a project is known as
   1. Project Management
   2. Manager life cycle
   3. Project Management Life Cycle
   4. All of the mentioned
5. A 66.6% risk is considered as
   1. very low
   2. low
   3. moderate
   4. high

Explanation: The probability of the risk might be assessed as very low (<10%), low (10–25%), moderate (25–50%), high (50–75%), or very high (>75%).

1. Which of the following is/are main parameters that you should use when computing the costs of a software development project?
   1. travel and training costs
   2. hardware and software costs
   3. effort costs (the costs of paying software engineers and managers)
   4. all of the mentioned
2. Quality planning is the process of developing a quality plan for
   1. team
   2. project
   3. customers
   4. project manager
3. Which of the following is incorrect activity for the configuration management of a software system?
   1. Internship management
   2. Change management
   3. Version management
   4. System management
4. Identify the sub-process of process improvement
   1. Process introduction
   2. Process analysis
   3. De-processification
   4. Process distribution
5. An independent relationship must exist between the attribute that can be measured and the external quality attribute.
   1. True
   2. False

**Software Engineering Questions and Answers – Object Oriented Software Design – 2**

1. How many layers are present in the OO design pyramid?
   1. three
   2. four
   3. five
   4. one
2. Which of the following early OOD methods incorporates both a “micro development process” and a “macro development process.” ?
   1. Booch method
   2. Rumbaugh method
   3. Wirfs-Brock method
   4. Coad and Yourdon method
3. Grady Booch, James Rumbaugh, and Ivar Jacobson combined the best features of their individual object-oriented analysis into a new method for object oriented design known as
   1. HTML
   2. XML
   3. UML
   4. SGML
4. A design description of an object is known as a class
   1. instance
   2. object
   3. case
   4. both instance and object
5. Which of the following is conceptually similar to objects?
   1. PACKAGE
   2. PROC
   3. PRIVATE
   4. None of the mentioned
6. A design description in OOD includes
   1. Protocol Description
   2. Implementation Description
   3. Type Description
   4. both Protocol and Implementation Description
7. Which of the following is not an operation as per OOD algorithms and data structures?
   1. operations that manipulate data in some way
   2. operations that perform a computation
   3. operations that check for syntax errors
   4. operations that monitor an object for the occurrence of a controlling event
8. Throughout the OOD process, a software engineer should look for every opportunity for creating new design process.
   1. True
   2. False

**Software Engineering Questions and Answers – Requirement Documentation**

1. Which of the following property does not correspond to a good Software Requirements Specification (SRS) ?  
a) Verifiable  
b) Ambiguous  
c) Complete  
d) Traceable

2. Which of the following property of SRS is depicted by the statement : “Conformity to a standard is maintained” ?  
a) Correct  
b) Complete  
c) Consistent  
d) Modifiable

3. The SRS is said to be consistent if and only if  
a) its structure and style are such that any changes to the requirements can be made easily while retaining the style and structure  
b) every requirement stated therein is one that the software shall meet  
c) every requirement stated therein is verifiable  
d) no subset of individual requirements described in it conflict with each other

4. Which of the following statements about SRS is/are true ?  
i. SRS is written by customer  
ii. SRS is written by a developer  
iii. SRS serves as a contract between customer and developer  
a) Only i is true  
b) Both ii and iii are true  
c) All are true  
d) None of the mentioned

5. The SRS document is also known as \_\_\_\_\_\_\_\_\_\_\_\_\_ specification.  
a) black-box  
b) white-box  
c) grey-box  
d) none of the mentioned

6. Which of the following is included in SRS ?  
a) Cost  
b) Design Constraints  
c) Staffing  
d) Delivery Schedule

7. Which of the following is not included in SRS ?  
a) Performance  
b) Functionality  
c) Design solutions  
d) External Interfaces

8. Arrange the given sequence to form a SRS Prototype outline as per IEEE SRS Standard.  
i. General description  
ii. Introduction  
iii. Index  
iv. Appendices  
v. Specific Requirements  
a) iii, i, ii,v, iv  
b) iii, ii, i, v, iv  
c) ii, i, v, iv, iii  
d) iii, i, ii

