ct) that contain graphs, charts, counters, and ...nals to real-time data such as ...d defects,

GoodLuck | Page No. 1
Date

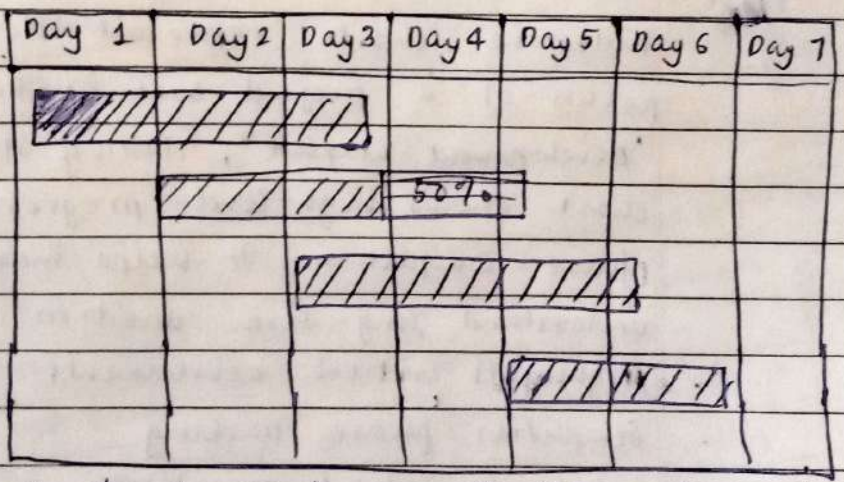
Software Project Management

Unit 04 Project Tracking, Monitoring and Control

Q.36) What are the different methods used in visualizing progress. Explain in detail

Ans Project progress visualization means representing the current status of a project in a graphical format, so that managers & stakeholders can easily understand how far the project has reached in terms of schedule, cost, and quality. Visualization tools make the project transparent, measurable & trackable.

1. Gantt charts



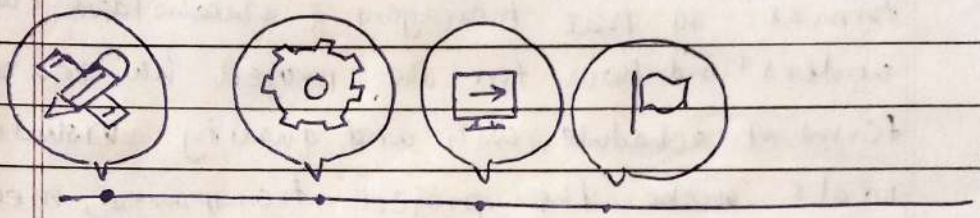
A Gantt chart is the most commonly used method for project visualization. It shows tasks in the form of horizontal bars along a timeline. It helps the manager understand when a task will start, when it will end, and how much progress has been made. Gantt charts visually clarify the overall project flow.

GoodLuck | Page No.
Date

- Start / End Dates visible
- Parallel activities trackable
- Delays easily identified
- % completion visible.

e.g. A bar for "UI Design – 6 days" is displayed, and if 3 days of work are completed, the bar will show 50% filled.

2. Milestone Charts



Milestone charts represent the major progress points of a project such as "Design complete", "Development Complete", "Testing Start". This chart shows high-level progress and visualizes phase completion. It helps managers understand long-term direction.

- Highlights critical achievements
- Simplifies phase tracking
- Good for client reporting
- Shows major events only

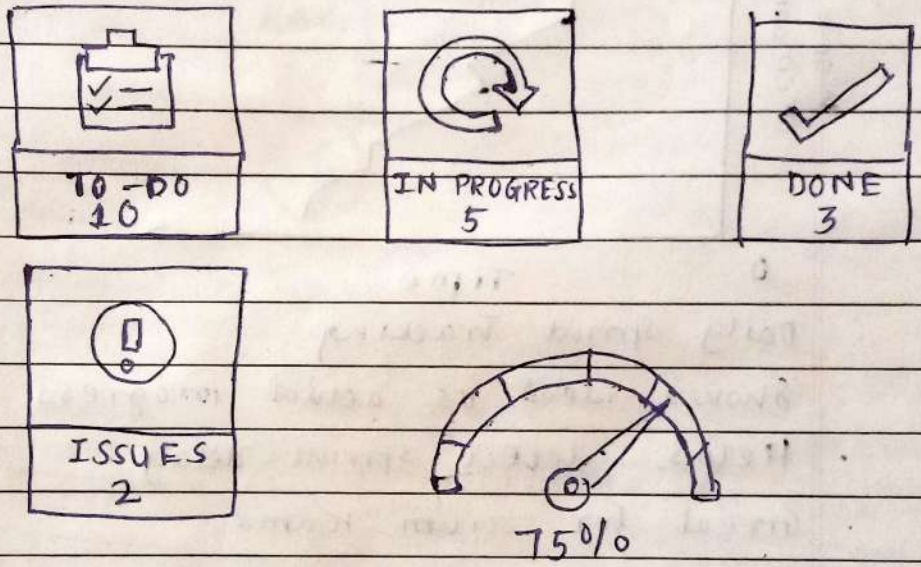
e.g. Week 4: "Prototype completed" milestone achieved means: the team is on track.

3. Dashboards

Dashboards are available in modern project management tools (Jira, Azure DevOps, MS

Project) that contain graphs, charts, counters, and indicators. They visualize real-time data such as tasks pending, completed, delayed, defects, workload, burndown, etc. A dashboards shows the entire project status on a single screen.

DASHBOARD



- Real-time visualization
- Interactive charts
- Customizable widgets
- Ideal for daily monitoring

e.g If a dashboard shows 10 To-Do, 5 In-Progress, 3 Done tasks, then the manager can instantly understand the status.

