CT-1

Q.1. Define software. What are the characteristics of a software?

Ans:

Software: Software is instructions that when executed provide desired function and performance, data structure that enable the programs to adequately manipulate information and documents that discribe the operations and use of programs.

The characteristics of a software are—

1. Software are developed or engineered, it is not manufactured in the classical sense.

2. Software doesn't "wear out."

3. Although the industry is moving toward component-based assembly, most software continues to be custom built.

Q.2. Briefly explain the support phase of software engineering.

Ans:

The support phase focuses on changes associated with error correction, adaptations required as the software's environment evolves, the changes due to enhancements brought about by changing customer requirements. four types of change are encountered during the support phase:

Correction: Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct defects.

Adaptation: Over time the original environment for which the software was developed is likely to change. Adaptive maintenace results in ~~to~~ modification to the software to accommodate changes to its external environment.

Enhancement: As software is used, the customer will recognize additional functions that will provide benefit. Prefective maintenace extends the software beyond its original functional requirements.

Prevention: Computer software deteriorates due to change, and because of this, preventive maintenance must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Q. 3. Write down the principles in agile.

Ans:

Principles in Agile:

1. Highest priority is to satisfy the customer.
2. Welcome changing requirements.
3. Deliver working version of software frequently.
4. Business people and developers must work together
5. Build projects around motivated individuals.
6. Emphasize face to face conversation.
7. Working software is the primary measure of progress.
8. Promote sustainable development.
9. Pays continuous attention to technical excellence and good design.
10. Maximize the amount of unnecessary work not done by defining value of each work.
11. The best architectures, requirements, and designs emerge from self organizing team.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

8.4. Write down two advantages and two disadvantages of test driven development and feature driven development.

Ans:

Test driven development

Advantages —

1. Increases the programmer's productivity.
2. Less time spent in the be debugger.

Disadvantages —

1. It takes a lot of time and effort which create a sense of slow development.
2. It's hard to test the interface because of continuous changes.

Feature driven development.

Advantages —

1. features can be organized into a hierarchical business-related grouping.
2. Because features are small, their design and code representations are easier to inspect effectively.

Disadvantages —

1. Not as powerful on smaller projects.
2. High reliance on chief programmer. He act as coordinator, lead designer, and mentor.

8.5. Provide two real examples of following categories of software

i. Real-time software
   Air traffic control system, Command control system

ii. Engineering and scientific software
   Data analysis software, Math calculation software.

iii. Artificial Intelligence software
   google assistant, Cortana.

iv. System software
   Windows 10, Android.

CT-2

Q.1. Briefly describe the 4P's associated with project management.

Ans: Effective software project management focuses on four P's: people, product, process, and project.

The people: The people factor is so important that the Software Engineering Institute has developed a people management capability maturity model (PM-eMM). The PM-eMM is a companion to the software capability maturity model that guides organizations in the creation of a mature software process.

The product: Before a project can be planned, product objectives and scope should be established, alternative solutions should be considered, and technical and management constraints should be identified.

The process: A software process provides the framework from which a comprehensive plan for software development can be established.

The project: We conduct planned and controlled software projects to manage ~~complexit~~ complexity.

Q. 2. What are the elements involved in risk item checklist.

Ans: One method for identifying risks is to create a risk item checklist. The checklist can be used for risk identification and focuses on some subset of known and predictable risks in the following generic subcategories:

- Product size — risk associated with the overall size of the software to be built or modified.
- Business impact — risk associated with constraints imposed by management or the marketplace.
- Customer characteristics — risk associated with the sophistication of the customer and the developer's ability to communicate with the customer in a timely manner.
- Process definition — risk associated with the degree to which the software process has been defined and is followed by the development organization.
- Development environment — risk associated with the availability and quality of the tools to be used to build the product.

- Technology to be built :— risk associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system.

- Staff size and experience —— risk associated with the overall technical and project experience of the software engineers who will do the work.

Q. 3. Briefly explain the project coordination techniques

Ans: Project coordination techniques —

- formal, impersonal approach: ~~software~~ Include software engineering documents and deliverables, technical memos, project control tools etc.

- formal, interpersonal procedures: focus on quality assurance activities applied to Software engineering ~~working~~ products.

- Informal, interpersonal procedures —— Include group meeting for information dissemination and problem solving and collection of requirements and development staff.

- Electronic communication :- encompasses electronic mail, electronic bulletine board, video-based conferencing system.

- Interpersonal networking — includes informal discussion with team members and those outside the project who may have experience or inright that can assist team members.

Q. 4. Write any foure sign that indicates an information systems project is in jeopardy.

Ans: Signs that indicates an information-systems project is in jeopardy —

1. The product scope is poorly defined.
2. Changes are managed poorly.
3. Deadlines are unrealistic.
4. Users are resistant.