9. Consider the following Statement: “The output of a program shall be given within 10 secs of event X 10% of the time.”What characteristic of SRS is being depicted here ?  
a) Consistent  
b) Verifiable  
c) Non-verifiable  
d) Correct

10. Consider the following Statement: “The data set will contain an end of file character.”What characteristic of SRS is being depicted here ?  
a) Consistent  
b) Non-verifiable  
c) Correct  
d) Ambiguous

**Software Engineering Questions and Answers – Agile Software Development**

1.Select the option that suits the Manifesto for Agile Software Development  
a) Individuals and interactions  
b) Working software  
c) Customer collaboration  
d) All of the mentioned

2. Agile Software Development is based on  
a) Incremental Development  
b) Iterative Development  
c) Linear Development  
d) Both Incremental and Iterative Development

3. Which on of the following is not an agile method?  
a) XP  
b) 4GT  
c) AUP  
d) All of the mentioned

4. Agility is defined as the ability of a project team to respond rapidly to a change.  
a) True  
b) False

5. How is plan driven development different from agile development ?  
a) Outputs are decided through a process of negotiation during the software development process  
b) Specification, design, implementation and testing are interleaved  
c) Iteration occurs within activities  
d) All of the mentioned

6. How many phases are there in Scrum ?  
a) Two  
b) Three  
c) Four  
d) Scrum is an agile method which means it does not have phases

7. Agile methods seem to work best when team members have a relatively high skill level.  
a) True  
b) False

8. Which of the following does not apply to agility to a software process?  
a) Uses incremental product delivery strategy  
b) Only essential work products are produced  
c) Eliminate the use of project planning and testing  
d) All of the mentioned

9. Which three framework activities are present in Adaptive Software Development(ASD) ?  
a) analysis, design, coding  
b) requirements gathering, adaptive cycle planning, iterative development  
c) speculation, collaboration, learning  
d) all of the mentioned

10. In agile development it is more important to build software that meets the customers’ needs today than worry about features that might be needed in the future.  
a) True  
b) False

**Software Engineering Questions and Answers - Function Oriented Software Design**

1. Choose the option that does not define Function Oriented Software Design.  
a) It consists of module definitions  
b) Modules represent data abstraction  
c) Modules support functional abstraction  
d) None of the mentioned

2. Which of the following is a complementary approach to function-oriented approach ?  
a) Object oriented analysis  
b) Object oriented design  
c) Structured approach  
d) Both Object oriented analysis and design

3. Function-oriented design techniques starts with functional requirements specified in  
a) SDD  
b) SRS  
c) All of the mentioned  
d) None of the mentioned

4. Structured Analysis is based on the principles of  
a) Top-down decomposition approach  
b) Divide and conquer principle  
c) Graphical representation of results using DFDs  
d) All of the mentioned

5. Which of the following is/are true with respect to functions ?  
a) A function such as “search-book” is represented using a circle  
b) Functions represent some activity  
c) Function symbol is known as a process symbol or a bubble in DFD  
d) All of the mentioned

6. Which of the following is not a use of a CASE tool ?  
a) Support structured analysis and design (SA/SD)  
b) Maintains the data dictionary  
c) Checks whether DFDs are balanced or not  
d) It complies with the available system

7. What DFD notation is represented by the Rectangle?  
a) Transform  
b) Data Store  
c) Function  
d) None of the mentioned

8. Structural decomposition is concerned with function calls.  
a) True  
b) False

9. A function-oriented design focuses on the entities in the system rather than the data processing activities.  
a) True  
b) False

10. In DFDs, user interactions with the system is denoted by  
a) Circle  
b) Arrow  
c) Rectangle  
d) Triangle

**Software Engineering Questions and Answers – Project Planing**

1. Which of the following is an important factor that can affect the accuracy and efficacy of estimates?  
a) Project size  
b) Planning process  
c) Project complexity  
d) Degree of structural uncertainty

2. What describes the data and control to be processed?  
a) Planning process  
b) Software scope  
c) External hardware  
d) Project complexity

3. A number of independent investigators have developed a team-oriented approach to requirements gathering that can be applied to establish the scope of a project called  
a) JAD  
b) CLASS  
c) FAST  
d) None of the mentioned

