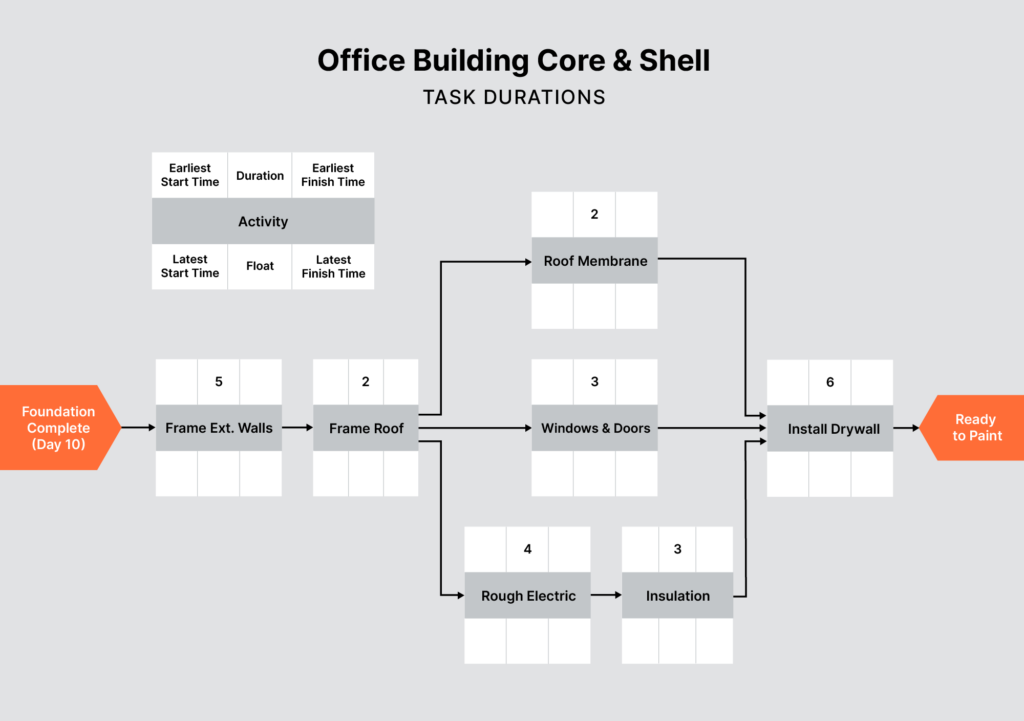
**Question 1: Application of Critical Path Method in SDLC [(Total Marks: 10)]**

You are managing a software development project following the SDLC process. The key activities, their durations, and dependencies are given below:

| **Activity** | **Description** | **Duration (Days)** | **Predecessors** |
| --- | --- | --- | --- |
| A | Requirements Gathering | 3 | - |
| B | System Design | 4 | A |
| C | Database Design | 3 | B |
| D | Frontend Development | 5 | B |
| E | Backend Development | 6 | C |
| F | Integration and Testing | 4 | D, E |
| G | Deployment | 2 | F |

**Tasks:**

1. Draw the activity network diagram. (3 marks)
2. Perform a forward pass and backward pass to determine the earliest and latest start/finish times. (4 marks)
3. Identify the critical path and total project duration. (2 marks)
4. Which activities have float, and how much? (1 mark)



**Question2: Earned Value Management in a Mobile App Development Project3**

**(Total Marks: 10)**

You are the project manager of a 10-week mobile app development project with a **total budget (BAC)** of **$100,000**. The work is divided into 5 phases:

| **Phase** | **Planned % of Total Work** | **Planned Duration (weeks)** |
| --- | --- | --- |
| Requirements Analysis | 10% | 1 |
| UI/UX Design | 20% | 2 |
| Backend Development | 30% | 3 |
| Frontend Development | 25% | 2 |
| Testing & Deployment | 15% | 2 |

At the **end of Week 5**, the following progress and costs are reported:

* **Actual Cost (AC)** = $55,000
* **Work completed = 40%**
* **Planned % of work by end of Week 5 = 60%**

**Tasks:**

1. Calculate the **Earned Value (EV)**. (2 marks)
2. Calculate the **Estimate at Completion (EAC)** assuming current cost performance continues. (3 marks)
3. Calculate the **Schedule Performance Index (SPI)** and estimate the **revised project duration** based on it. (3 marks)
4. Comment on the **project’s current cost and schedule status**. (2 marks)

Hint:

* EV = % of actual work completed × BAC
* EAC = BAC / CPI
* SPI = EV / PV

**Question3:Application-Based Scrum**

You have been hired as the **Scrum Master** for a tech startup building a food delivery mobile app. The team includes front-end and back-end developers, a Product Owner, UX designers, and QA testers. The app is expected to go live in four months. Stakeholders are pushing for faster releases, while the team is struggling with unclear user stories and repeated changes during the sprint.

