Software Engineering

Unit 3

Q.1) Explain design concepts :
i) Abstraction
ii) Modularity

⟹ (i) **Abstraction :**

- is a principle that simplifies complex systems by hiding detailed implementation and exposing only essential features.
- It allows developers to use functions, objects or data types without needing to understand their inner workings.

• Types of abstraction :

(1) Data Abstraction :
- This involves creating data models that hide the specifics of data representation.

(2) procedural Abstraction :
- This involves defining functions or procedures to perform tasks, allowing the details of the task implementation to be hidden.

(ii) **Modularity :**

- is the design principle that divides a system into distinct modules,
- each with a specific responsibility.

- These modules can be developed, tested & maintained independently.
- These modules can interact with each other through well-defined interfaces.

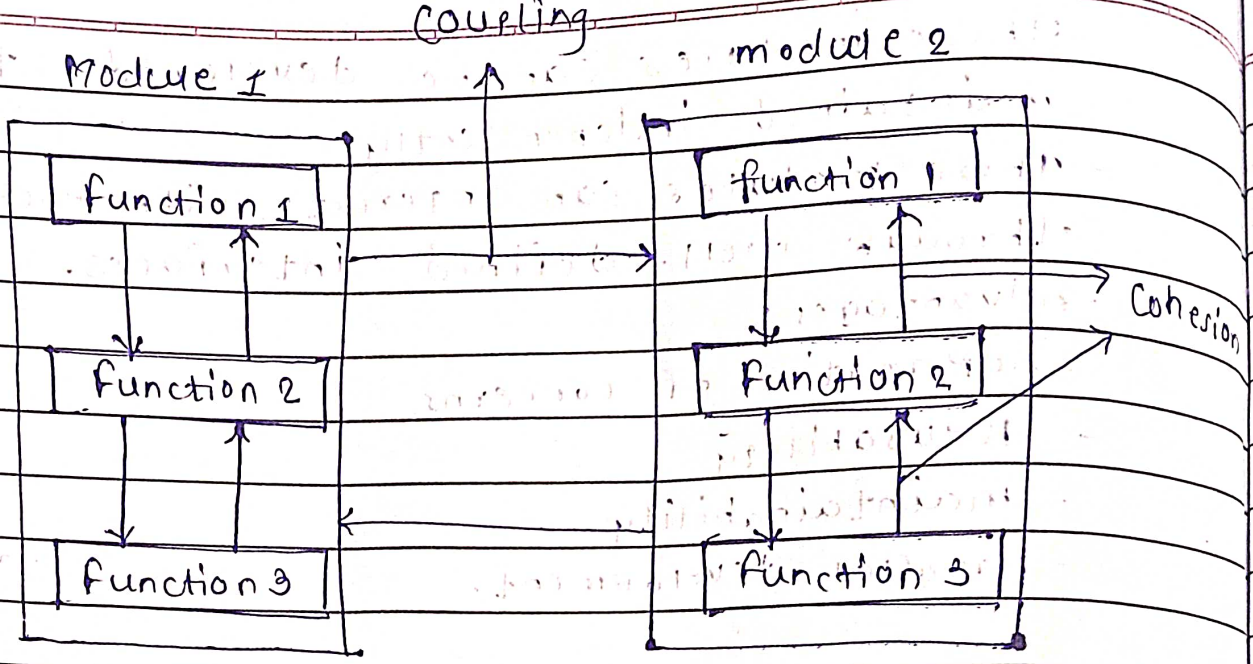
• Advantages :-
- Separation of concerns.
- Reusability
- maintainability.
- Parallel development.

Q.2) Write note on User Interface Design.

⟹ - User Interface (UI) design focuses on creating visually appealing, easy-to-use interfaces for software & devices.
- UI clearly convey fuctions & actions.
- UI maintain uniform visual elements & interaction patterns.
- UI Provide immediate responses to user actions.
- UI minimize effort needed to complete tasks
- It ensure the design is visually pleasing.
- make the interface usable for all abilities.
- organize elements logically.
- Use readable & appropriate fonts.
- Use intuitive symbols.
- Ensure easy movement through the application.

Q.3) Explain effective modular design with neat diagra

⟹ - Each module focuses on a single task or related tasks, ensuring all components within the module are highly related.



- Modules interact with each other through well-defined interfaces.
- minimizing dependencies & reducing the impact of changes.
- Internal details of each module are hidden from others.
- Modules are designed to be reused across different parts of the system or in different projects.
- saving time & effort
- making the system easier to maintain.

Q.4) ① Define Usability.
Explain different characteristics of Usability & Iso Characteristics.

→ - Usability is the measure of how easy & efficient it is for users to interact with a system, product or service to achieve their goals satisfactorily.

