1. Define Project Control in simple terms.

Project control is the process of **monitoring and regulating project activities** to ensure they are completed according to the planned scope, schedule, budget, and quality. It involves comparing actual project performance with the planned baseline and taking corrective actions when deviations occur. In simple words, project control means **keeping the project on track** and making adjustments so that objectives are achieved successfully.

Example: In a website development project, if the coding phase is delayed by 5 days, project control ensures adjustments are made—such as adding more developers or re-allocating resources—to meet deadlines.

---

2. List any two examples of quantitative techniques used in project monitoring.

- **Earned Value Analysis (EVA):** A method that integrates scope, cost, and schedule to measure project performance and progress objectively. It calculates indicators like Schedule Performance Index (SPI) and Cost Performance Index (CPI).
- **Trend Analysis:** The process of using historical project data to predict future outcomes. For example, analyzing monthly defect rates to forecast quality trends.

---

3. What is meant by piloting in project monitoring?

Piloting is a **trial implementation of a project deliverable on a small scale** before full-scale deployment. It helps detect issues, assess feasibility, and improve the final product before rolling it out to all users.

Example: Before launching a new student attendance system across all departments, a college tests it in one department for a month to gather feedback and fix any problems.

---

4. Write any two objectives of configuration management.

- **Ensure Consistency:** Maintain uniformity in performance, design, and functional specifications throughout the project lifecycle.
- **Change Control:** Prevent unauthorized modifications by ensuring that every change is properly documented, reviewed, and approved.

---

5. State the difference between Quality Assurance and Quality Control in one sentence each.

- **Quality Assurance (QA):** A process-oriented activity that focuses on preventing defects by following systematic methods and standards during development.
- **Quality Control (QC):** A product-oriented activity that focuses on identifying and fixing defects in the finished product through inspection and testing.

6. What is a quality audit? Mention one benefit.
A **quality audit** is a systematic and independent review of a project's processes, deliverables, and management practices to ensure they comply with predefined quality standards, policies, and procedures. It can be conducted internally by the organization's own quality team or externally by third-party auditors.

Purpose:

- Identify deviations from standards.
- Suggest improvements in the process.
- Maintain compliance with regulations.

Benefit:
One major benefit is that it **helps detect issues early**, ensuring corrective action is taken before the product is delivered, which ultimately improves customer satisfaction.
Example: In a banking software project, a quality audit may review coding guidelines, test coverage, and documentation accuracy.

---

7. Explain the term "Monitoring Process" with an example.
The monitoring process in project management is the continuous activity of **tracking project performance**, collecting progress data, comparing it with the plan, and reporting results to stakeholders. It ensures the project is on the right path and allows for timely corrective actions if deviations occur.

Steps in Monitoring:

1. Define what will be monitored (time, cost, quality, risks).
2. Collect real-time data during execution.
3. Compare actual vs. planned results.

Example:
In a mobile app development project, monitoring involves checking if the UI design phase is

completed within the planned 3 weeks, whether costs are within budget, and if the designs meet the agreed specifications.

---

8. Describe how effort data is useful in project control.
Effort data records the **actual amount of work done** in terms of hours or days for each task in a project.

Uses:

1. **Performance Tracking:** Compare actual vs. planned effort to evaluate progress.
2. **Identify Bottlenecks:** If one task takes more time than planned, managers can investigate reasons like lack of skills or poor resource allocation.
3. **Improve Future Planning:** Data from current projects can be used to make better estimates for future projects.

Example:
If a module was planned for 50 hours but took 80 hours, the data can help improve next project's estimation and prevent delays.

---

9. How is data collection important in project monitoring? Give one example.
Data collection is essential because it provides **accurate, objective, and measurable information** about project performance. Without data, managers rely on assumptions, which can lead to errors.

Importance:

- Detect problems early.
- Make informed decisions.
- Track progress toward project objectives.

Example:
In a software testing phase, tracking the number of defects found daily helps determine if the quality is improving and whether the software is ready for release.

---

10. Outline the process of configuration management in brief.
Configuration management ensures that all project deliverables are consistent and changes are controlled.

Steps:

1. **Identification:** Select and label all configuration items (code files, documents, design diagrams).
2. **Change Control:** Evaluate, approve, or reject changes to these items.
3. **Status Accounting:** Keep updated records of configuration changes and versions.
4. **Verification:** Ensure configurations match requirements and standards.

Example:
In a software project, version control tools like Git are used to manage and track code changes.

Q11. Explain any two tools and techniques used in project monitoring.

Definition:
Project monitoring tools and techniques help project managers **track, measure, and evaluate** the progress of a project to ensure it stays on schedule, within budget, and meets quality standards.