Write a detailed essay explaining how you would apply Scrum principles and practices to:

1. Structure the **roles and responsibilities** within the Scrum team
2. Manage and prioritize a **dynamic Product Backlog**
3. Facilitate effective **Sprint Planning, Daily Scrums, Sprint Review, and Retrospective**
4. Deal with frequent mid-sprint change requests
5. Improve **team performance and stakeholder collaboration** over time

Support your answer with specific tools (e.g., burndown chart, definition of done, backlog refinement) and real-world practices from similar agile projects.

|  |
| --- |
| 🔹 1. **Scrum Roles and Responsibilities**  * **Product Owner (PO):** Owns the Product Backlog, prioritizes features such as live tracking, payment gateway integration, and restaurant onboarding. * **Scrum Master:** Facilitates Scrum ceremonies, removes blockers, shields team from distractions, and coaches on agile best practices. * **Development Team:** Cross-functional members who design, build, test, and deliver a potentially shippable product increment each sprint.   Application: I would clarify each role during kickoff and revisit responsibilities in retrospectives when confusion arises. 🔹 2. **Managing and Prioritizing the Product Backlog** The Product Backlog is dynamic and should reflect evolving priorities.   * I would schedule **regular backlog refinement sessions** (at least once per sprint). * Use techniques like **MoSCoW** or **Weighted Shortest Job First (WSJF)** to help the PO prioritize. * Ensure backlog items are broken down into **INVEST-compliant user stories** (Independent, Negotiable, Valuable, Estimable, Small, Testable).   Example: "As a user, I want to track my order live so I can know when to be ready." 🔹 3. **Facilitating Scrum Ceremonies**  * **Sprint Planning:** Team selects backlog items and defines the **Sprint Goal**. I’d ensure only refined items are selected to prevent confusion during development. * **Daily Scrum:** 15-minute stand-up where team members discuss progress, plans, and blockers. As Scrum Master, I’d coach the team to keep it focused. * **Sprint Review:** Stakeholders review the **increment**. I’d help the team present demos that align with the sprint goal. * **Sprint Retrospective:** Discuss what went well, what didn’t, and actionable improvements. I’d encourage psychological safety for open feedback.   Example: After a sprint with delayed testing, we might agree to shift one developer to pair with QA mid-sprint next time. 🔹 4. **Dealing with Mid-Sprint Change Requests** Scrum discourages changes mid-sprint unless urgent.   * I’d coach the PO and stakeholders to **channel new requests** into the Product Backlog and prioritize them for the next sprint. * If change is unavoidable, I’d facilitate a conversation between PO and team to **cancel or revise the current sprint** if needed (as per Scrum Guide).   Example: A request to add a wallet feature mid-sprint would be deferred unless it blocks a critical path. 🔹 5. **Improving Team Performance and Stakeholder Collaboration**  * Implement a **Definition of Done (DoD)** to ensure clarity in deliverables. * Use **Velocity charts** and **Burndown charts** to help forecast future sprints and measure team capacity. * Encourage direct PO-user feedback loops via beta testing groups. * Foster team ownership by rotating demo presenters and encouraging cross-functional pairing.   Long-term: As the team matures, I would gradually reduce interventions, allowing them to self-organize and deliver consistently. |

### **Question4: Structured PMBOK® Approach for a Food Delivery App** You are managing a startup project to build a food delivery mobile application with features such as real-time order tracking, online payment integration, and restaurant onboarding. The project involves developers, designers, QA testers, investors, and restaurant partners and must be completed within six months.