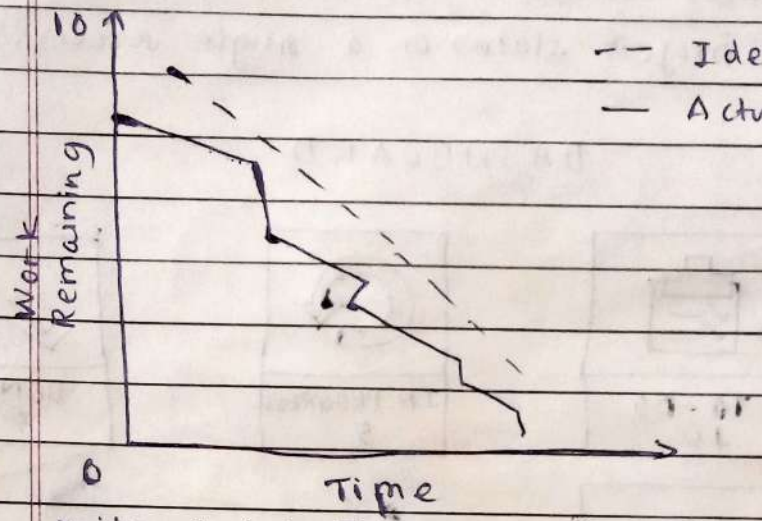
4) Burndown and Burnup charts (Agile visualization)

A Burndown chart shows how much work remains and the speed at which the sprint is being completed. A Burnup chart represents the amount of work completed. These charts are extremely useful in Agile projects because they clearly

GoodLuck | Page No.
Date

show, whether the team is ahead or behind schedule.

BURNDOWN CHART

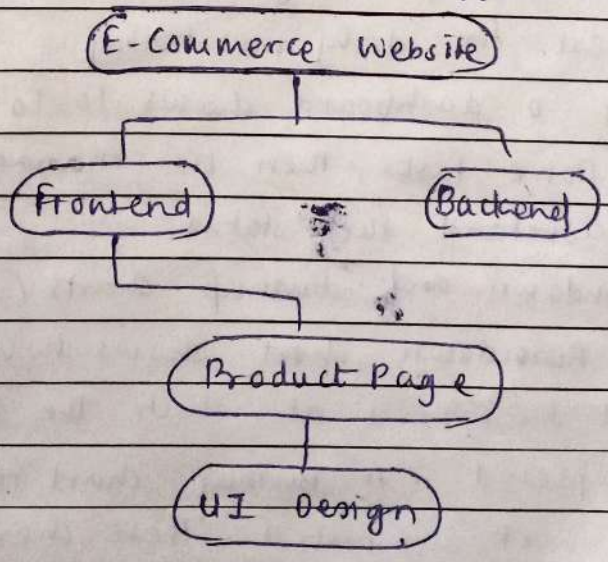


- Daily sprint Tracking
- Shows ideal vs actual progress
- Helps detect sprint delay
- Great for scrum teams

e.g. If the burndown chart becomes flat, it means team progress is slow.

5. Work Breakdown Structure (WBS) Charts

WBS is a hierarchical breakdown where



a project is divided into smaller,

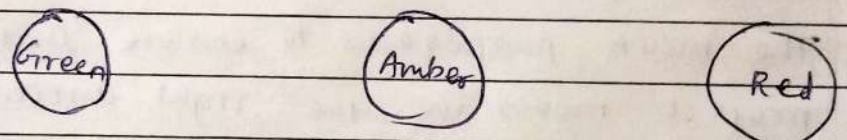
manageable tasks. In chart form, WBS visualizes the project structure and clarifies complexity.

- Clear task hierarchy
- Better task assignment
- Improves planning
- Shows module-wise progress

e-g. E-commerce Website → Frontend → Product Page → UI Design hierarchical tree

6. Traffic Light Indicators (RAG - Red, Amber, Green)

RAG Status Indicator



On Track Warning Critical

RAG Indicators are a simple visualization method where RED = problem, AMBER = warning, GREEN = on track. Stakeholders quickly understand project health. This method is useful in executive-level dashboards.

- Extremely simple
- Instant status visibility
- Good for top management
- Helps classify project risk

e-g. Backend module = RED (delayed), UI module = GREEN (on track)

Visualization methods make the project crystal clear. Gantt charts, milestone charts, dashboards, burndown charts, and RAG indicators visually represent project progress, making monitoring faster and more effective.

GoodLuck | Page No.
Date

(Q 4a) What is project control? Explain the different types of control mechanism in details.

Ans Project control is a continuous process in which the actual progress of the project is compared with the planned progress, and if any deviation is found, corrective actions are taken.

Project control = Check what was planned → see what is happening → identify problems → Take action → Bring the project back on track.

The main purpose is to ensure that the project moves in the right direction in terms of time, cost, quality, and scope.

- Need

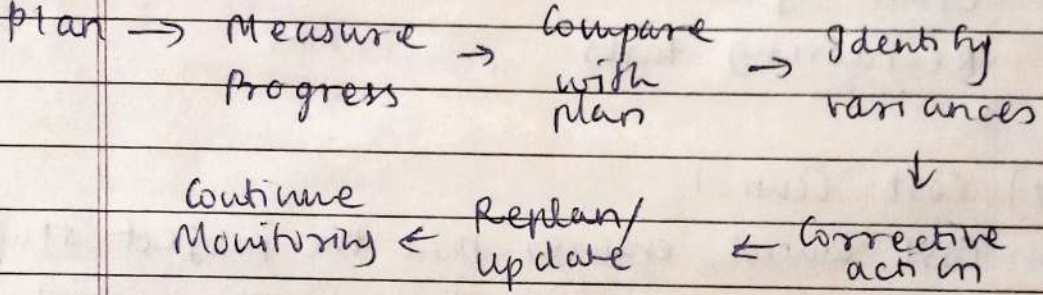
Project control is important because:

- Real projects never run exactly according to plan.
- Delays, cost increases, and requirement changes always occur.
- Without control, the chances of project failure are very high.
- It is easier to detect problems at an early stage.
- Helps the manager predict future performance.
- Ensures customers receive quality work on time.

