Q1.

In addition to providing project and resource schedules, activity planning aims to achieve a number of other objectives which may be summarized as follows.

• Feasibility assessment: - Is the project possible within required timescale and resource constraints? However, it is not until we have constructed a detailed plan that we can forecast a completion date with any reasonable knowledge of its achievability

• Resource allocation: - What are the most effective ways of allocating resources to the project. When should the resources be available? The project plan allows us to investigate relation between

timescales and resource availability (in general, allocating additional resources to the project shortens its duration) and the efficacy of additional spending on resource procurement

• Detailed costing: - How much will the project cost and when is that expenditure likely to take place? After producing an activity plan and allocating specific resources we can obtain more detailed estimates of costs and their timing

• Motivation: - Providing targets and being seen to monitor achievements against the targets is an effective way of motivating staff. Particularly where they have been involved in setting those targets in the first place

- Coordination: - When do the staff in different department needs to be available to work on a particular project and when do staff need to be transformed between projects? The project plan, particularly with large project involving more than a single project team. provides an effective vehicle for communication and coordination among.

Q2.

- Review of work items is a significant mechanism for checking the advancement of a project and guaranteeing the quality of the work items.
- It is important to kill as many deformities in these work items to understand a result of acceptable quality.
- Testing is a convincing effective defect removal technique, but testing is relevant to just executable code.
- Review is pertinent to all work items even the non-executable work items.
- Review is more cost-effective in removing defects.

UTILITY OF REVIEW

- Apart from review being the most cost-effective defect removal technique, it offers other advantages too.
- A review usually aids in identifying any deviations from standards, as well as issues that may affect the software's maintenance.
- Reviewers make suggestions for how to improve the work items.
- A review meeting, in recognising defects, frequently provides learning opportunities for not only the creator of a work items, but also the other review meeting participants.
- Participants in the review gain a thorough understanding of the work items under consideration, making it simpler for them to communicate with or use the work items in their work.

Q3.

The types of contract are

1. **FIXED PRICE CONTRACTS**
    a. When the contract is signed, the price is set.
    b. The customer understands that if the contract terms are not changed, this is the price they will pay upon completion.
    c. For this to work, the customer requirements must be established from the start.
    d. Once development has begun, the customer cannot modify their prerequisites without renegotiating the contract price.
    e. The advantages are known customer expenditure and supplier motivation.
    f. The disadvantages of this techniques are higher prices to allow for contingency, difficulties in modifying requirements, upward pressure on the cost of changes and threat to system quality.
2. **TIME AND MATERIALS CONTRACTS**
    a. The customer is charged a fixed rate per unit of effort under this type of contract.
    b. The supplier would provide an initial cost estimate depending on the current insight of the customer's requirements, but this does not constitute the grounds for the final payment.
    c. Typically, the supplier bills the customer on a regular basis for work completed.
    d. The advantages are ease of changing requirements and lack of price pressure.

e. The disadvantages are customer liability and lack of incentive for supplier.
f. Customers dislike this approach because it appears to give the supplier a blank check.
g. The hiring of contract development staff, on the other hand, may involve this type of contract.

3. FIXED PRICE PER UNIT DELIVERED CONTRACTS

a. It is frequently used in conjunction with function points (FP).
b. At the start of the project, the size of the system to be delivered is calculated or estimated.
c. The size could be estimated in terms of lines of code, but FPs can be derived more easily from requirements documents.
d. There is also a price per unit mentioned.
e. The unit price is then calculated by multiplying the number of units to arrive at the final price.

4. OPEN TENDERING PROCESS

a. In response to the invitation to tender, any supplier may submit a bid.
b. Tenders must all be evaluated in the same manner.
c. Local/international law (including EU and WTO, World Trade Organization, requirements) may compel government bodies to do so.
d. When client is a public body, then open tendering is a compulsory choice.

5. RESTRICTED TENDERING PROCESS

a. Only bids from suppliers who have been specifically invited by the customer are accepted in this case.
b. It has the potential to reduce the number of suppliers considered at any stage.
c. This is the best approach to implement.

6. NEGOTIATED TENDERING PROCESS

a. Nevertheless, there may be some valid reasons why the restricted tendering process is not appropriate in certain situations.
b. In these cases, a single supplier approach may be justified.
c. We negotiate with a single supplier, for example, for extensions to previously supplied software.
d. Approaching a single supplier, on the other hand, may expose the customer to accusations of favoritism and should be done only with a sufficient explanation.

Q4. Define terms

a. **Net profit** : Net profit is the difference between the total cost and total income of the entire life of the project.
b. **Return on Investment** : Return on investment (ROI) is a performance measure used to evaluate the efficiency or profitability of an investment or compare the efficiency of a number of different investments.
c. **Payback period** : It is the time taken to pay back the initial investment.
d. **Net Present Value** : Net Present Value (NPV), most commonly used to estimate the profitability of a project, is calculated as the difference between the present value of cash inflows and the present value of cash outflows over the project's time period. If the difference is positive, it's a profitable project and if it is negative, then it's not worthy.
e. **Internal Rate of Return** : Internal rate of return is the interest rate that discounts an investment's future cash flows to the present so that the present value of cash inflows exactly equals to the current worth of the cash outflow i.e., at that interest rate when the net present value will be equal to zero.

