1. **What are activities in the five core workflows of the united process?**

(Requirement workflow, Analysis workflow, Design workflow, Implementation workflow, Test workflow).

The five workflows within the united process is a set of activities that various project workers perform.

**Requirement**: The primary activities of the requirement workflow are aimed at building the use case model, which capture the functional requirement of the system being defined. This model helps the project stakeholders reach agreement on the capabilities of the system and the condition to which it must conform. The use case model also serve as the foundation for all other development work.

**Analysis**: The primary activities of the Analysis workflow are aimed at building the analysis model which helps the developers refine and structure the functional requirements captured within the use case model.

**Design**: The primary activities of the design workflow are aimed at building the design model, which describe the physical realizations of the use case from the use case model and also the contents of the analysis model. The design model servers as an abstraction of the implementation model. The design workflow also focuses on deployment model which defines the physical organization of the system in terms of computational modes.

**Implementation**: The primary activities of the implementation workflow are aimed at building implementation model, which describe how the elements of the design model are packaged into software components, such as source code files, dynamic link libraries (DLL) and EJB.

**Test**: The primary activities of test workflow are aimed at building the test model, which describe how integration and system test will exercise executable components from the implementation model. The test model also describe how the team will perform those test as well as unit test. The test model contains test case that are often derived directly from use cases. Tester perform black-box testing using the original use case text and white-box testing of the realization of those use cases, as specified within the analysis model. The test model also contain the result of all level of testing.

1. **Describe different types of risks in software development?**

For most software development project, we can define the five main risk impact areas:

New, unproven technologies, user and functional requirements, application and system architecture, performance and organization.

The first one new, unproven technologies because the majority of software project entail the use of new technologies. Ever changing tools, techniques, protocols, standards and development system increase the probability that technology risk will arise in virtually any substantial software engineering effort. Training and knowledge are of critical importance and the improper use of new technology most often leads directly to project failure.

The second one is user and functional requirement because software requirement capture all user needs with respect to the software system features, functions and quality of service. Too often the process of requirement definition is lengthy, tedious and complex. Moreover, requirements usually change with discovery, prototyping and integration activities. Change in elemental requirement will likely propagate through the entire project and modifications to user requirements might not translate to functional requirements. These disruptions often lead to one or more critical failures of a poorly planned software development project.

The third one is application and system architecture because taking the wrong direction with a platform, component, or architecture can have disastrous consequences. As with technological risk, it is vital that the team include expert who understand the architecture and the capability to make sound design choice.

The four one is performance because it’s important to ensure that any risk management plan encompasses user and partner expectation on performance. Consideration must be given to benchmark and threshold testing throughout the project to ensure that the work product are moving in the right direction.

The fifth and last one is organization because organization problems may have adverse effect on project outcome. Project management must plan for efficient execution of the project and find a balance between the needs of the development team and the expectations of the customer. Of course, adequate staffing includes choosing team members with skills set that are good match with the project.

1. **What are the characteristics of the object-oriented paradigm?**

The characteristics of object oriented programming language is encapsulation, inheritance and polymorphism.

Encapsulation is capturing data and keeping it safely and securely from outside interfaces.

Inheritance is the process by which a class can be derived from a base class with all features of base class and some of its own. This increases code reusability.

Polymorphism this is the ability to exist in various form. For example an operator can be overloaded so as to add two integer numbers and two floats.

Abstraction is the ability to represent data at a very conceptual level without any details.

Class definitions- basic building block OOP and a single entity which has data and operation on data together

Objects- the instance of a class which are used in real functionality its variables and operations.

Abstraction- specifying what to do but not how to do it, a flexible feature for having an overall view of an object functionality.

Genetic classes- class definition for unspecified data. They are known as container classes. They are flexible and reusable.

Class libraries- built in language specific classes

Message passing – object communication through invoking methods and sending data to them. This feature of sending and receiving information among object through function parameters is known as message passing.

1. **Describe software life-cycle models.**

A software life-cycle model is a description of the sequence of activities carried out in an SE project, and the relative order of these activities. It provides a fixed genetic framework that can be tailored to a specific report.

Code and fix life cycle model- it consist of no initial planning which put the development technology on risk for future errors. We need to start develop the application and fix the error at the same time. We will continue this process till the end of the development phase.

Waterfall model life cycle model- it requires pre-planning and makes the process of development easy because it catches all the flaws and errors which are going to occur during the developmental phase and how to handle them efficiently. The waterfall model consist of non-overlapping techniques which emphasizes planning at the foremost place and help to check the quality of the development application.

Evolution-tree model- the evolution tree model is the maintenance model. It is based on the requirement on the application that is to be developed. The evolution tree model is the only model which does the analysis of the design and the implementation of the given application and how should this be maintained.

Rapid prototyping- the development phase is essentially linear and hence the development procedure is very fast compared to others. No feedback is required because it is the rapid development process which only is delivered to have the development phase completed fast.

Open source life cycle – it is also a maintenance developmental model which requires a group of people qualitatively working together. A group of volunteers that work on the code on their spare time.

Agile processes- a controversial innovative approach which stories on what client wants. The estimate duration and cost of each story, the select stories for next build and each build is divided into tasks and test case for a task are draw up first, pair programming and continuous of the integration of the risk.

Synchronize and stabilize life cycle model- is used by Microsoft life cycle model and the requirements analysis is to interview potential customers, to draw up specifications and dived the group into 3 or 4 build. Each build is carried out by small team working in parallel. So at the end of the day synchronize (test and debug). Stabilize (freeze the build) and components always work together to get early insight into the operation of the product.