---

1. Gantt Charts

- **Explanation:** A Gantt chart is a visual representation of a project schedule in the form of a horizontal bar chart. Each bar represents a task, with its length showing duration and its position showing start and end dates.
- **Importance:**
    1. Shows task dependencies and overlapping activities.
    2. Helps identify delays and reallocate resources.
    3. Improves communication with stakeholders.
- **Advantages:** Simple to understand, visually clear, and effective for tracking deadlines.
- **Example:** In a mobile app project, the Gantt chart may show that the "UI Design" task overlaps with "Backend Development," allowing parallel work to save time.

---

2. Earned Value Management (EVM)

- **Explanation:** EVM is a quantitative performance measurement technique that integrates project scope, cost, and schedule. It uses specific metrics to evaluate the project's health.
- **Key Metrics:**
    - **Planned Value (PV):** Budgeted value of planned work.
    - **Earned Value (EV):** Budgeted value of actual completed work.
    - **Actual Cost (AC):** Actual cost of completed work.
    - **CPI (Cost Performance Index) = EV ÷ AC:** Budget efficiency.
    - **SPI (Schedule Performance Index) = EV ÷ PV:** Schedule efficiency.
- **Advantages:** Detects budget/schedule overruns early and provides reliable forecasts.
- **Example:** In a software upgrade project, a CPI of 0.8 means the project is 20% over budget; SPI of 1.1 means it's ahead of schedule.

Q12. Describe the steps involved in the configuration management process.

Definition:
Configuration Management (CM) is the process of **systematically handling changes** so that a project maintains its performance, functionality, and design consistency over its lifecycle.

Steps:

1. **Planning:**
    - Define the scope, objectives, responsibilities, tools, and processes for CM.
    - *Example:* Decide on a version control system like Git.
2. **Identification:**
    - Select and uniquely label all configuration items (CIs) such as documents, source code, test plans, and user manuals.
3. **Change Control:**
    - Review, approve, or reject proposed changes.

- Ensure only authorized changes are implemented.

4. **Status Accounting:**
    - Maintain updated records of all configuration items and their current versions.
5. **Verification & Audit:**
    - Verify that changes were correctly implemented.
    - Audit to ensure the configuration meets requirements.

Importance:
CM ensures **consistency, traceability, and controlled change management**, reducing errors in complex projects.

Q13. Compare Quality Assurance and Quality Control with suitable examples.

Definition:

- **Quality Assurance (QA):** Process-oriented approach focused on preventing defects by ensuring the right processes are followed.
- **Quality Control (QC):** Product-oriented approach focused on identifying and fixing defects in the final output.

Comparison Table:

Aspect	Quality Assurance (QA)	Quality Control (QC)
Objective	Prevent defects	Detect and correct defects
Nature	Proactive	Reactive
Focus	Process quality	Product quality
Timing	Before and during development	After development
Example	Conducting code reviews to ensure coding standards	Testing a completed software module for bugs

Conclusion:
Both QA and QC work together—QA ensures processes are correct, while QC ensures the end product is correct.

Q14. Explain the importance of quantitative techniques in project monitoring.

Definition:
Quantitative techniques involve using **numerical data and mathematical methods** to measure, track, and predict project performance.

Importance:

1. **Objective Decision-Making:** Eliminates guesswork and bias.
2. **Performance Tracking:** Tracks cost, schedule, and quality using measurable indicators.
3. **Early Warning System:** Identifies potential problems before they become critical.
4. **Forecasting:** Helps predict completion dates, final costs, and overall project success.

Example:
Earned Value Analysis (EVA) can show that the project is 15% behind schedule but 5% under budget, allowing managers to take corrective measures.

Conclusion:
Without quantitative data, project monitoring becomes subjective; with it, managers can make precise, data-driven decisions.

Q15. Describe how piloting helps in minimizing project risks.

Definition:
Piloting is the process of implementing a **small-scale trial version** of a system before rolling it out on a large scale.

Benefits in Risk Minimization:

1. **Problem Detection:** Identifies technical issues in a controlled environment.
2. **User Feedback:** Gathers real user opinions for improvements.

3. **Cost Efficiency:** Fixing problems during the pilot is cheaper than after full deployment.
4. **Stakeholder Confidence:** Builds trust by showing the system works.

Example:
Before deploying a new railway ticketing system nationwide, the railway authority pilots it at two stations to ensure smooth functioning.

Conclusion:
Piloting acts as a safety net that detects risks early, ensuring smoother implementation and higher project success rates.

Q16. Explain the role of performance metrics in project control with examples.

Definition:
Performance metrics are **quantitative measures** used to evaluate and track the effectiveness, efficiency, and progress of a project against planned objectives. They form the backbone of project control, as they provide factual data rather than relying on assumptions.