4. CLSS stands for  
a) conveyor line sorting system  
b) conveyor line sorting software  
c) conveyor line sorting speed  
d) conveyor line sorting specification

5. The project planner examines the statement of scope and extracts all important software functions which is known as  
a) Association  
b) Decomposition  
c) Planning process  
d) All of the mentioned

6. The environment that supports the software project is called  
a) CLSS  
b) SEE  
c) FAST  
d) CBSE

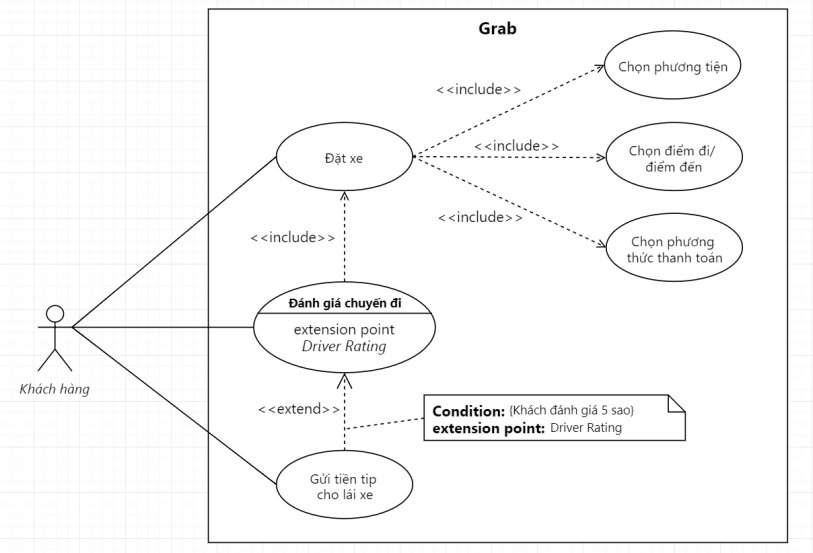
7. Which of the following is not an option to achieve reliable cost and effort estimate?  
a) Base estimates on similar projects that have already been completed  
b) Use one or more empirical models for software cost and effort estimation  
c) Use relatively simple decomposition techniques to generate project cost and effort estimates  
d) The ability to translate the size estimate into human effort, calendar time, and dollars

8. What can be used to complement decomposition techniques and offer a potentially valuable estimation approach in their own right?  
a) Automated estimation tools  
b) Empirical estimation models  
c) Decomposition techniques  
d) Both Automated estimation tools and Empirical estimation models

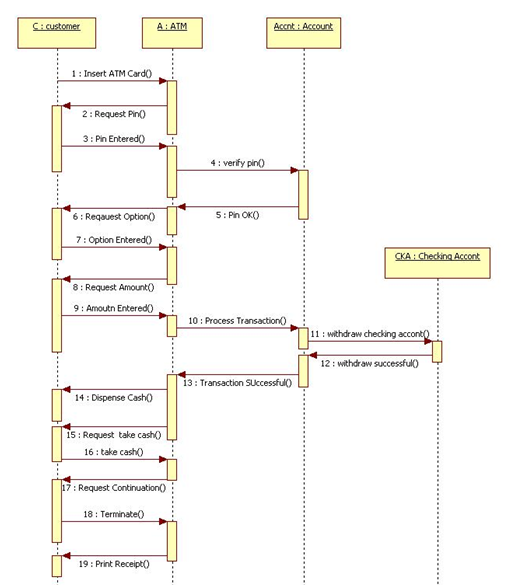
9. Which of the following is not achieved by an automated estimation tools?  
a) Predicting staffing levels  
b) Predicting software cost  
c) Predicting software schedules  
d) Predicting clients demands

10. Software project estimation can never be an exact science, but a combination of good historical data and systematic techniques can improve estimation accuracy.  
a) True  
b) False

