SOFTWARE PROGRAMMING.

SECTION A

1a/ Software engineering

Is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures

ii/ Software evolution.

Refers to the process of continous improvement and development of a software system overtime. This include adding new features, fixing bugs, improving performance and adpting to changing user requirements and technological advancement.

iii/ Software development life cycle.

Is a well defined, structured sequence of stages in software engineering to develop the intended soft

b/ Basic constraints for software development

b i/ Time => Software must be delivered within a specified time frame, which is often limited by project deadlines or availability of resources.

ii/ Cost => The budget for the software development project must be managed and costs must be controlled to stay within the allocated budget

iii/ Scope => The scope of the software project must be defined and managed to ensure that the final product meets the specified requirement and expectations.

iv/ Quality => The software must meets the specified quality standards and be reliable, efficient and user friendly.

v Resources => The availability of resources, such as personnel, hardware and software must be managed to ensure the succesful completion of the project.

c/ Differentiate between Coupling and Cohesion.
Coupling refers to the degree of interdependence between modules or components in software system

WHILE

Cohesion refers to the degree of functional relatedness within a single module or component.

d/ When and Why yo chose.
@ Water fall Model. (b) Prototype Model.

General in software development choosing the appropiat model can greatly impact the success of a project

@ Waterfall => Waterfall model is a sequential, linear model that is best suited for project with well defined requirements, a clear under of the problem and focus on delivering a finished product in a predictable and controlled manner

water fall model is suitable for project with well defined deliverables and limited scope.

(b) Prototype => This model is an iterative model that is best suited for projects with rapid changing requirement and a need for early user feedback

=> This is particularly suitable for projects with complex or evolving requirements.

e/i) Factors for defining software quality.

(a) Functionality => This refers to the degree to which the software meets the need and expectation of the users.

(b) Usability => This refers to ease with which the software can be used by users. Include factors like interface design, navigation, and overall user experience.

2) Explain role of data flow diagram in software development.

Data flow diagram is a graphical representation of flow of data in a software system.

=> Role of data flow diagram in software development is to provide clear and comprehensive picture flow of data within a software system and help developers design more effective and efficient system.

SECTION B

3) i) Main X-ites of a good software.

(a) Usability => It must be user-friendly, intuitive and easy to use.

(b) Reliability => It must be stable, consistent and reliable, it must produce correct result and does not crash frequently.

(c) Performance => It must run efficiently and quickly with a responsive interface and minimal downtime.

(d) Security => It must protect user data and information with robust security measures in place to prevent hacking and data breaches.

@ Accessibility ⇒ It must be accessible to a wide range of users

(e) Compatibility (g) flexibility (h) Maintainability

3 ii/ Software design levels

@ High-level design:

This is the overall plan for the software system. It defines the main components of the system, their relationship and interactions and the overall architecture of the system.

(b) Detailed design

This level of design is more specific and focus on the implementation of individual components and modules. It defines algorithms, data structure and interface required to build the software system.

(c) Low level design:

This is the lowest level of design and provide a detailed implementation of the software system.

It includes the actual code and specific implementation details such as data storage and memory management.

3 (ii) Concepts that are basic in software development paradigm

(a) Abstraction.

Abstraction refers to the process of hiding the details of a system or component and presenting only the essentials features to user.

(b) Encapsulation.

Refers to grouping of data and functions into a single, self contained unity known as object.

(c) Modularity

Refers to division of software system into smaller, independent component or modules.

4) CASE TOOLS

CASE TOOLS ⇒ Computer Aided software Engineering.

CASE Tools ⇒ A reset of software application programs which are used to automate software development life cycle activities

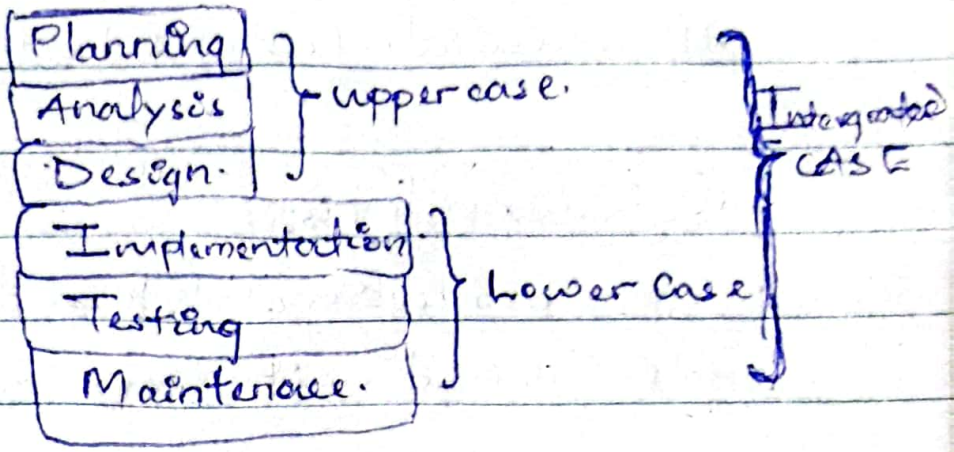
(ii) Components of case tool.

(a) Central Repository

(b) Upper Case tools {Planning, analysis and design stages}

(c) Lower case tools {Implementation, testing & maintenance}

(d) Integrated case tools



(iii) Primary Reason for using a CASE Tool

(a) Improved software development process:

⇒ CASE tools provide support for various aspects of software developing process including requirement design and modeling etc. This help to streamline the development process.

(b) Increased productivity:

Case tools provide comprehensive environment for software development and promote the development

5 i/ Software requirement.

⇒ Refers to description of features and functionalities of the target system.

⇒ Refers to the set of statement that define the fuctions, capabilities, constrains and performance characteristics of a software system.

ii/ Requirement Engineering.

⇒ The process of gathering the software requirement from client, analyze and document them.

⇒ Requirement engineering is the process of gathering analyzing, specify and managing the needs and contrains of stakeholders of the software system.

(iii) Importance of Requirement Gathering.

(a) Improved project management

(b) Increased efficiency.

(c) Better understanding of the project

(d) Increased stakeholder satisfaction.

(e) Decrease risk and re-work

(iv) Process of Requirement Gathering:

(a) Elicitation

This step involves identifying and gathering requirements from various stakeholders

(b) Analysis

In this step, the gathered information/requirements are analyzed and documented to ensure clear concise and complete.

(c) Validation.

In this step involves ensuring that the requirement accurately reflects the needs of stakeholders.

(d) Management

In this step, the requirement are managed and tracked throughout the development process.

6. (i Software testing

Is the process of evaluating a software with the intent to find whether it meets the specified requirements or not

The purpose of software testing is to identify, defects, errors and bugs in software product and improve its quality before it is released to end users.

ii/ Software validation.

Is the process of evaluating a software product during or at the end of development process to determine whether it satisfies specified requirements validation is typically performed before the software is release to the end users.

Software verification.

Is the process of evaluating a software product or system during the development process to determine whether it meets the specified design and requirements.

6 iii) Basic system testing.

@ Functional testing

ⓑ Performance testing

@ Security testing.

@ Functional testing ⇒ evaluates the software system functionality and verifies that it meets specified requirements

ⓑ Performance testing ⇒ evaluates software system's performance and scalability under different loads and conditions.

@ Security testing ⇒ evaluates software system's ~~functionality~~ security and identifies any vulnerabilities or weakness that could be exploited by attackers.

QUESTION 2.

2 i/ Unified modeling language.

Is the standardized language for specifying, visualizing, constructing and document the artifacts of software, as well as for business modeling and other non-software systems.