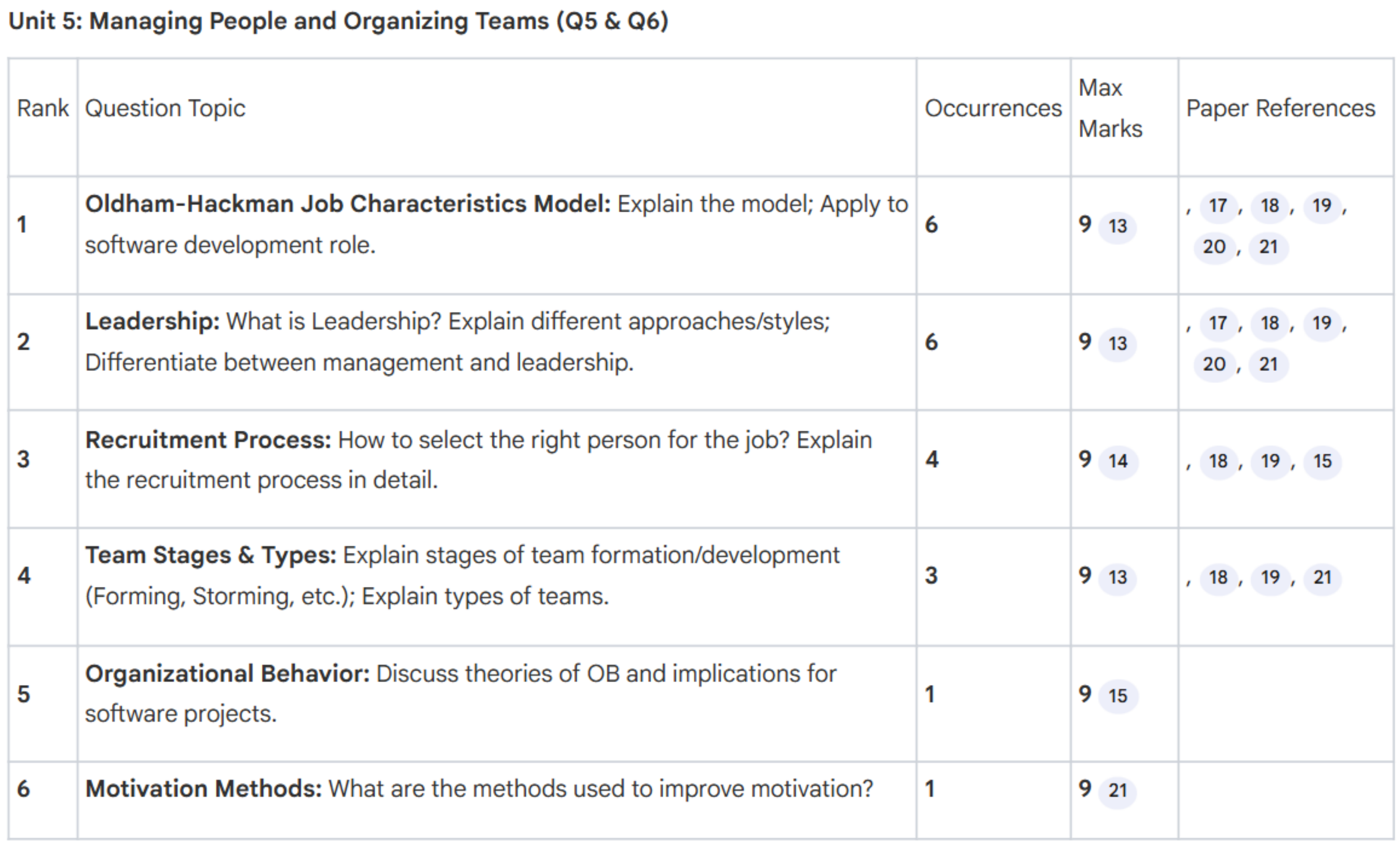
UNIT 5



Unit 5: Managing People and Organizing Teams (Q5 & Q6)

Rank	Question Topic	Occurrences	Max Marks	Paper References
1	**Oldham-Hackman Job Characteristics Model:** Explain the model; Apply to software development role.	6	9 13	, 17 , 18 , 19 , 20 , 21
2	**Leadership:** What is Leadership? Explain different approaches/styles; Differentiate between management and leadership.	6	9 13	, 17 , 18 , 19 , 20 , 21
3	**Recruitment Process:** How to select the right person for the job? Explain the recruitment process in detail.	4	9 14	, 18 , 19 , 15
4	**Team Stages & Types:** Explain stages of team formation/development (Forming, Storming, etc.); Explain types of teams.	3	9 13	, 18 , 19 , 21
5	**Organizational Behavior:** Discuss theories of OB and implications for software projects.	1	9 15	
6	**Motivation Methods:** What are the methods used to improve motivation?	1	9 21	

Q1) Explain Oldham-Hackman job characteristic model. **[9**

The Oldham-Hackman model proposes that managers should group task elements together to form "meaningful and satisfying assignments" rather than disjointed sets of activities. The model identifies five core job dimensions that influence a worker 's motivation, satisfaction, and performance,.

These five dimensions are categorized into factors that make work feel meaningful, factors that foster responsibility, and factors that provide knowledge of results.

1. Core Job Dimensions

A. Skill Variety

Definition: *This refers to the range of different abilities and skills a job holder needs to perform the work.*
Impact: *A job that requires a variety of skills is perceived as more challenging and meaningful than one that is monotonous or repetitive.*

B. Task Identity

Definition: This refers to the degree to which the job requires completing a "whole" and identifiable piece of work, from start to finish, rather than just a small, disconnected fragment.
Impact: Doing a job clearly from beginning to end makes the work feel more meaningful to the employee.