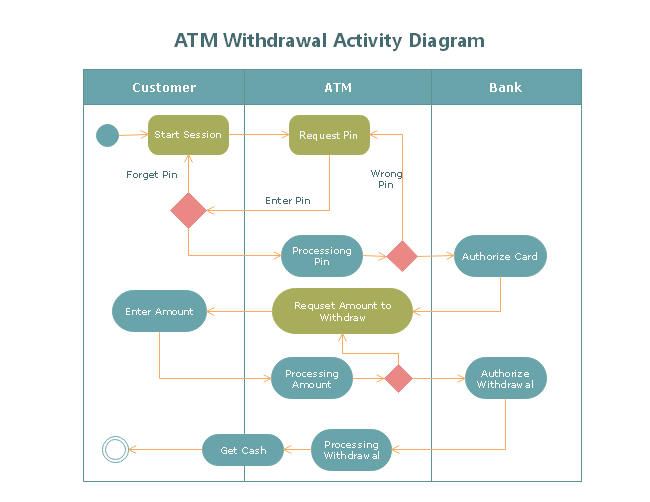
**Ví dụ Use case**



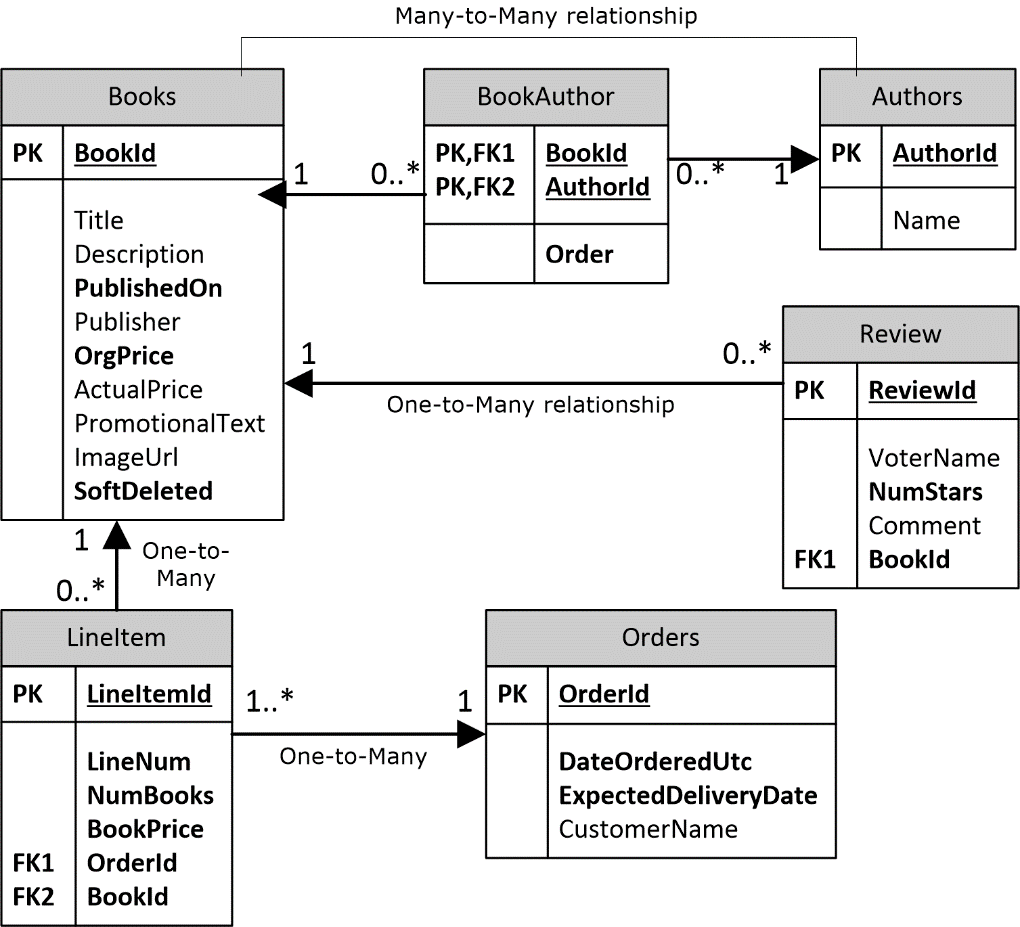
**Ví dụ Sequence diagram**



**Ví dụ Activity diagram**



**Ví dụ Entity class**



**Ưu điểm microservice**

Thứ nhất, giúp đơn giản hóa hệ thống. Với tổng số chức năng không đổi, kiến trúc Microservices chia nhỏ hệ thống ra làm nhiều dịch vụ nhỏ lẽ dể dàng quản lý và triển khai từng phần so với kiến trúc nguyên khối. Mỗi dịch vụ thì được định nghĩa để giao tiếp với các dịch vụ khác thông qua RPC (Remote Procedure Call) hay message-driven API. Kiến trúc Microservices thúc đẩy việc phân tách rạch ròi các dịch vụ nhỏ, việc khó có thể làm nếu xây dựng ứng dụng bằng kiến trúc một khối. Trên hết với mỗi dịch vụ nhỏ, chúng ta sẽ có thời gian phát triển nhanh hơn, dễ nắm bắt cũng như bảo trì hơn.

Thứ hai, kiến trúc này cho phép việc mỗi dịch vụ được phát triển độc lập bởi những team khác nhau. Cũng như cho phép developer có thể tự do chọn lựa technology stack cho mỗi dịch vụ mình phát triển. Dĩ nhiên tự do lựa chọn nhưng không phải là tạo ra một mớ hỗn độn về technology, điều này thì chằng có một dự án hay sản phẩm nào mong muốn cả. Tuy nhiên, sự tự do này có nghĩa là các developer không còn phải bắt buộc phải sử dụng các công nghệ lỗi thời có thể đã tồn tại vào lúc bắt đầu dự án. Khi viết một dịch vụ mới, họ có tùy chọn của việc sử dụng công nghệ bắt kịp với xu thế. Hơn nữa, vì dịch vụ là tương đối nhỏ, việc viết lại một dịch vụ cũ dựa trên nền tảng công nghệ mới hơn là hoàn toàn khả thi.

Thứ ba, kiến trúc Microservices cho phép mỗi dịch vụ có thể được triển khai một cách độc lập. Cùng với đó thì việc triển khai hệ thống theo kiểu continuous deployment là hoàn toàn có thể.