- Objectives

1. Keep the project moving in the planned

2. Identify deviations in schedule, cost, quality, and resources.
3. Take corrective actions on time
4. Use resources efficiently.
5. Maintain quality and performance
6. Reduce risks and avoid project failures
7. Ensure customer satisfaction.



~~Project Control Process~~

* Control Mechanisms in Project Management

1) Schedule control

Schedule control ensures that the project activities are completed according to the planned timeline. The project manager regularly tracks the start and finish dates of activities, monitors milestones, and updates Gantt charts. Techniques like CPM/PERT help to identify the exact source of the delay. If the project is behind schedule, corrective actions like: fast-tracking (parallel work) or crashing (adding extra resources) are taken.

Purpose: Ensures timely completion.

* Working:

GoodLuck | Page No.
Date

- Tracking activity start-end dates
- Updating Gantt charts
- Monitoring milestones
- Checking delays using CPM/PERT
- Calculating SPI (Schedule Performance Index)
- Corrective actions
  - Fast-tracking
  - Crashing
  - Reordering tasks

2) Cost Control

Cost control ensures that the project stays within the allocated budget. The project manager compares planned cost vs actual cost and calculates cost variance. Earned Value Management (EVM) tools such as CPI (Cost Performance Index) helps analyze cost deviations. If the project is going over budget, unnecessary expenses are reduced, resource distribution is optimized, and future spending forecasts are revised.

Includes:

- Comparing planned and actual cost
- Using EVM
- Calculating CPI
- Forecasting future budget
- Corrective actions
  - Cost cutting
  - Controlling over-budget tasks

handle risks, and lead the project towards successful completion.

Ro- Solid plan What

GoodLuck | Page No. 5 | Date

3) Quality Control

Quality control ensures the deliverables meet the required quality standards. This mechanism uses testing, reviews, inspections, audits, and checklists to ensure quality compliance. Identifying issues easly reduces rework and time loss. Quality control improves customer satisfaction and project credibility.

- Working
- Testing
- Reviews and inspections
- Quality audits
- Using checklists
- Identifying defects

Result: Improved quality and reduced rework.

4) Resource Control

Resource control involves managing human, material, and equipment resources efficiently. Workload balancing and resource leveling ensure no resource is overburdened or underutilized. Additional resources may be assigned if delays occur. It also helps avoid conflicts and shortages.

- Includes:
- Checking team workload
- Resource leveling
- Preventing overload
- Maintaining availability of tools and equipment.

Goal: Resources should be neither overused nor underused

...ndle risks, and lead the project toward
...tion.

what ...dule

GoodLuck | Page No.
Date

5) Change Control

Change control manages any change in scope, requirement, or design systematically.

* Steps:

1. Change Request is raised.
2. Impact analysis (on cost, time, quality)
3. Approval from CCB (Change Control Board)
4. Change implementation
5. Updating baselines

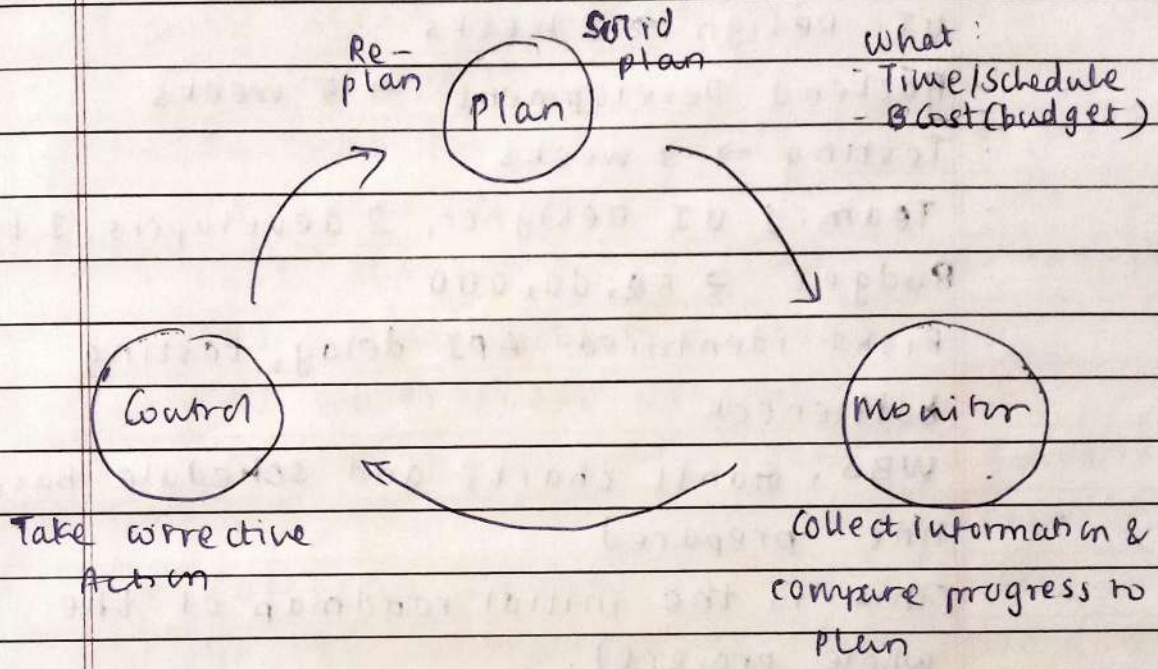
Purpose: Prevents unauthorized changes and scope creep

Project control is the backbone process that keeps the project on track. Schedule, cost, quality, resource, and change control mechanisms ensure that the project is completed in the right direction, with the right quality, and within the right budget.

Q. 4 b) Explain plan monitor control cycle used in the project in detail with example.

Ans. The Plan - Monitor - Control (PMC) cycle is a continuous feedback loop in project management whose main objective is to keep the project moving in the planned direction. This cycle ensures that after planning, the execution does not run randomly, but every stage is continuously monitored, and if any deviation (like delays, cost increases, or quality issues) is found, immediate

control actions are taken. The PMC cycle provides a systematic approach that enables the project manager to make timely decisions, handle risks, and lead the project toward successful completion.