• Characteristics of Usability:

① Effectiveness : Accuracy & completeness with which users achieve their goals.
② Efficiency.
③ Satisfaction : Users comfort and positive attitudes towards the system.
④ Learnability :
⑤ Memorability :

• ISO Characteristics of Usability :

① Effectiveness
② Efficiency
③ Satisfaction
④ Context of use.

Q.5) Explain the mechanisms applied in User interface design for fulfilling the Mandel's three golden rules.

→ ① Place Users in Control :
- Ensure intuitive & consistent menus & buttons.
- Provide immediate responses to user actions.
- Allow users to easily reverse actions.
- Enable personalization of the interface.

② Reduce User's Memory load :
- Maintain uniformity in design elements.
- Use visual aids like icons & labels.
- Show only essential information initially, revealing more as needed.

③ Make the Interface Consistent:
- Use consistent colors, fonts & icons.
- Implement familiar & standard UI elements
- Ensure similar actions yield similar results.
- Use the same terms for similar functions throughout the interface.

Q.6) What is Software Architecture? Explain Data Centered & object oriented architectural style of the system.

⟹ - Software Architecture is the high-level structure of a software system, defining its components, their relationships & interactions.

• Data-centered Architectural style:

• Central data Repository:
- A single source of truth for data accessed by multiple components.
• Loose Coupling:
- components interact primarily through the shared repository, ensuring independe
• Scalability:
- Easily scaled by adding more Component
• Data integrity:
- Centralized data management enhances consistency.

• Object-oriented Architectural style:

• Encapsulation:
object encapsulate both data & behavior.

• Modularity:
System is divided into discrete, reusable objects or classes.

• Inheritance:
objects can inherit properties & behaviors, promoting reuse.

Q.7) Explain guidelines for Component level design & principles for User Interface design.

→ • Guidelines for Component level design:

① ~~Single~~
- Each component should have one responsibility.
- Component should be extendable without modifying existing code.
- Use specific, not general-purpose, interfaces for each component.
- Ensure each component's element's are closely related.
- Aim for low dependencies bet① components.
- ①esign components to be used in multiple parts of the system.

: principles for User Interface Design:

- maintain uniform design elements across the interface.
- Ensure the UI is understandable & intuitive.
- provide immediate responses to user actions.

- optimize for quick & easy user interface interactions.
- make important information & controls easily accessible.
- minimize user errors & provide ways to recover.

Unit 4

Q.1) Write note on effort estimation & scheduling.

⟹ - effort estimation & scheduling are crucial aspects of project management.
- Effort estimation involves predicting the amount of time, resources & manpower needed to complete a project task or deliverable.
- This often requires breaking down the project into smaller, more manageable tasks.

- Scheduling involves creating a timeline that outlines when each task or deliverable will be completed.
- It considers dependencies between tasks, resource availability & project deadlines.
- Both effort estimation & scheduling require careful planning, communication & flexibility to account for unforeseen challenges or changes in project scope.

Q.2) Enumerate the 4p's of project management.

⟹ ① Plan :

- This involves defining project objectives, scope, timelines, resources & budget.
- Planning sets the foundation for the project & outlines how it will be executed.

② process :

- process refers to the methodologies, procedures & workflows used to execute the project plan.
- It includes tasks, activities & the sequence in which they are performed to achieve project goals efficiently.

③ People :

- People are the individuals involved in the project including project managers, team members, stakeholders & Clients.

④ progress :

- progress involves monitoring & tracking project performance against the plan.
- It includes assessing milestones, identifying risks, resolving issues & making necessary adjustments to keep the project on track & ensure its successful completion.

Q.3) Explain in detail. Categories of software metrics with example.

⇒

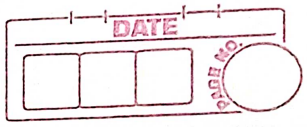
① Product metrics:
- measures the characteristics of quality of the software product itself.
- including its size, complexity, quality attributes & performance.

② Process metrics:
- Evaluate the efficiency, effectiveness & resource utilization of the software development process, including productivity, time-related aspects of overall process efficiency.

③ project metrics
- Assess the financial, risk-related & scheduling aspects of the software development project, including cost management, risk assessment & schedule monitoring.

Q.4) What is cocomo II?
What area does cocomo II address?

⇒ - Cocomo (constructive cost model) is a model used for estimating the cost, effort & schedule of software development projects.
- It is an extension of the original cocomo model developed by Barry Boehm in 1980s.



- cocomo II provides methods for estimating the size of software projects based on lines of code, function points or project points.
- It helps the estimate the effort required to develop the software based on its size, complexity & other project attributes.