Cuối cùng, kiến trúc Microservices cho phép mỗi dịch vụ có thể thực hiện việc scale một cách độc lập. Bạn có thể scale dễ dàng bằng cách tăng số instance phục vụ cho mỗi dịch vụ lên và phân tải bằng load balancer. Ngoài ra bạn cũng có thể tối ưu chi phí vận hành dịch vụ bằng cách triển khai mỗi dịch vụ lên server có resource thích hơp. Ví dụ về việc này tôi đã nêu ra trong bài [Điạ ngục kiến trúc một khối](https://codeaholicguy.wordpress.com/2015/11/13/gioi-thieu-ve-microservices-phan-1-dia-nguc-kien-truc-mot-khoi/" \t "_new).

**Nhược điểm microservice**

Nhược điểm đầu tiên của Microservices cũng chính từ tên gọi của nó. Microservice nhấn mạnh kích thước nhỏ gọn của dịch vụ. Một số lập trình đề xuất dịch vụ siêu nhỏ cỡ dưới 100 dòng code. Nhưng làm sao để chia nhỏ, và chia làm sao để khi chia quá nhiều sẽ dẫn đến manh mún, vụn vặt, khó kiểm soát.

Nhược điểm thứ hai của Microservices đến từ đặc điểm hệ thống phân tán (distributed system). Lập trình viên cần phải lựa chọn phát triển mỗi dịch vụ nhỏ giao tiếp với các dịch vụ khác bằng cách nào messaging hay là RPC. Hơn thế nữa, họ phải xử lý sự cố khi kết nối chậm, lỗi khi thông điệp không gửi được hoặc thông điệp gửi đến nhiều đích đến vào các thời điểm khác nhau.

Thứ ba, phải đảm bảo giao dịch phân tán (distributed transaction) cập nhật dữ liệu đúng đắn (all or none) vào nhiều dịch vụ nhỏ khác nhau khó hơn rất nhiều, đôi khi là không thể so với đảm bảo giao dịch cập nhật vào nhiều bảng trong một cơ sở dữ liệu trung tâm. Theo nguyên tắc CAP (CAP theorem) thì giao dịch phân tán sẽ không thể thỏa mãn cả 3 điều kiện: consistency (dữ liệu ở điểm khác nhau trong mạng phải giống nhau), availablity (yêu cầu gửi đi phải có phúc đáp), partition tolerance (hệ thống vẫn hoạt động được ngay cả khi mạng bị lỗi). Những công nghệ cơ sở dữ liệu phi quan hệ (NoSQL) hay môi giới thông điệp (message broker) tốt nhất hiện nay cũng chưa vượt qua nguyên tắc CAP.

