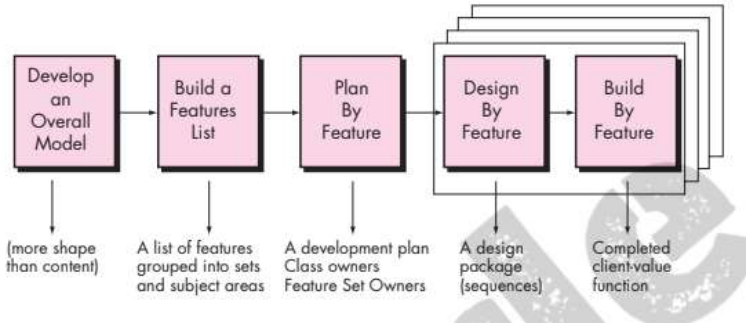
Internal Assessment Test 2 – NOV 2025
SCEHME & SOLUTION

Sub:	**Software Engineering & Project Management**					Sub Code:	**BCS501**	Branch:	**ISE**
Date:	**28/11/2025**	Duration:	**90 min**	Max Marks:	**50**	Sem/Sec:	**V / A, B & C**		**OBE**

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	a. Explain Feature Driven Development?	[5]	3	L2
Ans	b. **Feature-Driven Development (FDD)** was introduced by Peter Coad and later refined by Stephen Palmer and John Felsing to serve as a practical, agile methodology for object-oriented software development. c. It is well-suited for medium to large projects. d. FDD focuses on three core ideas: e. **Strong team collaboration**: ensuring smooth communication among all members. f. **Feature-based decomposition**: where the system is broken into small, client-valued features that are designed and built incrementally. g. **Clear communication**: using verbal discussions, diagrams, and concise documents. h. In the context of FDD, a feature "is a client-valued function that can be implemented in two weeks or less" [Coa99]. The emphasis on the definition of features provides the following benefits: i. Because features are small blocks of deliverable functionality, users can describe them more easily; understand how they relate to one another more readily; and better review them for ambiguity, error, or omissions. j. Features can be organized into a hierarchical business-related grouping. k. Since a feature is the FDD deliverable software increment, the team develops operational features every two weeks. l. Because features are small, their design and code representations are easier to inspect effectively. m. Project planning, scheduling, and tracking are driven by the feature hierarchy, rather than an arbitrarily adopted software engineering task set. Coad and his colleagues [Coa99] suggest the following template for defining a feature: **<action>** the **<result>** **<by | for | of | to>** a(n) **<object>** where an **<object>** is "a person, place, or thing (including roles, moments in time or intervals of time, or catalog-entry-like descriptions)." Examples of features for an e-commerce application might be: *Add the product to shopping cart* *Display the technical-specifications of the product* *Store the shipping-information for the customer*			

Ans	 **FIGURE 3.5** Feature Driven Development [Coa99] (with permission) n. List and explain the principles that guide the process elements of software development? o. **Principle 1. Be agile:** Whether the process model you choose is prescriptive or agile, the basic tenets of agile development should govern your approach. Every aspect of the work you do should emphasize economy of action—keep your technical approach as simple as possible, keep the work products you produce as concise as possible, and make decisions locally whenever possible. p. **Principle 2. Focus on quality at every step:** The exit condition for every process activity, action, and task should focus on the quality of the work product that has been produced. q. **Principle 3. Be ready to adapt:** Process is not a religious experience, and dogma has no place in it. When necessary, adapt your approach to constraints imposed by the problem, the people, and the project itself. r. **Principle 4. Build an effective team:** Software engineering process and practice are important, but the bottom line is people. Build a self-organizing team that has mutual trust and respect. s. **Principle 5. Establish mechanisms for communication and coordination:** Projects fail because important information falls into the cracks and/or stakeholders fail to coordinate their efforts to create a successful end product. These are management issues and they must be addressed. t. **Principle 6. Manage change:** The approach may be either formal or informal, but mechanisms must be established to manage the way changes are requested, assessed, approved, and implemented. u. **Principle 7. Assess risk:** Lots of things can go wrong as software is being developed. It's essential that you establish contingency plans. v. **Principle 8. Create work products that provide value for others:** Create only those work products that provide value for other process activities, actions, or tasks. Every work product that is produced as part of software engineering practice will be passed on to someone else. A list of required functions and features will be passed along to the person (people) who will develop a design, the design will be passed along to those who generate code, and so on. Be sure that the work product imparts the necessary information without ambiguity or omission.			
2 Ans	Explain the software quality models with neat diagrams (any 4)? **Software Quality Models** Different quality models: ❑ Garvin's quality dimensions ❑ McCall's Model ❑ Dromey's Model ❑ Boehm's Model	[10]	5	L2