* Stages:

1) PLAN Phase

The Plan phase is the foundation of the project where the entire project roadmap is created. In this phase, the project manager defines what needs to be done (scope), how it will be done (process), ~~for~~ when it will be done (schedule), how much it will cost it will involve (budget), who will do it (resources), and what quality standards need to be maintained. Outputs of the planning phase include Work Breakdown Structure (WBS), Gantt chart, milestone list, cost baseline, risk management plan, and communication plan.

GoodLuck | Page No.
Date

The main goal of the Plan phase is to create a clear and achievable project blueprint that can be followed during execution

e.g. Consider a mobile App development project with a timeline of 10 weeks.

- UI Design → 2 weeks
- Backend Development → 5 weeks
- Testing → 3 weeks
- Team: 1 UI Designer, 2 developers, 1 tester
- Budget: ₹ 50,00,000
- Risks identified: API delay, testing bottleneck
- WBS, Gantt chart, and schedule baseline prepared

(This is the initial roadmap of the whole project).

2. MONITOR Phase

In the Monitor phase, the actual progress of the project is continuously observed. The project manager tracks the status of activities, compares the actual schedule with the planned schedule, checks cost expenditure, verifies quality, and reviews resource utilization. For monitoring, tools and techniques ~~like~~ such as Gantt chart updates, milestone tracking, review meetings, dashboards, status reports, and Earned Value Management (SPI/CPI) are used.

The purpose of Monitor phase is to identify deviations at an early stage so that problems can be controlled before they increase.

e.g. By week 4, monitoring reveals:

- UI completed
- Backend progress = only 2 weeks work done (1-week delay)
- Cost spent = ₹ 2,20,000 (higher than planned)
- SPI = 0.85 (schedule slow)
- CPI = 0.91 (cost overrun)

(Monitoring clearly identifies deviations.)

3. CONTROL Phase

In the control phase, corrective and preventive actions are taken to solve the deviations or issues identified during monitoring. If the project is behind schedule, tasks may be fast-tracked, or
- additional resources may be allocated.

If a cost overrun occurs, the budget is adjusted or unnecessary expenses are reduced. If a quality issue appears, additional testing or reviews are performed. In this phase, chage requests are approved and project baselines are updated. The main purpose of the control phase is to bring the project back to the planned path.

eg Corrective actions taken by the manager:

GoodLuck | Page No.
Date

- One more developer added to the backend team
- Some early tasks assigned to the testing team
- Review meetings increased from weekly to twice a week.
- Schedule cost and baseline adjusted.

Result: within the next 2 weeks, the project is back on track.

The PMC cycle is the backbone of project management, ensuring that the project is completed within planned scope, time, and cost constraints. The Plan phase provides direction, the Monitor phase provides real-time insight, and the Control phase corrects deviations. Through this cycle, the project manager continuously guides the project and ensures successful delivery.