Role in Project Control:

1. **Tracking Progress:**
    - Metrics like Schedule Performance Index (SPI) and Cost Performance Index (CPI) help compare planned vs. actual performance.
    - Example: SPI < 1 indicates the project is behind schedule.
2. **Quality Assurance:**
    - Metrics such as defect density, code coverage, and mean time to failure ensure the final product meets quality standards.
3. **Resource Utilization:**
    - Measures such as hours worked vs. planned hours ensure resources are being used optimally.
4. **Decision-Making Support:**
    - Data from metrics helps managers decide whether to continue, adjust, or stop a project.
5. **Early Risk Identification:**

- Variations in performance metrics can signal potential risks, allowing preventive action.

Example:
In a mobile banking app project, a high number of defects during system testing may lead to rescheduling the release date to maintain quality.

Conclusion:
Performance metrics turn project control into a **data-driven discipline**, ensuring timely delivery, budget compliance, and product quality.

Q17. Describe the process of risk management in software projects.

Definition:
Risk management is the systematic process of **identifying, analyzing, planning for, and responding to risks** that may affect the success of a software project.

Process Steps:

1. **Risk Identification:**
   - Detect all potential risks—technical, managerial, resource-related, or external.
   - Example: Delay in hardware delivery or sudden requirement change.
2. **Risk Analysis:**
   - Assess the probability and impact of each risk.
   - Use qualitative (high/medium/low) or quantitative (numerical) analysis.
3. **Risk Prioritization:**
   - Focus on risks with the highest potential impact.
4. **Risk Response Planning:**
   - **Avoidance:** Change the plan to eliminate the risk.
   - **Mitigation:** Reduce the likelihood or impact.
   - **Acceptance:** Prepare to deal with it if it occurs.
5. **Risk Monitoring and Control:**

- Continuously review risks and update the risk register throughout the project.

Example:
In an e-commerce platform project, the risk of payment gateway failure can be mitigated by integrating multiple payment providers.

Conclusion:
Effective risk management increases the chances of project success and reduces costly surprises.

Q18. Discuss the importance of earned value analysis in project monitoring.

Definition:
Earned Value Analysis (EVA) is a **project performance measurement technique** that integrates scope, schedule, and cost to assess project health and predict future performance.

Importance:

1. **Integrated View:**
    - Combines cost, schedule, and work progress in a single framework.
2. **Early Warning Signals:**
    - Identifies deviations from the plan before they become major problems.
3. **Objective Measurement:**
    - Replaces subjective progress reports with factual numbers.
4. **Forecasting Capability:**
    - Predicts final project cost (Estimate at Completion – EAC) and schedule completion date.
5. **Better Communication:**
    - Provides stakeholders with a clear picture of project status.

Example:
If EV = ₹4,00,000, AC = ₹5,00,000, and PV = ₹6,00,000:

- CPI = 0.8 → Project is over budget.

- SPI = 0.66 → Project is behind schedule.

Conclusion:
EVA allows project managers to **control projects proactively** rather than reactively, ensuring better results.

Q19. Explain the steps of project closure in detail.

Definition:
Project closure is the final phase in the project lifecycle where all activities are completed, deliverables are handed over, and resources are released.

Steps:

1. **Confirm Deliverables:**
    - Verify all deliverables meet requirements and are accepted by the client.
2. **Finalize Documentation:**
    - Complete user manuals, technical documentation, and final project reports.
3. **Release Resources:**
    - Return equipment and reassign team members to new projects.
4. **Financial Closure:**
    - Settle all invoices, vendor payments, and release unused budget funds.
5. **Lessons Learned:**
    - Conduct a meeting to discuss successes, failures, and areas for improvement.
6. **Formal Acceptance:**
    - Obtain official sign-off from the client or sponsor.

Example:
In a hospital management system project, closure includes client sign-off, delivery of admin manuals, and archiving source code in a secure repository.

Conclusion:
A proper closure ensures no loose ends remain, enhancing organizational learning for future projects.

Q20. Write a note on configuration audits and its benefits.

Definition:
A configuration audit is a **formal review process** to verify that a product's configuration items match their documented requirements and approved changes.

Types:

1. **Functional Configuration Audit (FCA):**
    - Ensures the system performs according to requirements.
2. **Physical Configuration Audit (PCA):**
    - Verifies the product matches physical design specifications.

Benefits:

1. **Accuracy:** Ensures that deliverables are consistent with documentation.
2. **Quality Assurance:** Confirms that approved changes have been implemented correctly.
3. **Traceability:** Provides a clear record of changes made.
4. **Compliance:** Meets industry or regulatory standards.
5. **Risk Reduction:** Prevents deployment of incorrect or outdated versions.

Example:
In a defense software project, a configuration audit ensures that only approved security modules are included before deployment.

Conclusion:
Configuration audits act as a **quality checkpoint**, ensuring product integrity and reliability before release.