Garvin proposed **eight dimensions of quality** which can be used to judge the quality of any product, including software.
These dimensions help understand quality from both **objective** and **subjective** views.

8 general attributes

1. **Performance** - how well it performs the jobs
2. **Features** - how well it supports the required features
3. **Reliability** - probability of a product working satisfactorily within a specific period of time
4. **Conformance** - degree to which the product meets the requirements
5. **Durability** - measure of the product life
6. **Serviceability** - Speed and effectiveness maintenance
7. **Asthetics** - the look and feel of the product
8. **Perceived quality** - user's opinion about the product quality

2. McCall's Model

McCall defined the quality of a software in terms of three broad parameters:

- its operational characteriestics,
- how easy it is to fix defects, and
- how easy it is to port it to different platforms

11 attributes of the software

1. **Correctness**: extent to which a software product satisfies its specifications
2. **Reliability**: probability of the software product working satisfactorily over a given duration
3. **Efficiency**: amount of computing resources required to perform the required functions
4. **Integrity**: the extent to which the data of the software product remains valid

5. **Usability:** the effort to operate the software product

6. **Maintainability:** ease with which it is possible to locate and fix bugs in the software product

7. **Flexibility:** effort required to adapt the software product to changing requirements

8. **Testability:** effort required to test a software product to ensure that it performs its intended function.

9. **Portability**: the effort required to transfer the software product from one hardware or software system environment to another.

10. **Reusability**: the extent to which a software can be reused in other applications

11. **Interoperability**: the effort required to integrate the software with other software

3. Dromey's Model

Dromey proposed that software product quality depends on 4 major high-level properties of the software:

- Correctness
- Internal characteristics
- Contextual characteristics
- Certain descriptive properties

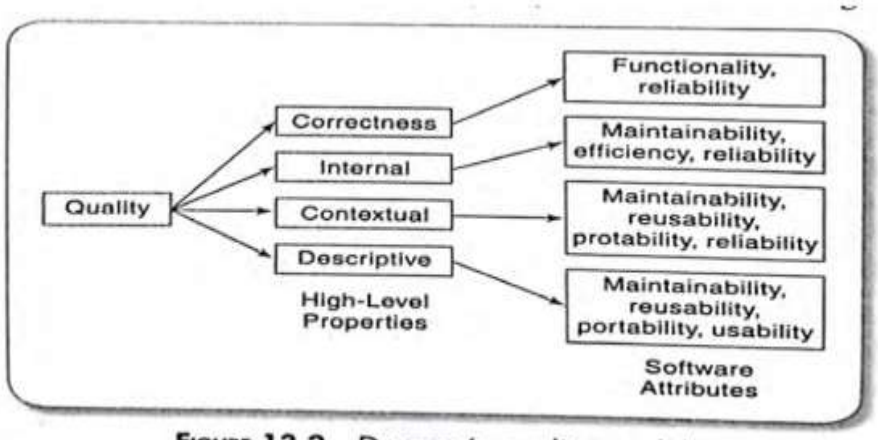


FIGURE 13.2 Dromey's quality model

4. Boehm's Model

- Boehm postulated that the quality of the software can be defined based on 3 high level characteristics that are important for the users of the software.
- 3 characteristics are:
  - **As-is utility** : how well (easily, reliably, and efficiently) can it be used?
  - **Maintainability** : How easy is it to understand , modify, and then retest the software?
  - **Portability:** how difficult would it be to make the software ina changed environment ?