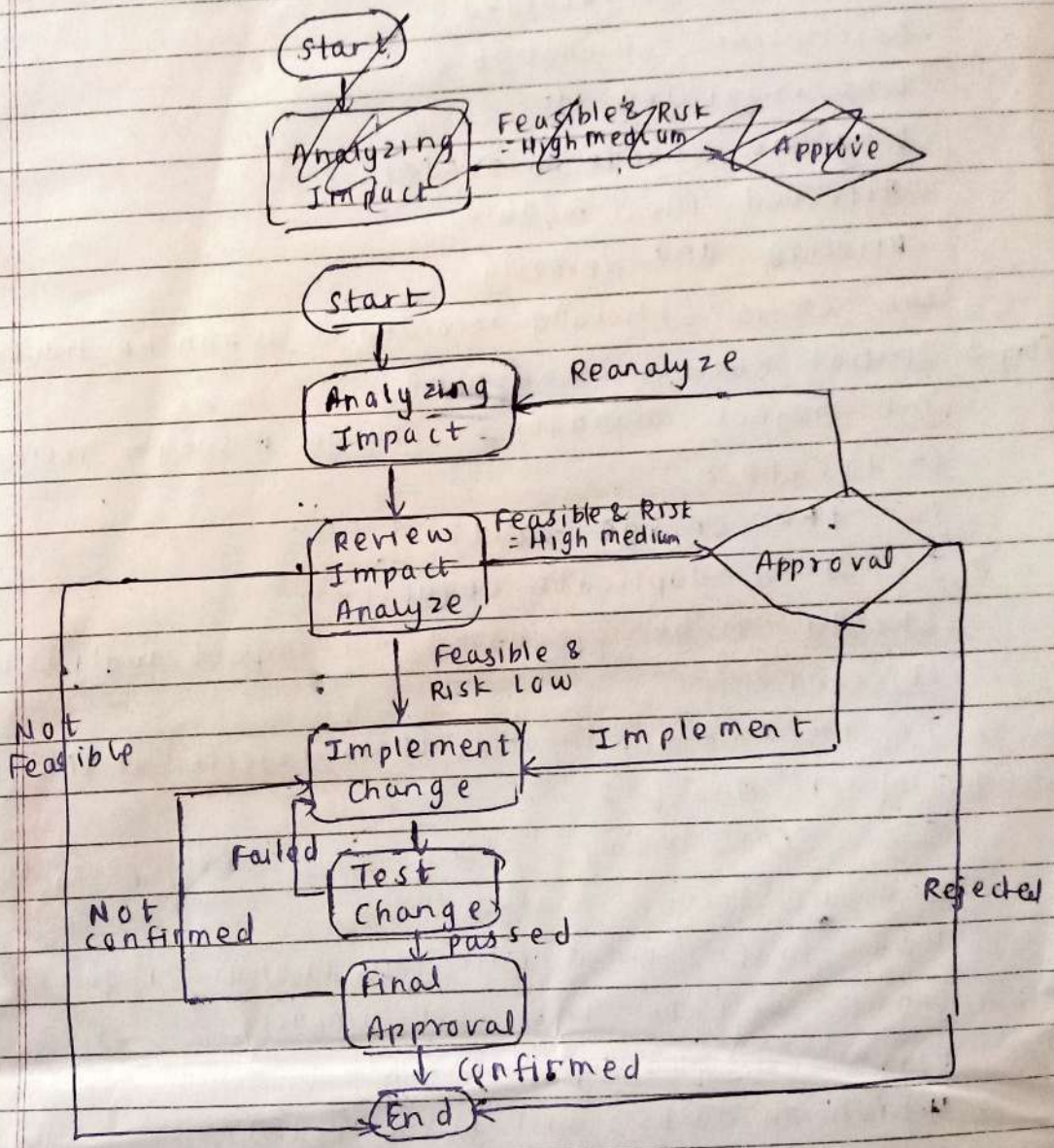
May-June 2023

Q.3 a) Draw & explain the flowchart for change control function in SCM

Ans Change control is a core process of Software Configuration Management (SCM). Its main purpose is to:

- Control changes made to the software

Analyze their Impact
Approve only authorized changes
Track and implement approved changes effectively.
Change control ensures that unauthorized modifications, scope creep, and project instability are avoided.



Step 1 Change Request (CR) Creation

The Change Control process begins when any stakeholder (developer, tester, client, or manager) raises a Change Request (CR). A CR may be raised for a bug fix, new feature, improvement, or required modification

"Need a change → Submit CR form → Process starts"

Step2 Logging the Change Request

The change request is logged in an SCM tool (such as Git, Jira, Bugzilla, SVN, etc.) At this stage, the CR is assigned a unique ID and detailed information is recorded:

- Description of change
- Who requested it
- Why the change is needed
- Affected files/ modules
- Priority and severity

The CR is officially recorded so it can be tracked

Step3 Initial Review / Screening

The Project Manager or Change Manager reviews the CR to check:

- Is the change valid?
- Is it a duplicate request?
- Should it be processed for impact analysis or be rejected?

If the CR is invalid, it is rejected at this stage.

Step4 Impact Analysis

The SCM team, along with developers, testers, and architects, analyze:

- How much time will the change require?
- What will be the cost impact?
- Effect on quality....
- Which modules will be affected?
- Risks involved
- Need for additional resources

This step determines whether the change is feasible or not.

What will happen due to the change? Time? Cost? Risks? Everything is evaluated here.

Step5 Change Approval by CCB (Change Control Board)

Based on the impact analysis, the CCB decides

- Approve the change
- Reject the change
- Defer (postpone the change)

The CCB includes the project manager, senior developers, quality manager, and client representatives
Approval is mandatory - no change is allowed without CCB.

Step 6 Change Implementation

Approved changes are assigned to the development team

Developers modify the code, testers verify the changes, and the SCM team maintains the updated version. This stage includes:

- Coding
- Testing
- Integration
- Documentation updates.

Step 7 verification and Validation

After implementation

- Testing team confirms that the change works correctly
- Ensures no other module is damaged
- Regression testing is performed

Is the change working correctly? If yes, proceed, if not, fix again.

Step 8 Closure of change Request

If the change is successful, the CR is marked as Closed.

Documentation is updated, version numbers are adjusted, and final status recorded

Step 9 Reporting and Auditing

The SCM team maintains change logs, history, and version updates.

Audits verify that the change went through a controlled process.

Records are maintained for future traceability

Q.4 a) Write short notes on Team Structure and Virtual Team in a Software Project

Ans 1) TEAM STRUCTURE

Team structure refers to the arrangement of reporting, responsibility distribution, and communication within a project team.

It defines:

- Who will report to whom
- How work will be distributed
- How communication will take place
- What the flow of decision-making will be

A) Chief Programmer Team Structure

- At the top is the chief programmer / project manager
- Below are multiple team members
- Communication is top-down
- Team members report to the chief programmer
- Design, coding, and testing tasks are assigned by the chief programmer.

In the Chief Programmer Team Structure, a highly skilled senior programmer leads the entire team. They create the high-level design, define specifications, and takes all major technical decisions. Other team members are assigned smaller tasks such as coding, testing, and documentation. The chief programmer integrates everyone's work to prepare the final solution. This structure is effective for small and simple projects where a strong technical leader can guide the project quickly.

* Advantages

- Chief programmer can quickly ~~asste~~ finalize design and assign work
- High efficiency because authority is clear
- Best for small teams (5-6 members)

- Strong control and discipline
- Faster work due to centralized decision-making

* Disadvantages
- Excessive power may lower morale of team members
- Limited creativity and innovation
- Single person may become bottleneck due to workload
- Not suitable for large projects
- High dependency - project may stop if the chief programmer becomes unavailable

B) Democratic Team Structure
- All team members are at equal levels
- Communication occurs freely among all members
- No strict hierarchy
- Decisions are group-based through discussion

In a Democratic Team structure, there is no formal hierarchy. All members can directly communicate with each other. Discussions, debates, and brainstorming sessions are common. Technical leadership roles may rotate based on expertise. This structure is best suited for research projects, creative tasks, and small collaborative teams.

* Advantages
- High team morale
- Better job satisfaction
- More productivity and creativity
- Promotes ego-less programming
- Multiple ideas lead to better solutions.

* Disadvantages
- Discussions may become chaotic in large projects
- Decision-making is slow
- Frequent disagreements

- Useful for research projects but complex in large software development

c) Mixed Control Team Structure

- Top: Project Manager
- Middle: Senior Members / Module Leaders
- Bottom: Team Members
- Communication: Democratic horizontally, hierarchical vertically

Mixed control team is a combination of hierarchical and democratic structures Senior members manage reporting, but horizontal communication remains flexible. Projects are divided into manageable sub-units. Senior developers lead sub-teams, and decision-making is partially democratic.

