1. What is Software Engineering?

Software engineering is defined as a process of analyzing user requirements and then designing, building, and testing software application which will satisfy those requirements.

-----------------------------------------

2. Software Development methodologies:

2-1 Waterfall : The waterfall methodology is a project management approach that emphasizes a linear progression from beginning to end of a project.

Linear

Different roles

Document driven

Customer is outside the project

Large projects(time, nr. of people)

Req. statements

Rational Unified Process (RUP) is an agile software development methodology. RUP splits the project life cycle into four phases.

Iterations

Different roles

Document driven

Customer is outside the project

Large projects(time, nr. of people)

Use cases

XP: Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team.

Iterations

Cross-functional team

face to face

Customer inside project

small projects

user stories

Scrum : Agile scrum methodology is a project management system that relies on incremental development.

Scrum team including roles :

Product owner

ScrumMaster

Team

Ceremonies :

Sprint planning

Sprint review

Sprint retrospective

Daily scrum meeting

Product backlog refinement

Artifacts :

Product backlog

Sprint backlog

Burndown charts

Definition of done

Advantages :

Increased productivity

Improved project visibility

Higher software quality

Higher customer satisfaction

Less risks

Faster time-to-market

challenges :

The organizational culture needs to change

Scaling scrum

Scrum does not solve all your problems

The Product Owner role is difficult

Scrum does not always fit

Lean :Provide perfect value to the customer through a perfect value creation process that has the least amount of waste

Eliminate Waste

Amplify Learning

Decide as late as possible

Deliver as fast as possible

Empower the team

Kanban : method is designed to meet minimal resistance. It encourages continuous small incremental and evolutionary changes to the current process by implementing collaboration and feedback forms.

Scrum without fixed iterations(sprints)

DEVOPS : DevOps is a methodology meant to improve work throughout the software development lifecycle.

Close collaboration between developers and operations

Streamlines the delivery process of software from business requirements to production

Better communication

Identical development and production environment

Shared tools

Automate everything

Monitor everything

-----------------------------------------------

3. Question: What is the difference between RUP and Agile/Scrum?

While both Scrum and Rational Unified Process (RUP) follow the Agile framework, RUP involves more formal definition of scope, major milestones, and specific dates (Scrum uses a project backlog instead of scope). In addition, RUP involves four major phases of the project lifecycle (inception, elaboration, construction, and transition), whereas Scrum dictates that the whole “traditional lifecycle” fits into one iteration.

--------------------------------------------------

4 - What does RUP and Agile/Scrum have in common?

both Agile methods and UP are Iterative and Incremental Development (IID) methods

---------------------------------------------------

5 - Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

---------------------------------------------------

6- Git is a version control system that lets you manage and keep track of your source code history. GitHub is a cloud-based hosting service that lets you manage Git repositories

-----------------------------------------------------

7 - A software requirement can be of 3 types:

Functional requirements.

Non-functional requirements.

Domain requirements.

----------------------------------------------------

Agile requirments :

1. Define document properties

2. Communicate the overall goals

5. User Stories

6. User interaction and design

----------------------------------------------------

User story. What is the standard format for specifying :

A user story is usually written from the user's perspective and follows the format: “As [a user persona], I want [to perform this action] so that [I can accomplish this goal].”

---------------------------------------------------

2. Characteristics of Good User Stories (INVEST);

2.1. Independent

2.2. Negotiable

2.3. Valuable

2.4. Estimable

2.5. Small

2.6. Testable

---------------------------------------------------

The user story should have the following qualities:

Be complete enough to demonstrate user value.

Be user-centric.

Start with an epic.

Be short, simple, and clear.

-----------------------------------------------------------

User story Map :

User story mapping is a visual exercise that helps product managers and their development teams define the work that will create the most delightful user experience.

-----------------------------------------------------------

Acceptance Criteria : In Agile, acceptance criteria refer to a set of predefined requirements that must be met to mark a user story complete.

------------------------------------------------------------

Sppliting story : Story splitting is the process of breaking one single user story into smaller stories. However, it's not about breaking it into component tasks, but rather complete stories or slices that still deliver value to the user.

-----------------------------------------------------------

User stories estimation

Story points represent the relative sizing of the user story. It is a unit of estimation used by Agile teams to estimate User Stories. When the product owner wants some features to be developed he/she desires to know how soon the team can complete the features and how many resources it will take to complete the work.

--------------------------------------------------------------

User stories estimation

Story points represent the relative sizing of the user story. It is a unit of estimation used by Agile teams to estimate User Stories. When the product owner wants some features to be developed he/she desires to know how soon the team can complete the features and how many resources it will take to complete the work.

-------------------------------------------------------------

Name some NFRs

Scalability.

Capacity.

Availability.

Reliability.

Recoverability.

Maintainability.

Serviceability.

---------------------------------------------------------------------

Domain-driven design (DDD) is a software design approach focusing on modelling software to match a domain according to input from that domain's experts.

----------------------------------------------------

Domain Modeling is a way to describe and model real world entities and the relationships between them, which collectively describe the problem domain space.

-------------------------------------------------------------------

Software architecture refers to the fundamental structures of a software system and the discipline of creating such structures and systems.

--------------------------------------------------------------------------

Software architecture styles :

Domain Driven Design

Layering

Object oriented

Component based

Microservices

---------------------------------------------------------------------------

What is the difference between Loose-Coupling versus Tight

Coupling? Give example(s).

Tight coupling means classes and objects are dependent on one another. In general, tight coupling is usually not good because it reduces the flexibility and re-usability of the code while Loose coupling means reducing the dependencies of a class that uses the different class directly.

example : If you change your shirt, then you are not forced to change your body – when you can do that, then you have loose coupling. When you can't do that, then you have tight coupling.

--------------------------------------------------------------------------------

What is SOA?

service-oriented architecture, defines a way to make software components reusable and interoperable via service interfaces

People : employee decision makers / platform : increase operational efficiency / practice : emplot best practice methodology / process : aligne it with business operation

----------------------------------

Lesson7: Design

1. Domain entities : Entities are Domain concepts that have a unique identity in the problem domain.

2. Value objects : value object is a small object that represents a simple entity whose equality is not based on identity

3. Domain Services : Domain Services stores centralized directory information and lets users and domains communicate.

4. Data Access objects : data access object (DAO) is a pattern that provides an abstract interface to some type of database or other persistence mechanism.

5. Domain events

6. In Design, what does the Principle of High Cohesion, Loose Coupling mean?

In essence, high cohesion means keeping parts of a code base that are related to each other in a single place. Low coupling, at the same time, is about separating unrelated parts of the code base as much as possible. In theory, the guideline looks pretty simple

7. What are the SOLID principles for OO software Design?

Single responsability principle

open for exptension close for modification

Liskov substitution principle

interface segregation principle

dependency inversion principle

----------------------------

Clean code is a reader-focused development style that produces software that's easy to write, read and maintain.