Thứ tư, testing một dịch vụ trong kiến trúc microservices đôi khi yêu cầu phải chạy cả các dịch vụ nhỏ khác mà nó phụ thuộc. Do đó khi phân rã ứng dụng một khối thành microservices cần luôn kiểm tra mức độ ràng buộc giữa các dịch vụ. Nếu các dịch vụ nhỏ thiết kế phục thuộc vào nhau theo chuỗi. A gọi B, B gọi C, C gọi D. Nếu một mắt xích có giao tiếp API thay đổi, liệu các mắt xích khác có phải thay đổi theo không? Nếu có thì việc bảo trì, kiểm thử sẽ phức tạp tương tự ứng dụng một khối. Thiết kế dịch vụ tốt sẽ giảm tối đa ảnh hưởng lan truyền đến các dịch vụ khác.

Cuối cùng, triển khai dịch vụ microservices nếu làm thủ công theo cách đã làm với ứng dụng một khối phức tạp hơn rất nhiều. Ứng dụng một khối bổ sung các server mới giống hệt nhau đằng sau load balancer. Trong khi ở kiến trúc microservices, các dịch vụ nhỏ nằm trên nhiều máy ảo hay Docker container khác nhau, hoặc một dịch vụ có nhiều thực thể phân tán ra nhiều. Theo Adrian Crockcroft, [Hailo có 160 dịch vụ](https://sudo.hailoapp.com/services/2015/03/09/journey-into-a-microservice-world-part-3/" \t "_new), NetFlix có hơn 600 dịch vụ. Với cloud, các máy ảo, docker container có thể linh động bật tắt, dịch chuyển. Vậy cần thiết phải có một cơ chế phát hiện dịch vụ (service discovery mechanism) để cập nhật tự động địa chỉ IP và cổng, mô tả, phiên bản của mỗi dịch vụ.

### **Những ưu điểm trong việc sử dụng mô hình Agile**

* Đạt được sự hài lòng của khách hàng bằng cách bàn giao nhanh chóng, liên tục các sản phẩm phần mềm có ích.
* Con người và tương tác được nhấn mạnh hơn là quá trình và công cụ. Khách hàng, nhà phát triển và người thử nghiệm liên tục trao đổi với nhau.
* Phần mềm làm việc được bàn giao thường xuyên (vài tuần chứ không phải vài tháng).
* Cuộc đối thoại trực tiếp (face-to-face) là hình thức giao tiếp tốt nhất.
* Gần gũi với nhau hơn, hợp tác hàng ngày giữa các khách hàng và các lập trình viên.
* Chú ý liên tục đến sự xuất sắc về kỹ thuật và bản thiết kế tốt.
* Thường xuyên thích nghi với hoàn cảnh thay đổi.
* Ngay cả những thay đổi muộn cũng được chào đón, cho dù là trong quá trình [UAT](https://dthshare.com/uat-la-gi-cam-nhan-dev-voi-viec-uat/).

### **Các nhược điểm khó tránh trong việc sử dụng mô hình Agile**

* Trong trường hợp một số sản phẩm phần mềm, đặc biệt là các sản phẩm lớn, rất khó để đánh giá những nỗ lực bắt buộc khi bắt đầu chu trình phát triển phần mềm.
* Thiếu sự nhấn mạnh vào thiết kế và tài liệu cần thiết.
* Dự án có thể dễ dàng off-track nếu đại diện khách hàng không rõ kết quả cuối cùng mà họ muốn.
* Chỉ những lập trình viên cao cấp mới có thể đưa ra các quyết định cần thiết trong quá trình phát triển. Do đó nó không có nơi cho các lập trình newbie, trừ khi kết hợp với các nguồn lực có kinh nghiệm.