* Advantages
- Balanced combination of creativity and control
- Effective for large projects
- Even distribution of work
- Smooth but controlled communication
- Clear responsibility and accountability

* Disadvantages
- Higher coordination effort required
- Communication gaps may occur between senior members
- Too complex for small teams

2) VIRTUAL TEAMS

A virtual Team is a team where members work from different locations, cities, countries, or time zones. Work coordination happens entirely through online tools

- Geographically distributed: Members work remotely - e.g., team members from India, USA, UK collaborating

- Collaboration tools: Zoom, Microsoft Teams, Slack, Jira, GitHub, Trello, Email
- Flexible Timing: Members work according to their time zone
- Technology-driven communication: All interaction happens through virtual meetings, chat, and shared documentation

* Advantages
- Ability to hire global talent
- Reduced office costs
- 24x7 productivity through time-zone rotation
- Higher employee satisfaction (remote work flexibility)
- Faster delivery due to parallel workflows

e.g. India team codes during daytime → US team tests at night → work never stops

* Challenges
- Communication delay due to time-zone differences
- Lack of bonding and trust
- Cultural and language differences
- Difficult to monitor performance
- Home distractions

Q.3 Nov-Dec 2023

Q3 a) Explain the data visualization tools: Kanban boards, project calendars, and earned value charts. What are the two main components of Project tracking and control?

Ans. Data visualization tools present project planning, tracking, and monitoring in a graphical format. Kanban boards, Project Calendars, and Earned Value Charts are the most widely used visualization tools in project management.

1) Kanban Boards

A kanban board is a visual workflow management

tool where all work is divided into columns on a board. Each task is represented as a card, and as work progresses, the card moves from left to right across the board.

Basic Principal:

"Visualize the work → Limit the work → Improve the flow"

	Backlog	In-Progress	To-DO	Done
Web Team	3rd Party Integration	UI Improvements	On-Premise Backup	Code Review
	New Admin Console		Product Review	Shopping Cart Improvements
	Security 2.0		self-service portal	
Mobile Team	Android Application	Application upgrade	Cloud support	Q3 Priorities list
	Mobile Mock up	Ticketing System	~~Interactive~~ Dialog Box	~~Ticketing~~ System
	~~UX Improvements~~			
Marketing Team	Customer Outreach	Pricing Review	SEO plan	Content Review
	Lead Gen			Customer Outreach
	Market Analysis			Market Analysis
				Performance Management
				Proactive Email Campaign
				SEO plan

Page No. --

Page No. 12
Date

Columns: To Do, In Progress, In Review, Testing, Done

Each card represents a work item (bug, task, feature, test case, etc.)

- Visual Workflow Representation: The team instantly understands pending, on going & completed tasks
- WIP Limits (Work - In Progress Limits): Only a limited number of tasks are allowed in a column to avoid overload.
- Pull System: A developer picks the next task only when they are free.
- Real-time Progress Visibility: Live task status visible to the entire team.
- Team Collaboration: Issues, blockers, or priority changes are immediately visible.
- Advantages: Transparent work visibility, quick identification of delays, Improved team coordination, bottlenecks easily visible, smooth task flow.

e.g. In a software testing team's Kanban board:
"Login test case" → In progress
"Signup validation Bug" → To Do
"Payment Gateway Testing" → Done
The team can instantly understand the status of each task.

2) PROJECT CALENDARS

Dec 29 - Jan 4 2020

	Sun, 29	Mon, 30	Tue, 31	Wed 1,	Thu, 2	Fri, 3	Sat, 4
8.00		Meeting Conceptual Design	Math model prototype	Art design Meeting client	Implement product test product	Improve product Test product	Delivery to client
9.00		Morning Meeting Analysis Requirement		Meeting Client Review Prototype Discuss features	Implement Product Finish product		Deliver product to client

	29	30	31	1	2	3	4
10.00		Design Math Model					
11.00							
12.00	Conceptual Design write design document		Art design		Improve & test product		
13.00							
14.00		Make Simple prototype					
15.00				Test the product			

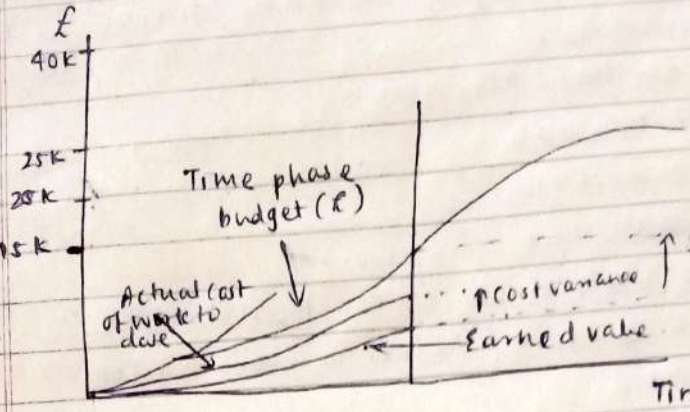
A Project Calendar is a time-based visualization tool that displays: Daily tasks, resource schedules, deadlines, meetings, holidays, team availability. Main purpose is to make time and resource planning visually simple and well organized.

* They show: Task start & end dates, weekly & monthly schedules, team availability status, leave & holiday management, client meeting dates, sprint timelines, deliverable deadlines.
- Day-wise schedule clarity: Each day's work clearly visible.
- Workload balancing: Helps evaluate if any team member is overloaded
- Clash detection: Prevents overlapping tasks on same resource
- Reminder Notifications: Reduces the risk of missing deadlines.
- Team-level visibility: Everyone knows who is doing what and when.

e.g. A project calendar may show
- 15 Feb - Client Review Meeting
- 18 - 20 Feb - UI Testing

- 25 Feb - Payment Module Deadline

3) Earned Value Charts



An Earned Value Chart is a performance visualization tool that displays the health of project cost and schedule on a single graph.

* It includes three lines:

1. PV (Planned Value) - Work expected to be completed by this time
2. EV (Earned Value) - Work actually completed
3. AC (Actual Cost) - Actual ~~or~~ money spent

This provides an instant project performance view.