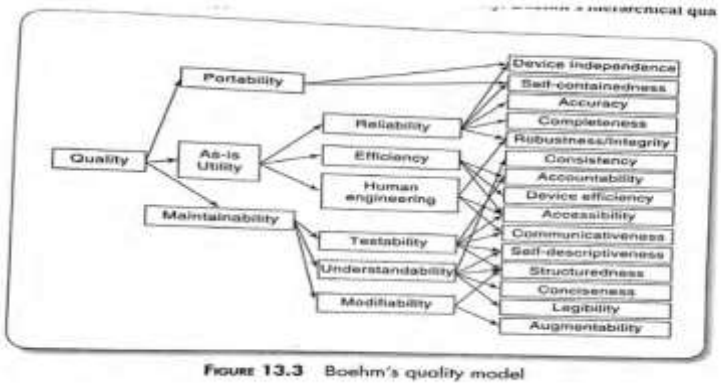
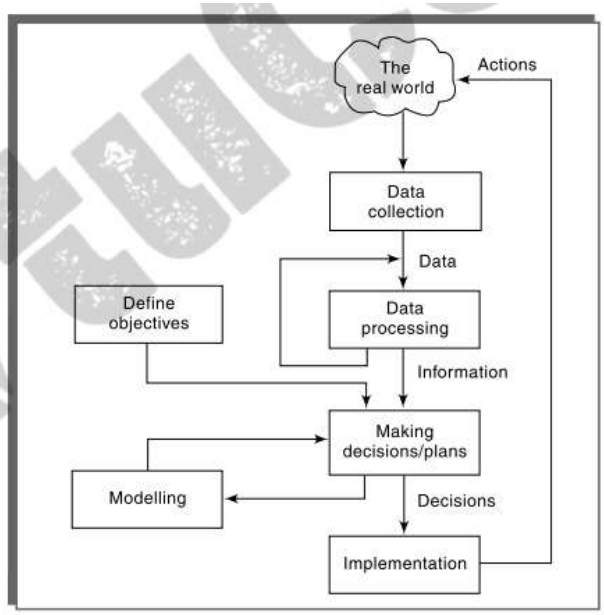


FIGURE 13.3 Boehm's quality model

Model	Main Focus	Perspective	Keywords / Core Idea
Garvin's	Customer satisfaction	User-oriented	Fitness for use, perceived quality
McCall's	Engineering process & measurable factors	Developer-oriented	Product operation, revision, transition
Boehm's	Balancing user utility & technical quality	Mixed	Utility, maintainability, portability
Dromey's	Relationship between design properties & quality	Product-based	Component properties, defect linkage

3 Ans	a. What is management? Explain the different activities involved in management?	[5]	4	L3
	b. **Defining Objectives** Management begins by setting clear goals and targets for the project or system.	[5]		
Ans	c. **Data Collection** Managers gather raw data from real-world operations to understand ongoing activities. d. **Data Processing** The collected data is converted into useful information (e.g., percentage completed, average work rate, estimated completion dates). e. **Performance Comparison** Actual progress is compared with planned objectives to identify delays or deviations. f. **Decision-Making and Planning** When issues are detected, managers decide what corrective actions to take, such as reallocating staff or resources. g. **Modelling Possible Actions** Before implementing decisions, different options are analysed and modelled to understand their consequences and avoid new problems. h. **Implementation of Decisions** The chosen plan is put into action to correct the performance issues. i. **Continuous Monitoring and Review** After implementation, new data is collected and processed to check if the actions are effective, and further adjustments are made if needed.  b. Explain the project management control life cycle with a suitable diagram?			

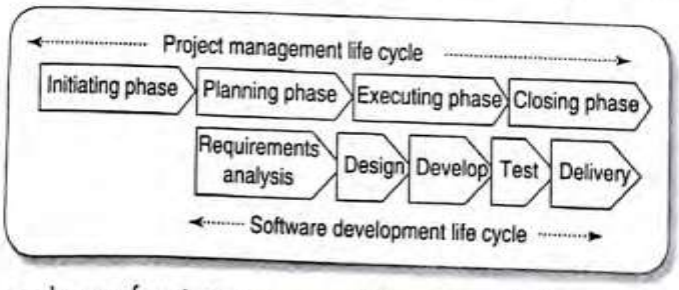


FIGURE 1.8 Different phases of project management life cycle and software development life cycle

1. Project Initiation

c. The **project initiation phase** begins with **project concept development**.
d. During this stage, the **project scope, constraints, costs, and expected benefits** are analyzed.
e. A **feasibility study** is conducted to assess financial and technical viability.
f. Based on this, a **business case** is prepared.
g. Once **top management approves** the business case, a **project manager** is appointed.
h. A **project charter** is created, and the **project team** is formed.
i. This establishes the foundation for the **project planning phase** to begin.