### **ƯU ĐIỂM Scrum:**

* Một người có thể làm nhiều việc ví dụ như dev có thể test
* Phát hiện lỗi sớm hơn rất nhiều so với các phương pháp truyền thống
* Khách hàng nhanh chóng thấy được sản phẩm qua đó đưa ra phản hồi sớm.
* Có khả năng áp dụng được cho những dự án mà yêu cầu khách hàng không rõ ràng ngay từ đầu.
* Giảm thời gian dành cho quản lý, tăng thời gian dành cho việc phát triển

### **NHƯỢC ĐIỂM Scrum:**

* Trình độ của nhóm là có một kỹ năng nhất định
* Phải có sự hiểu biết về mô hình aglie .
* Khó khăn trong việc xác định ngân sách và thời gian.
* Luôn nghe ý kiến phản hồi từ khách hàng và thay đổi theo nên thời gian sẽ kéo dài khi có quá nhiều yêu cầu thay đổi từ khách hàng.
* Vai trò của PO rất quan trọng, PO là người định hướng sản phẩm. Nếu PO làm không tốt sẽ ảnh hưởng đến kết quả chung
* Khi phát triển dự án theo Scrum thì dự án sẽ không có detail design. Do vậy mỗi thành viên của dự án cũng sẽ là một người thiết kế hệ thống. Do vậy nếu phối hợp không tốt thì có thể dẫn đến việc sản phẩm rất khó "sửa chữa" (thực tế ở VN, không có nhiều dự án có detail design).

**Ưu điểm waterfall**

**Thích nghi tốt với những nhóm linh hoạt:** Dù không chỉ mình mô hình thác nước có ưu điểm này, ứng dụng nó giúp toàn bộ dự án được duy trì kỹ càng, có mục tiêu bao quát và thiết kế có cấu trúc nhờ vào việc phác thảo và pha tài liệu hóa từ trước. Điều này rất phù hợp với những nhóm lớn mà thường có các thành viên rời khỏi hoặc tham gia mới trong các chu trình sống của dự án. Nó cho phép thiết kế cốt lõi của dự án được đặt chủ yếu trong một tài liệu cụ thể, chứ không chỉ ở một thành viên nào đó của nhóm.

**Áp đặt một tổ chức có kết cấu chặt chẽ:** Điều này có thể bị coi là gánh nặng hơn là một lợi thế, nhưng sự thật là để duy trì mô hình thác nước bắt dự án, và thậm chí cả tổ chức xây dựng một dự án vô cùng chính xác, tuân nghiêm ngặt theo thiết kế và cấu tạo của nó. Những dự án lớn sẽ cần bao gồm những tiến trình cụ thể để quản lý toàn bộ khía cạnh của dự án, từ thiết kế và phát triển đến kiểm thử và triển khai.

**Cho phép những thay đổi thiết kế sớm:** Mặc dù sẽ rất khó để thay đổi thiết kế ở những giai đoạn sau, phương pháp thác nước giúp triển khai các thay đổi ở đầu vòng đời của ứng dụng khá dễ dàng. Vì chưa hề có mã hay bất cứ triển khai nào ở giai đoạn này, việc chỉnh sửa các tài liệu trở nên nhanh chóng và vô cùng đơn giản.

**Thích hợp cho những dự án theo hướng đến mốc:** Khi ứng dụng cấu trúc tuần tự của mô hình thác nước, những dự án sẽ rất phù hợp với những tổ chức vào nhóm hoạt động tốt dựa chủ yếu vào các mốc hoặc ngày. Với các pha rõ ràng và cụ thể, các thành viên trong nhóm có thể dễ dàng hiểu và chuẩn bị cho nó. Việc có một lịch trình cho toàn bộ quá trình và đề ra một vài thời điểm cụ thể hay dấu mốc quan trọng cho từng giai đoạn cũng đơn giản hơn. Tất nhiên điều này không có nghĩa rằng phát triển phần mềm thì không xảy ra chậm trễ, nhưng mô hình thác nước sẽ thích hợp cho những dự án có hạn chót hoàn thành.