* Components

- Planned Value (PV) - Shows expected progress based on project timeline.
- Earned Value (EV) - Monetary value of completed work
- Actual Cost (AC) - Budget actually spent
- SPI (Schedule Performance Index).

  $SPI = \frac{EV}{PV}$ → shows delay or ahead performance.

  CPI (Cost performance Index)

  $CPI = \frac{EV}{AC}$ → shows over-budget or under budget status

* They show:

- Project chart is ahead or behind schedule
- Whether cost is overspending
- Planned vs actual comparison
- Real-time performance trends
- Need for corrective action

* Advantages: Accurate performance measurement, combined analysis of cost + schedule, early detection of deviations, predict future project condition, accepted by PMI and global standards

e.g. Assume at the 4th week:

- PV = ₹ 100,000
- EV = ₹ 80,000
- AC = ₹ 90,000

So, SPI = 0.80 → 20% project delay

CPI = 0.88 → Project is over-budget

In the chart,

EV curve will be below PV

AC curve will be above EV → indicates an unhealthy project condition

** Two Main components of project tracking & control

1. Schedule Tracking (Time Tracking)

This focuses on whether the project tasks are being completed in time as per the planned schedule.

Checks planned vs actual progress

Monitors delays or tasks running behind schedule

Uses tools like Gantt charts, Project calendars, Kanban Boards, Milestone charts

2. Cost Tracking (Budget control)

This focuses on whether the project is spending within the approved budget

Compares planned budget vs actual expenditure

Controls overspending & resource cost usage

Uses techniques like Earned Value Management (CPI, CV, AC, & EV)

Q.16) Discuss the different change request types and the change approval process

Ans. A Change Request (CR) is a formal proposal for making

modifications in a software. It helps keep the system error-free, updated, optimized, and compliant. In software projects, there are generally 6 major types of Change Requests.

In software development, requirements, design, technology, environment, security, or user needs frequently change.

The process of formally requesting, analyzing, approving, and implementing these changes is called Change Request Management.

A CR is a formal proposal to modify:

- Requirements
- Code
- Design
- Documents
- Interfaces
- Database
- Test cases
- Configuration items

The CR process ensures software stability, traceability, and controlled evolution.

* TYPES

1) Corrective Change Request

- Raised for bug fixing or error correction
- Identified during testing or through user feedback
- Restores system to its expected behavior
- Usually high-priority

A Corrective Change Request is raised when a bug, failure, crash, or any error is detected that impacts the expected functioning of the system. These changes are usually found during testing phases such as UAT (User Acceptance Testing) or during real production ~~stage~~ usage. The primary goal is to bring the system back to its

originally correct state. It is the most common type of CR and often requires immediate fixing, especially in mission-critical modules.

2) Adaptive Change Request

- To adjust the system to a new environment
- Required after OS upgrade, hardware change, or browser updates
- Triggered by external dependency updates.

An Adaptive Change Request is raised due to changes in the software environment - for example, operating system upgrade, hardware replacement, changes in network architecture, updated legal regulations, or third-party API version changes. It ensures that the software remains compatible with the new environment. Without adaptive changes, the system may malfunction or fail.

3) Perfective Change Request

- For performance improvement
- UI / UX enhancement or code optimization
- Adding new or improved features
- Enhances system usability

The primary goal of a Perfective Change Request is to improve the system - not because something is wrong, but to make it better. It includes UI enhancements, code refactoring, database tuning, feature extensions, and performance optimizations. It improves user experience and helps make the software more efficient, modern, and user-friendly.

4) Preventive Change Request

- To avoid future problems
- Fixes vulnerabilities
- Reduces risk and preserves long-term performance
- Ensures long-term stability

A Preventive Change Request aims to prevent potential future issues It includes removal of technical debt, outdated code cleanup, fixing security vulnerabilities, and replacing risky components. These proactive changes help prevent future system crashes, failures, or cyberattacks.

5) Emergency Change Request
- For highly urgent situations
- Production outage, critical bug, or security breach
- Fast-track approval
- Implemented immediately

An Emergency Change Request is used when there is a critical failure, severe security breach, or production system outage. Time is crucial, so normal approval workflows are bypassed and an expedited approval process is used. The main objective is to restore system stability as quickly as possible to prevent business loss.

6) Regulatory / Compliance Change Request
- Based on government rules or legal compliance
- Common in financial, telecom, and banking industries
- Mandatory changes
- Non-compliance may lead to penalties

A Regulatory Change Request is raised when the software must comply with new laws, government policies, industry standards, or regulatory guidelines Examples include change in tax laws (like GST), GDPR privacy rules, banking security standards, or telecom regulations. These changes are mandatory and failure to implement them can lead to legal consequences

Q4a) What are common methods of collecting project

data from stakeholder? Explain how to calculate earned value and use it to calculate Schedule Performance Index (SPI) and cost performance index (CPI)

Ans: Common Methods of Collecting Project Data from Stakeholders

Project managers use various techniques to collect information from stakeholders, ensuring accurate understanding of requirements, expectations, risks, and feedback.

- Interviews
  - One-to-one or group discussions with stakeholders
  - Helps gather detailed qualitative information, expectations, and assumptions
- Surveys / Questionnaires
  - Set of standardized questions sent to multiple people
  - Useful when many stakeholders must provide input efficiently
- Workshops / Brainstorming sessions
  - Interactive sessions to gather ideas, clarify conflicts, and build consensus
- Focus Groups
  - A facilitated discussion with selected stakeholder representatives
  - Helps understand opinions, experiences, and preferences
- Observation
  - Project team observes users interacting with current systems or processes
  - Helps understand real workflows and existing pain points
- Document Analysis
  - Reviewing existing reports, contracts, policies, system logs, proposals, etc.

- Gives historical and technical understanding
- Prototyping
  Creating mockups or demo models to gather feedback on functionality and design
- JAD (Joint Application Development) sessions
  Collaborative Meetings involving business stakeholders and technical teams
- Helps define requirements quickly and accurately

* Calculate Earned value
Earned value (EV) represents the value of work actually completed at a specific time compared to the approved project budget.