Q5. Define the following terms

a. **Critical path** : Critical path is a method for modelling projects where you input all necessary factors involved in your project and output the optimal timeline for completing it. Factors to input in your model include time estimates, task dependencies, milestones or deliverables, and any hard deadlines set by clients or stakeholders.
b. **Float** : The time a task will take is defined as the float time, which indicates how long you have before it must be completed. The amount of time a task can be delayed without delaying the project's deadline is known as total float time.
c. **Free float :** The time by which an activity may be delayed without affecting any subsequent activity. It is calculated as the difference between the earliest completion date for the activity and the earliest start date of the succeeding activity. This might be considered a more satisfactory measure of float for publicising to the staff involved in undertaking the activities.

d. **Interfering float** : The difference between total float and free float. This is quite commonly used, particularly in association with the free float. Once the free float has been used (or if it is zero), the interfering float tells us by how much the activity may be delayed without delaying the project end date — even though it will delay the start of subsequent activities.

e. **Hammock Activities** : Hammock activities are activities which_ in themselves, have zero duration but are assumed to start at the same time as the first `hammocked' activity and to end at the same as the last one. They are normally used for representing overhead costs or other resources that will be incurred or used at a constant rate over the duration of a set of activities.

Q. Software quality models

Certainly, here's a concise summary of each of the mentioned quality models:

1. McCall's Quality Model (1977):

- Developed for the US military to bridge the gap between users and developers.
- Three major representations: Product Revision, Product Transition, Product Operations.
- Key factors include maintainability, flexibility, testability, portability, re-usability, interoperability, correctness, reliability, efficiency, integrity, and usability.

2. Boehm's Quality Model (1978):

- Structured around high-level, intermediate, and primitive characteristics.
- High-level characteristics address as-is utility, maintainability, and portability.
- Intermediate characteristics include portability, reliability, efficiency, usability, testability, understandability, and flexibility.

3. Dromey's Quality Model (1995):

- Analyzes software quality through tangible quality properties.
- Components (e.g., variables, functions, modules) associated with quality evaluation models.
- Four categories of properties: Correctness, Internal, Contextual, Descriptive.
- Emphasizes the relationship between quality attributes, sub-attributes, and product properties.

4. ISO/IEC 9126 Model:

- Part of ISO 9000 standard, a widely recognized quality assurance standard.
- Hierarchical tree structure of characteristics and sub-characteristics.

- Six main characteristics: Functionality, Reliability, Usability, Efficiency, Maintainability, Portability.
- Provides consistent terminology and a framework for trade-offs.

5. FURPS Model:

- Categorizes requirements into Functional and Non-Functional (Usability, Reliability, Performance, Supportability).
- Focuses on expected input/output, human factors, reliability, performance, and support.

6. Ghezzi Model:

- Emphasizes internal and external qualities.
- Overall qualities include accuracy, flexibility, integrity, maintainability, portability, reliability, re-usability, and usability.

7. IEEE Model:

- Provides specifications for software maintenance and a quality model.
- Offers measurement techniques for various qualitative factors like efficiency, functionality, maintainability, portability, reliability, and usability.

8. SATC's Model (Software Assurance Technology Centre):

- Aims to improve software quality through metrics and discussions.
- Evaluates results of metrics and discussions based on project goals and process attributes.

9. Capability Maturity Model (CMM):

- Defines software quality standards with five levels: Initial, Repeatable, Defined, Managed, Optimizing.
- Focuses on process maturity and improvement.

Q. Comparison of models

Certainly, here's a comparison of various software development models and methods, including both traditional and agile approaches:

Waterfall Model:

- Sequential and linear approach.
- Progresses through stages like requirements, design, implementation, testing, deployment, and maintenance.
- Well-defined and documented processes.
- Changes are difficult to accommodate once a phase is completed.

Spiral Model:

- Iterative and risk-driven approach.

- Cycles through planning, risk analysis, engineering, and evaluation phases.
- Emphasizes risk management and flexibility.
- Suitable for complex, long-term projects.

Software Prototyping:

- Involves creating a simplified version of the software to gather user feedback.
- Useful for refining requirements and user interfaces.
- Not suitable for all projects, especially those with strict regulatory requirements.

Incremental Delivery:

- Divides the project into small, manageable parts.
- Each increment is developed and delivered independently.
- Allows for early product releases and ongoing improvements.
- Suitable for projects with evolving requirements.

Atern/Dynamic Systems Development Method (DSDM):

- An iterative and incremental approach.
- Emphasizes collaboration, communication, and user involvement.
- Focuses on delivering a minimum usable subset of the system quickly.
- Suitable for projects with changing requirements.

Rapid Application Development (RAD):

- Focuses on rapid prototyping and quick development.
- Involves stakeholders early and often.
- Emphasizes user involvement and feedback.
- Suitable for projects with well-understood requirements.

Agile Methods:

- A family of iterative and incremental methods.
- Values individuals and interactions, working software, customer collaboration, and responding to change.
- Agile methods include Scrum, Extreme Programming (XP), and Lean Software Development.

Extreme Programming (XP):

- Emphasizes customer satisfaction and collaboration.
- Involves practices like test-driven development, continuous integration, and pair programming.

- Encourages frequent small releases.
- Suitable for small to medium-sized teams.

Scrum:

- Employs time-boxed iterations called sprints.
- Roles include Scrum Master, Product Owner, and Development Team.
- Daily stand-up meetings to track progress.
- Suitable for projects with evolving requirements.

Lean Software Development:

- Focuses on delivering value to the customer while minimizing waste.
- Eliminates unnecessary steps and processes.
- Continuous improvement and customer focus.
- Suitable for optimizing processes in any development model.