Spiral life-cycle model- is simplified form of rapid prototyping model plus risk analysis preceding each phase and if all risk cannot be mitigated the project is immediately terminated. So, each phase is determined by alternative and risk analysis and follow up evaluation and planning of the next phase. The radical dimension is to cumulative cost to date. Strength, so it easy to judge how much to test and there is no distinction made between development and maintenance. Weakness is large scale software only and internal in house software only.

1. **What are metrics used in different aspect of software engineering?**
2. **Describe key issues in software reusability and portability?**

Reusability refers to using components of one product to develop a different product with different functionality. Portability means to modify the product as a whole to execute it on different platform, complier, operating system etc.

The key issues in Reusability are:

NIH (Not invented here) syndrome: it means that software professional prefer to write their own component from scratch rather than reusing some other component. The developers would be willing to reuse a component provided they are assured that the component would not be harming their product. Reuse can be very expensive. Legal issues can arises in case of contract software.

Another issue arises when commercial off-the-shelf (COTS) components are used source code that is not known which brings difficulty in reusing it. So hardware incompatibility, operating system incompatibility, compiler incompatibility, numerical software incompatibility.

1. **Describe key issues in software design, for example cohesion and coupling.**

Software design is a process of converting the user requirements into a suitable from which helps the programmer in software coding and implementation. A number of key issues are dealt with while designing a software here are some of the key issues.

Cohesion is the measure of the degree of interpretability between the elements and module.

Coupling is define as the measure of interpretability between modules of the program.

Concurrency is the concern with decomposing software into processes, task, and threads and dealing with related issues of atomicity, efficiency, synchronization and scheduling.

Control and handling the events- the issue deals with how to organize data and control flow.

Data persistence- the issue deals how to handle long live data.

Security- this issue deals with how to prevent unauthorized access of the information or its creation modification or deletion.

1. **What are UML diagrams? Can describe several typical diagrams.**
2. **Describe key issue in software maintenance.**

Software maintenance means modification of the developed software to correct faults, improve performance and adapt to the environment.

Some key issues include the people writing the program are often isolated from each other,

Sometimes the program is changed by a person who did not understand the program properly leading to changes to original program,

There is information gap between the user of the software and programmer, the users are often not able to convery their requirements clearly to the programmer.

1. **Understanding iteration and incrementation in software engineering.**

Incrementation approach in software engineering is a method of software development which involves the design, implementation and testing of the software product incrementally until the final stage of the product is reach it also involve the development and maintenance the incremental approach follows a proper set of steps.

Iterative approach means cyclic process of prototyping, testing and analysis and refine a product based on the results of testing the most recent iteration the changes are made the iterative approach does not follow any proper set of steps.

1. **What are objectives of the analysis workflows?**

The number of task that are need to carry out as a set of conditions that determine the sides of the process. The goal of workflow is to manage flow of work such that can be done at the night time by the proper person. It is important to improve services and to make better decisions, employee satisfaction, and privacy access to achieve all the workflow is important. Groupwave it is required in collaboration with companies which promote team work and efficiency reducing communication overboard, System Architecture is to compare present business with earlier stages. Workflow modeling is flow chart, data flow diagrams, UML, transition system etc.

1. **What are concerns of the design workflow?**

This is a comprised model of process it need to accomplish, the set of people other resource available to perform those processes. The introduction of new functionalities and implemental of changes to make changes to the patents. Also the cost and efficiently pressure to improve flow.

1. **What are activities for requirement elicitation and analysis?**

Requirement elicitation is the process of discovering, reviewing and documenting the user needs and constrains for the system.

To know what problems to be solved and recognized system boundaries

To identity stakeholders and to recognize goals of system.

Requirement analysis is a process of obtaining, analyzing, and structuring the information carried by requirement engineer to under whole parts of problem and relation.

Building understanding about characteristics of problem

Developing detailed quality of attributes.

1. **What are activities for risk management?**

Risk management and its activities is to identity the risk and draw up plans to minimize their effect on project.

Risk identification- is the idea to identified technological risk, people risk and organization risk and need to identify them.

Risk Analysis- accessing the probability and seriousness of each risk.

Risk planning- consider each risk and developed a strategy to manage them.

Risk monitoring- assess the identified risk regularly

1. **For maintenance, describe it categories and corresponding activities.**

There are 4 type of maintenance.

Corrective maintenance is finding errors that are observed when software is use.

Adaptive maintenance is concerned with changing in software that take place to make the adaptive environment.

Perfective maintenance is change occurs when new functionalities are added it will happen.

Preventive maintenance involves in preventing the occurrence of errors.

1. **What is the test driven software development?**

Test driven software development is a methodology in developing software by conducting tests of features even before the development of those features.

Depends on the test result the actual product will be developed by incorporating necessary modifications if needed.

After the tests the developer will get necessary changes required in the design of the produce to pass the test.

Thus in this approach the tests are conducted prior to the actual development.

1. **What is extreme programming?**

Extreme programing is a special programming strategy in which specific needs of the software will be assigned to small teams.

It comes under one of the agile software development methodology.

The team behavior and structure is important in extreme programing since the teams has to be self-efficient and self-organize.

Each practice taken by the team will be self-complete and simple, then the integration of practices will produce the complex software.

1. **Describe the difference between cohesion and coupling. For each level of cohesion and coupling, understand its definition and weakness.**

Cohesion and coupling:

Cohesion and coupling will comes into discussion when software development is modularized.

Cohesion measures the dependency of the elements in the module.

The dependency among the elements of the module defined by the collision.

The interdependency among the different modules is defined by the coupling.

How two modules and inter related is defined by the coupling between them.

High cohesion means there will be higher level of dependency among the elements of the module.

High coupling means there will be high dependency among the modules of the software.

1. **What are activities for implementation workflow?**
2. **Understand different type of metrics in software engineering.**