EV = % of work completed × Total Budget (BAC)

e.g. Total project budget (BAC) = $100,000
Completed work = 40%

EV = 0.40 × 100,000 = $40,000

* Schedule Performance Index (SPI)
SPI measures project schedule efficiency - whether the project is ahead or behind schedule.

SPI = EV ÷ PV

EV = Earned value
PV = Planned value (budgeted cost of planned work)
SPI = 1 → On schedule
SPI > 1 → Ahead of schedule
SPI < 1 → Behind schedule

e.g. EV = $40,000
PV = $50,000

$$SPI = \frac{40,000}{50,000} = 0.80$$

∴ Project is behind schedule (80% efficiency)

4. Cost Performance Index (CPI)

CPI measures cost efficiency - whether the project is overspending or saving money

$$CPI = EV - AC$$

AC = Actual cost spent for completed work

CPI = 1 → On budget

CPI > 1 → Under budget (cost - saving)

CPI < 1 → Over budget

e.g. EV = $40,000

AC = $45,000

$$CPI = \frac{40000}{45000} = 0.89$$

:Project is over budget (only 89 cents of value earned per dollar spent).

Q.4b) Explain the different SCM tools and how they are used to track and manage changes to software artifacts

Ans. SCM (software config mgmt) tools are used to track, control, and manage changes to s/w artifacts such as source code, documents, configuration files, databases, test scripts, and design specifications. They help maintain version control, support team collaboration, ensure traceability, and prevent conflicting changes.

1. Git

Git is a distributed version control system widely used in modern software development.

- To racks changes to source code and merges updates from multiple developers.
- Maintains a complete change history through commits
- Allows branching and merging for parallel development

Supports remote repositories through platforms like GitHub, GitLab, BitBucket

- Enables rollback to previous versions if issues occur

2. Subversion (SVN)

SVN is a centralized version control system

- Maintains a central repository for storing project versions
- Tracks and logs changes with commit history
- Allows controlled access and locking for large binary files
- Enables version comparison and rollback features
- Used in enterprises where central control is required

3. Mercurial

Another distributed version control tool similar to Git

- Helps manage versions of source code and documentation
- Supports branching, merging, and distributed repositories
- Keeps detailed change history and efficient performance
- Used in large-scale projects requiring high speed (e.g., game engines).

4. IBM Rational ClearCase

Enterprise-level SCM tool used in large and complex projects

- Manages versioning for large configuration items and system components
- Tracks baseline configurations and supports parallel development
- Integrates with build and test automation systems
- Provides strict process control and auditing

5. Microsoft Azure DevOps / TFS (Team Foundation Server)

Version control and project management platform from Microsoft

- Provides Git or TFVC (Team Foundation Version Control)
- Tracks code changes, work items, defects, builds, and releases
- Integrates with CI/CD pipelines and testing tools
- Used in agile and DevOps environments.

6 Perforce Helix Core
SCM tool designed for high-performance enterprise environments
- Tracks changes to large and complex codebases
- Efficiently handles big binary assets (e.g., videos, game files, CAD models)
- Supports centralized repository and strong security control
- Provides history, branching, merging, and access control features

7 CVS (Concurrent Version System)
An older centralized version control system
- Tracks and stores software artifact ~~versions~~ revisions
- Supports developers working in parallel
- Maintains historical logs for auditing and traceability

* How SCM tools track & manage changes
- Version control: Maintains history of all changes to artifacts
- Change tracking: Logs who made changes, what changed, when, and why
- Branching & Merging: Allows parallel development and integration
- Baselining: Creates stable reference points for releases
- Access control: Manages user permissions to protect source code

- Conflict Resolution: Handles merge conflicts when multiple people change same files
- Rollback and recovery: Restores previous versions in case of errors
  Audit and traceability: Generates detailed reports for verification and compliance

Nov - Dec 2024

Q3a) Explain the different types of contracts List the various typical terms of a contract

Ans. Contracts are legally binding agreements between two parties in project management & procurement. They define the responsibilities, payments, and deliverables involved in a project. The main goal is to ensure clarity, control cost, & reduce risks.

* Types: -

1. Fixed - Price / Lump - Sum Contracts
   In this contract, a fixed total price is agreed upon for the entire project scope before the work begins
   - Cost does not change unless scope changes
   - Low risk for buyer; high risk for seller
   - Suitable when requirements are clear and stable
2. Cost - Reimbursable Contracts
   The buyer pays the seller for actual costs incurred plus an additional fee or profit
   - Flexible when scope is unclear
   - High risk for buyer, low risk for seller
   - Used in research, R&D projects, or uncertain work

   Types:
   CPFF (Cost Plus Fixed Fee): seller is paid cost + fixed fee
   CPIF (Cost Plus Incentive Fee): seller receives incentive for cost savings
   CPAF (Cost Plus Award Fee): Award fee based on performance

3. Time and Material (T&M) Contracts
Buyer pays for the time spent and materials used
- Cost based on hourly / daily rate + material cost
- Balanced risk for both parties
- used when work cannot be estimated accurately. (e-g., maintenance, consulting)

* Typical Terms of a Contract
A contract usually includes formal terms that define expectations, rules, pricing, and legal protection.
- Scope of Work (SOW): Defines what work will be performed
- Deliverables: Items or results to be delivered
- Schedule / Timeline: Start date, milestones, deadlines
- Payment Terms: Cost, billing, invoices, penalties
- Responsibilites of parties: Roles and obligations of buyer and seller
- Quality Requirements: Standards and acceptance criteria
- Change Control Process: Process to handle modifications
- Confidentiality / NDA: Protection of sensitive information
- Warranty and support: Maintenance and responsibility
- Termination Clauses: Conditions to cancel the contract
- Risk, Liability & Insurance: Damage coverage and responsibility
- Dispute Resolution: Arbitration, mediation, legal actions
- Intellectual Property & Rights: Ownership of product or code.