**Unit 1: Basics of Software Project Management**

1. **Define Software Project Management**:  
   It is the process of planning, organizing, leading, and controlling resources to achieve specific software development goals within constraints of time, budget, and quality. It ensures that software projects meet user requirements and are delivered efficiently.
2. **Project Management Life Cycle**:
   * **Initiation**: Define the project’s objectives, scope, and stakeholders.
   * **Planning**: Develop detailed plans, including schedules, budgets, and resource allocation.
   * **Execution**: Carry out the plan, coordinate teams, and produce deliverables.
   * **Monitoring and Controlling**: Track progress, manage risks, and ensure quality.
   * **Closure**: Deliver the completed product and evaluate the project.
3. **Differentiate Traditional and Modern Project Management**:

| **Aspect** | **Traditional PM** | **Modern PM** |
| --- | --- | --- |
| **Approach** | Sequential (e.g., Waterfall) | Iterative (e.g., Agile, Scrum) |
| **Flexibility** | Rigid | Flexible |
| **Customer Involvement** | Limited | High |
| **Risk Management** | Reactive | Proactive |
| **Focus** | Processes | People and collaboration |

**Unit 2: Planning and Estimation**

1. **Work Breakdown Structures (WBS)**:  
   A hierarchical decomposition of tasks required to complete a project. It breaks down the project into smaller, manageable components for better planning and monitoring.
2. **Generic Software Model**:  
   Includes models like:
   * **Waterfall Model**: Sequential approach.
   * **Spiral Model**: Combines iterative and risk analysis.
   * **Agile Model**: Incremental and iterative development.
3. **COCOMO Model**:  
   The Constructive Cost Model (COCOMO) estimates the effort, time, and cost required for software projects based on:
   * **Basic**: For simple, straightforward projects.
   * **Intermediate**: Includes more factors like reliability and experience.
   * **Detailed**: Includes system-specific attributes.
4. **Planning Methods**:  
   Techniques to develop plans:
   * **Critical Path Method (CPM)**.
   * **PERT (Program Evaluation Review Technique)**.
   * **Gantt Charts**.
   * **Milestone tracking**.

**Unit 3: Project Tracking and Control**

1. **PERT Chart**:  
   A graphical representation of tasks, timelines, and dependencies, used for estimating project duration.
2. **Gantt Chart**:  
   A bar chart showing project schedule, with activities and their timelines.
3. **Critical Path Method (CPM)**:  
   Identifies the longest sequence of dependent tasks (critical path) that determines project duration.
4. **Project Metrics**:
   * **Productivity**: Output per unit of effort.
   * **Quality**: Number of defects.
   * **Effort**: Hours or resources consumed.
5. **Earned Value Analysis (EVA)**:  
   A technique to measure project performance:
   * **PV (Planned Value)**: Budgeted value of work planned.
   * **EV (Earned Value)**: Value of work actually completed.
   * **AC (Actual Cost)**: Cost incurred.

**Unit 4: Risk Management**

1. **Define Risk**:  
   An uncertain event that can affect project objectives like cost, time, and quality.
2. **Risk Management Activities**:
   * **Identification**: Determine potential risks.
   * **Analysis**: Assess the likelihood and impact of risks.
   * **Planning**: Develop strategies to mitigate risks.
   * **Monitoring**: Track identified risks.
3. **Categories of Risk**:
   * **Project Risks**: Issues with schedule, budget.
   * **Technical Risks**: Issues with technology, requirements.
   * **Business Risks**: Market changes, financial constraints.
4. **Approaches to Risk Management**:
   * **Avoidance**: Eliminate the risk.
   * **Mitigation**: Reduce its impact.
   * **Acceptance**: Acknowledge and prepare for it.

**Unit 5: Configuration Management**

1. **Software Configuration Management (SCM)**:  
   **Process**: Managing changes in software artifacts during the lifecycle.  
   **Goal**: Ensure consistency and control over versions.
2. **Key Activities**:
   * **Control**: Manage modifications.
   * **Change Control**: Evaluate and approve changes.
   * **Configuration Audit**: Verify compliance with baselines.
   * **Status Reporting**: Track changes and updates.

**Unit 6: Quality Assurance (QA)**

1. **QA and Its Activities**:  
   A systematic process to ensure software quality, involving:
   * Reviews.
   * Testing.
   * Process improvements.
2. **Standards of QA**:
   * **ISO 9001**: Focuses on quality management systems.
   * **CMM (Capability Maturity Model)**: Measures the maturity of processes.
3. **Difference Between CMM and ISO 9001**:

| **Aspect** | **CMM** | **ISO 9001** |
| --- | --- | --- |
| **Focus** | Process maturity | Quality management systems |
| **Goal** | Process improvement | Consistency in product quality |
| **Application** | Specific to software | General across industries |

**Unit 7: Software Re-engineering**

1. **Define Re-engineering**:  
   The process of improving existing software by redesigning or updating components without changing functionality.
2. **Process**:
   * Analyze the system.
   * Reverse engineer the code.
   * Modify and optimize.
   * Re-implement the system.

**Unit 8: Project Closure**

1. **Define Project Closure**:  
   The final phase in a project lifecycle, ensuring all objectives are met, deliverables are handed over, and stakeholders approve.
2. **Elements**:
   * Final deliverables.
   * Documentation.
   * Client approval.
   * Post-project evaluation.