Using the **PMBOK® Guide’s five Process Groups and ten Knowledge Areas**, present a **structured approach** for managing this project.  
Your answer should:

* Use a **table format** to map each **Process Group** with the relevant **Knowledge Areas**
* Describe specific actions and tools applied to the food delivery app project under each combination
* Demonstrate practical understanding with scenario-based examples

|  |
| --- |
| **1. Initiating Phase** 🔹 Objective: Define the project, authorize work, and identify stakeholders.  **Key Knowledge Areas:**   * **Integration Management:**   + Develop the **Project Charter** outlining scope, objectives, constraints, and high-level budget/timeline.   + App example: Charter includes launch goal in 6 months with GPS tracking, payment integration, and 100+ restaurant partners. * **Stakeholder Management:**   + Identify key stakeholders (founders, developers, investors, restaurant partners, end-users).   + Create a **Stakeholder Register** to classify by influence and engagement needs.  **2. Planning Phase** 🔹 Objective: Develop detailed plans for scope, schedule, cost, quality, risk, resources, communication, procurement, and stakeholder engagement.  **Key Knowledge Areas:**   * **Scope Management:**   + Define **Product Scope** (features like live tracking, ratings, loyalty program).   + Create a **WBS** to break features into manageable deliverables. * **Schedule Management:**   + Sequence activities, estimate durations, and define the **project timeline** using **Gantt charts/CPM**. * **Cost Management:**   + Use **bottom-up estimation** to forecast cost: developer salaries, API costs, marketing.   + Establish a **Cost Baseline**. * **Quality Management:**   + Define quality standards (response time < 3s, crash rate < 1%).   + Include **code review, usability testing**, and **performance benchmarks** in the plan. * **Resource Management:**   + Identify team roles: front-end, back-end devs, QA, UI/UX, support staff.   + Develop **staffing and resource acquisition plans**. * **Communications Management:**   + Build a **Communication Plan** outlining frequency, format, and channels of reporting.   + Ex: Slack for daily dev updates, investor summary every 2 weeks. * **Risk Management:**   + Identify risks like app downtime, vendor delays, poor user adoption.   + Build a **Risk Register** and plan response strategies. * **Procurement Management:**   + Plan contracts for payment gateway, map APIs, SMS providers.   + Decide on **contract types** (fixed-price vs. T&M). * **Stakeholder Management (continued):**   + Develop a **Stakeholder Engagement Plan**.   + Plan investor demos, user feedback forums, and restaurant partner training.  **3. Executing Phase** 🔹 Objective: Perform the planned activities to produce deliverables.  **Key Knowledge Areas:**   * **Integration Management:**   + Direct and manage project execution.   + Use **change control process** to handle new feature requests (e.g., adding dark mode). * **Resource Management:**   + Onboard team members.   + Manage collaboration using tools like GitHub, Jira, or Notion.   + Apply **conflict resolution and team development strategies**. * **Communications Management:**   + Maintain **transparent communication** with all stakeholders.   + Conduct sprint reviews and demos for stakeholders. * **Procurement Management:**   + Select vendors and issue contracts.   + Manage relationships and track delivery of third-party services. * **Stakeholder Management:**   + Ensure stakeholder feedback is incorporated regularly.   + Conduct **engagement meetings, beta testing feedback loops**.  **4. Monitoring and Controlling Phase** 🔹 Objective: Track, review, and regulate performance. Implement necessary changes.  **Key Knowledge Areas:**   * **Scope Management:**   + Control scope with **formal change approvals**.   + Ensure no scope creep. * **Schedule Management:**   + Track progress using **SPI (Schedule Performance Index)**.   + Re-baseline if needed. * **Cost Management:**   + Monitor expenses and compare **CPI (Cost Performance Index)**.   + Adjust resource allocation to control budget. * **Quality Management:**   + Perform **Quality Control** via UAT, regression testing, bug tracking. * **Risk Management:**   + Monitor identified risks and reassess or trigger mitigation plans.   + Update the **Risk Register**. * **Communications Management:**   + Provide consistent **status reports** to stakeholders. * **Procurement Management:**   + Validate vendor deliverables and address non-performance. * **Stakeholder Management:**   + Measure stakeholder satisfaction using engagement metrics and feedback.  **5. Closing Phase** 🔹 Objective: Finalize all activities and formally close the project.  **Key Knowledge Areas:**   * **Integration Management:**   + Confirm all project work is complete and objectives met.   + Close contracts, archive documents, and hand over final product. * **Procurement Management:**   + Verify final vendor deliveries and close contracts. * **Stakeholder Management:**   + Conduct final stakeholder presentation and collect lessons learned.   + Capture feedback and recommendations for future projects. |