W5HH Principle: Boehm suggested that during project initiation, the project champions should have comprehensive answers to a set of key questions pertaining to the project. The answers to these questions would lead to the definition of key project characteristics. The name of this principle (W5HH) is an acronym constructed from the first letter of each question. This set of seven questions is the following:

- Why is the software being built?
- What will be done?
- When will it be done?
- Who is responsible for a function?
- Where are they organizationally located?
- How will the job be done technically and managerially?
- How much of each resource is needed?

Project planning

An important outcome of the project initiation phase is the project charter. During the project planning phase, the project manager carries out several processes and creates the following documents:

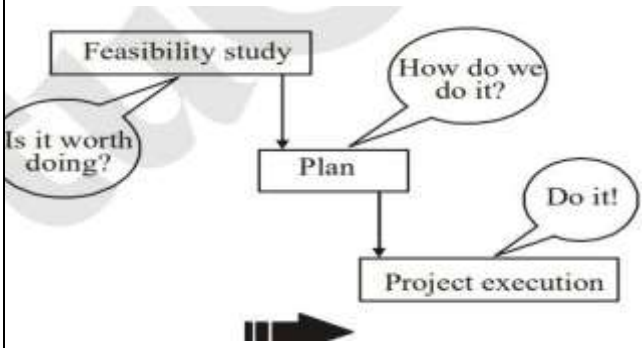
- *Project plan* This document identifies the project tasks, and a schedule for the project tasks that assigns project resources and time frames to the tasks.
- *Resource plan* It lists the resources, manpower and equipment that would be required to execute the project.
- *Financial plan* It documents the plan for manpower, equipment and other costs.
- *Quality plan* Plan of quality targets and control plans are included in this document.
- *Risk plan* This document lists the identification of the potential risks, their prioritization and a plan for the actions that would be taken to contain the different risks.

Project execution

In this phase the tasks are executed as per the project plan developed during the planning phase. A series of management processes are undertaken to ensure that the tasks are executed as per plan. Monitoring and control processes are executed to ensure that the tasks are executed as per plan and corrective actions are initiated whenever any deviations from the plan are noticed. However, the project plan may have to be revised periodically to accommodate any changes to the project plan that may arise on account of change requests, risks and various events that occur during the project execution. Quality of the deliverables is ensured through execution of proper processes. Once all the deliverables are produced and accepted by the customer, the project execution phase completes and the project closure phase starts.

	Project closure Project closure involves completing the release of all the required deliverables to the customer along with the necessary documentation. Subsequently, all the project resources are released and supply agreements with the vendors are terminated and all the pending payments are completed. Finally, a postimplementation review is undertaken to analyse the project performance and to list the lessons learnt for use in future projects.			
4	a. Explain the importance of software quality? reviewed. **13.3 Importance of Software Quality** We would expect quality to be a concern of all producers of goods and services. However, the special characteristics of software create special demands. • *Increasing criticality of software* The final customer or user is naturally anxious about the general quality of software, especially its reliability. This is increasingly so as organizations rely more on their computer systems and software is used in more safety-critical applications, for example to control aircraft. • *The intangibility of software* can make it difficult to know that a project task was completed satisfactorily. Task outcomes can be made tangible by demanding that the developer produce 'deliverables' that can be examined for quality. • *Accumulating errors during software development* As computer system development comprises steps where the output from one step is the input to the next, the errors in the later deliverables will be added to those in the earlier steps, leading to an accumulating detrimental effect. In general, the later in a project that an error is found the more expensive it will be to fix. In addition, because the number of errors in the system is unknown, the debugging phases of a project are particularly difficult to control. For these reasons quality management is an essential part of effective overall project management.	[5] [5]	5	L2
	b. Explain McCall's Model? **2. McCall's Model** McCall defined the quality of a software in terms of three broad parameters: • its operational characteriestics, • how easy it is to fix defects, and • how easy it is to port it to different platforms 11 attributes of the software 1. **Correctness**: extent to which a software product satisfies its specifications 2. **Reliability**: probability of the software product working satisfactorily over a given duration 3. **Efficiency**: amount of computing resources required to perform the required functions 4. **Integrity**: the extent to which the data of the software product remains valid 5. **Usability:** the effort to operate the software product 6. **Maintainability:** ease with which it is possible to locate and fix bugs in the software product 7. **Flexibility:** effort required to adapt the software product to changing requirements 8. **Testability:** effort required to test a software product to ensure that it performs its intended function.			