**Nhược điểm waterfall**

**Ràng buộc thiết kế thích ứng kém:** Mặc dù có thể viết hẳn một quyển sách viết riêng về chủ đề này, thiếu sót quan trọng nhất của mô hình thác nước là khả năng thích ứng trước thay đổi trong toàn bộ vòng đời phát triển. Khi việc kiểm thử ở pha thứ năm phát hiện ra một số lỗi ở phần thiết kế hệ thống, nó không chỉ yêu cầu một bước lùi lớn về các bước cũ, trong một vài trường hợp còn phá hủy tính thống nhất của toàn bộ hệ thống. Trong khi phần lớn các nhóm và lập trình viên có kinh nghiệm sẽ rất khó để xảy ra những phát hiện muộn màng như vậy ngay từ đầu, tình trạng này vẫn có thể xảy ra, đặc biệt khi là các pha thường được để ở cuối của toàn chu trình.

**Bỏ qua phản hồi người dùng ở các giai đoạn sau:** Vì có một quá trình nghiêm ngặt từng bước một, mô hình thác nước gặp khó khăn trong vấn đề nhận phản hồi của người dùng ở những giai đoạn sau của vòng đời sản phẩm. Người quản lý dự án đương nhiên có thể đưa quá trình về các giai đoạn trước vì những yêu cầu hoặc thay đổi mới từ khách hàng, nhưng điều này sẽ vô cùng tốn kém và ngốn thời gian cho cả nhóm phát triển và khách hàng.

**Thời gian kiểm thử bị trì hoãn:** Trong khi phần lớn các mô hình SDLC hiện đại luôn tích hợp kiểm thử là một phần tất yếu và luôn luôn xuyên suốt mọi quá trình trong quá trình phát triển, mô hình thác nước để kiểm thử vào cuối vòng đời. Điều này không chỉ làm cho phần lớn lỗi kỹ thuật hay thậm chí vấn đề thiết kế không được phát hiện cho đến tận cuối chu trình sống, nó còn dễ gây ra thói quen viết mã kém chất lượng do việc kiểm thử chỉ thường khá ít và quá muộn.

**Mô hình dữ liệu**

1 Mô hình cơ sở dữ liệu phân cấp

Mô hình cơ sở dữ liệu phân cấp là khái niệm rất đơn giản nhưng việc thực hiện mô hình này rất phức tạp. Mô hình này ít có khả năng quản lý dữ liệu khổng lồ; hạn chế hỗ trợ tính toàn vẹn của cơ sở dữ liệu và độc lập về cấu trúc.

2. Mô hình cơ sở dữ liệu mạng

Mô hình cơ sở dữ liệu mạng là khái niệm rất đơn giản như mô hình phân cấp nhưng việc triển khai thực hiện rất phức tạp. Mô hình này xử lý nhiều loại mối quan hệ giữa các dữ liệu hơn và hỗ trợ tính toàn vẹn dữ liệu nhiều hơn so với mô hình cơ sở dữ liệu phân cấp. Mô hình này thiếu sự độc lập về cấu trúc và các tiêu chuẩn dữ liệu.

3. Mô hình cơ sở dữ liệu quan hệ

Mô hình cơ sở dữ liệu quan hệ hoạt động dựa trên khái niệm về tập hợp các lý thuyết về toán học. Trong tập hợp lý thuyết, việc thu thập thông tin hai chiều được gọi là mối quan hệ. Hệ thống quản lý cơ sở dữ liệu quan hệ RDBMS (Relational Data Base Management System) cung cấp cách thức đơn giản để xây dựng, truy cập và cập nhật cơ sở dữ liệu. Mô hình này chỉ hỗ trợ lưu trữ dữ liệu dưới dạng văn bản và giá trị số, nó không hỗ trợ lưu trữ kiểu dữ liệu trừu tượng như âm thanh, video và thông tin địa lý.

4. Mô hình cơ sở dữ liệu hướng đối tượng

Mô hình cơ sở dữ liệu hướng đối tượng là sự kết hợp giữa khái niệm về lập trình hướng đối tượng với công nghệ cơ sở dữ liệu để cung cấp một hệ thống phát triển ứng dụng tích hợp. Mô hình này hỗ trợ kiểu dữ liệu trừu tượng như âm thanh, video và thông tin địa lý. Trong Hệ thống quản lý cơ sở dữ liệu hướng đối tượng OODBMS (Object-Oriented Database Management System), chúng ta có thể cải thiện năng suất dựa trên việc hỗ trợ truy cập điều hướng và liên kết thông tin.