	9. **Portability**: the effort required to transfer the software product from one hardware or software system environment to another. 10. **Reusability**: the extent to which a software can be reused in other applications 11. **Interoperability**: the effort required to integrate the software with other software			
5 Ans	What is contract management and explain contract management process with a neat diagram? • **Definition:** Contract management is the process of **defining, establishing, monitoring, and controlling agreements** between the software developer and the customer. • **Purpose:** It ensures that both parties understand their **roles, responsibilities, deliverables, timelines, and payments** clearly. **Key Activities:** • Preparing and reviewing the **software contract/agreement**. • Specifying **scope of work**, **schedule**, and **costs** • Monitoring progress to ensure **compliance with contract terms**. • Managing **changes or disputes** during the project. • Ensuring **delivery and acceptance** as per agreed terms. **Importance:** • Avoids misunderstandings and conflicts. • Ensures **legal and financial protection** for both client and developer. • Provides a **formal basis for project control and accountability**. Various Stages of Contract Management 1. Request and Creation 2. Negotiation 3. Obligations and Performance 4. Modification and Renewal 5. Closure Benefits of Effective Contract Management • **Risk Mitigation:** Identifies and manages potential risks early in the contract lifecycle. • **Improved Compliance**: Ensures that all parties comply with legal and regulatory requirements. • **Cost Savings:** Avoids unnecessary costs and penalties by managing contracts efficiently. • **Performance Tracking**: Monitors performance against contract terms to ensure objectives are met. • **Relationship Management**: Maintains positive relationships between contracting parties through clear and consistent communication.	[10]	4	L3

	• **Speed to Market**: Accelerates project timelines by leveraging the vendor's expertise and resources.  			
6 Ans	Briefly explain the COCOMO II model? • COCOMO II (Constructive Cost Model II), developed by Barry Boehm, is an advanced and widely used software cost-estimation model. • It is an evolution of the original COCOMO model and provides more accurate estimates for modern software development environments. • COCOMO II is structured as a **hierarchy of estimation models**, each used at different stages of a project. **1. Application Composition Model** • Used during the **early stages** of software engineering. • Helpful when **prototyping user interfaces**, evaluating system interactions, assessing performance, and checking technology feasibility. • Uses **Object Points** (screens, reports, 3GL components) to size the application. **2. Early Design Stage Model** • Applied once **requirements are stabilized** and a basic software architecture is defined. • Provides estimates before the system is fully detailed. **3. Post-Architecture Stage Model** • Used during the **construction phase**, when the software architecture is complete and detailed information is available. • Offers the most refined and detailed cost estimates. **Object Point–Based Estimation** COCOMO II often uses **object points** as an indirect sizing measure. Object points depend on counts of: • Screens • Reports • 3GL components Each is classified into **simple, medium, or difficult**, with corresponding weights. **New Object Points (NOP)** If reuse is involved, the adjusted object point value is calculated as: $\text{NOP} = (\text{object points}) \times [(100 - \%\text{reuse})/100]$ **Effort Estimation** After computing NOP, the effort is estimated using: Where **PROD** is the productivity rate based on **developer experience** and **environment maturity**. $\text{PROD} = \frac{\text{NOP}}{\text{person-month}}$ $\text{Estimated effort} = \frac{\text{NOP}}{\text{PROD}}$ Example PROD values (given): • Very Low: 4 • Low: 7 • Nominal: 13 • High: 25 • Very High: 50	[10]	5	L2