C. Task Significance

Definition: This is the extent to which the job has a substantial impact on the lives or work of other people, whether within the immediate organization or in the external environment.
Impact: Knowing that the work is important to others enhances its perceived value and meaningfulness.

D. Autonomy

Definition: This refers to the degree of freedom, independence, and discretion the worker has in scheduling the work and determining the procedures to be used.
Impact: High autonomy leads to a sense of "experienced responsibility" for the outcomes of the work. The worker feels personally accountable for success or failure.

E. Feedback

Definition: This is the degree to which carrying out the work activities results in the worker obtaining direct and clear information about the effectiveness of their performance.

Impact: This provides the worker with "knowledge of the actual results" of their activities, allowing them to adjust and improve,.

2. Psychological States and Outcomes

The model suggests that these five dimensions trigger specific psychological states:

- **Meaningfulness of Work:** Derived from **Skill Variety**, **Task Identity**, and **Task Significance**. The employee feels the work is worthwhile.
- **Responsibility for Outcomes:** Derived from **Autonomy**. The employee feels personally accountable.
- **Knowledge of Results:** Derived from **Feedback**. The employee knows how well they are performing.

3. Individual Differences (Growth Needs)

The effectiveness of this model depends on the individual's "requirement for personal development" (often called Growth Need Strength).

- People with a high need for personal growth will respond positively to jobs high in these five characteristics.
- Conversely, some employees may prefer more structure and less autonomy; for them, high responsibility might cause stress rather than motivation. As the sources note, "Some authors have argued that people are more inclined to rank their work higher on the Oldham-Hackman dimensions if they are happy with it for other reasons".

Q2) Apply the Oldham-Hackman Job Characteristics Model to a software development role and discuss how the job design could be improved to increase job satisfaction and motivation. **[9]**

Based on the Oldham-Hackman Job Characteristics Model, job satisfaction and motivation in a software development role are driven by five core job dimensions. To improve job design, managers should structure work to enhance these dimensions, thereby creating "meaningful and satisfying assignments" rather than disjointed activities,.

Here is the application of the model to a software development role and how job design can be improved:

1. Skill Variety

Definition: This refers to the degree to which a job requires a range of different activities and skills to perform the work.

- **Current State:** A developer might be restricted to writing boilerplate code or fixing minor bugs, utilizing only a narrow set of coding skills.
- **Improvement Strategy:** Implement **Job Enrichment**. Instead of restricting a developer to coding, the role should be designed to include requirements gathering, system design, and testing. This forces the use of diverse skills (e.g., analytical, design, and communication skills), making the job feel more challenging and meaningful,.

2. Task Identity

Definition: This is the degree to which a job requires completing a "whole" and identifiable piece of work from beginning to end.

- **Current State:** A developer works on small, disconnected fragments of a larger system (e.g., a single subroutine) without seeing the final product.
- **Improvement Strategy:** Assign ownership of complete modules or features. Allow the developer to see a feature through from the initial concept to the final deployment. Doing a job from start to finish makes the work more meaningful than doing just a small part,.

3. Task Significance

Definition: This refers to the extent to which the job has a substantial impact on the lives or work of other people, whether within the organization or the external world.

- **Current State:** The developer feels they are just "moving data around" without understanding who benefits.
- **Improvement Strategy:** Communicate the "big picture." Managers should demonstrate how the software improves user efficiency, safety, or business revenue. Knowing the work is important to others enhances its perceived value,.

4. Autonomy

Definition: This is the degree of freedom, independence, and discretion the worker has in scheduling the work and determining the procedures to be used.

- **Current State:** A manager dictates exactly how to write the code, which tools to use, and micromanages the daily schedule.
- **Improvement Strategy:** Grant "experienced responsibility." Define the *what* (requirements) but let the developer decide the *how* (algorithms, design patterns, specific tools). Increasing autonomy makes the developer feel personally accountable for the success or failure of their work,.

5. Feedback

Definition: This is the degree to which carrying out the work activities results in the worker obtaining direct and clear information about the effectiveness of their performance.

- **Current State:** The developer submits code and hears nothing back until a formal performance review months later.
- **Improvement Strategy:** Implement systems that provide "knowledge of actual results." This can be achieved through automated testing results, immediate peer code reviews, or direct user feedback mechanisms. The job itself should provide feedback so the developer knows immediately if their work is effective,.

Summary of Outcomes

By improving these five dimensions, the software development role triggers three critical psychological states:

1. **Meaningfulness of work** (derived from Skill Variety, Task Identity, and Task Significance).
2. **Responsibility for outcomes** (derived from Autonomy).
3. **Knowledge of results** (derived from Feedback).

These states collectively lead to high internal work motivation, high-quality work performance, and high satisfaction with the work.

---

q3) What is Leadership? Explain Different approaches of Leadership. [9]

What is Leadership?

Leadership is typically understood as the capacity to persuade others to follow a course of action. It goes beyond simple management or administration; it involves the ability to influence others to perform tasks they might not otherwise do.

While leaders may not always hold official authority, leadership is founded on the concept of **power** or **influence**. The sources identify several forms of power that a leader might utilize:

- **Coercive Power:** The power to force someone to act by threatening punishment.
- **Connection Power:** Derived from having access to those in positions of power.
- **Legitimate Power:** Founded on the idea that a person's title bestows a unique status.
- **Reward Power:** Where the holder can reward those who perform their duties to their pleasure.
- **Expert Power:** Results from being capable of performing a specific task.
- **Power of Information:** Grants the holder exclusive access to information.
- **Referent Power:** Based on the leader's personal allure or charisma.

Different Approaches (Styles) of Leadership

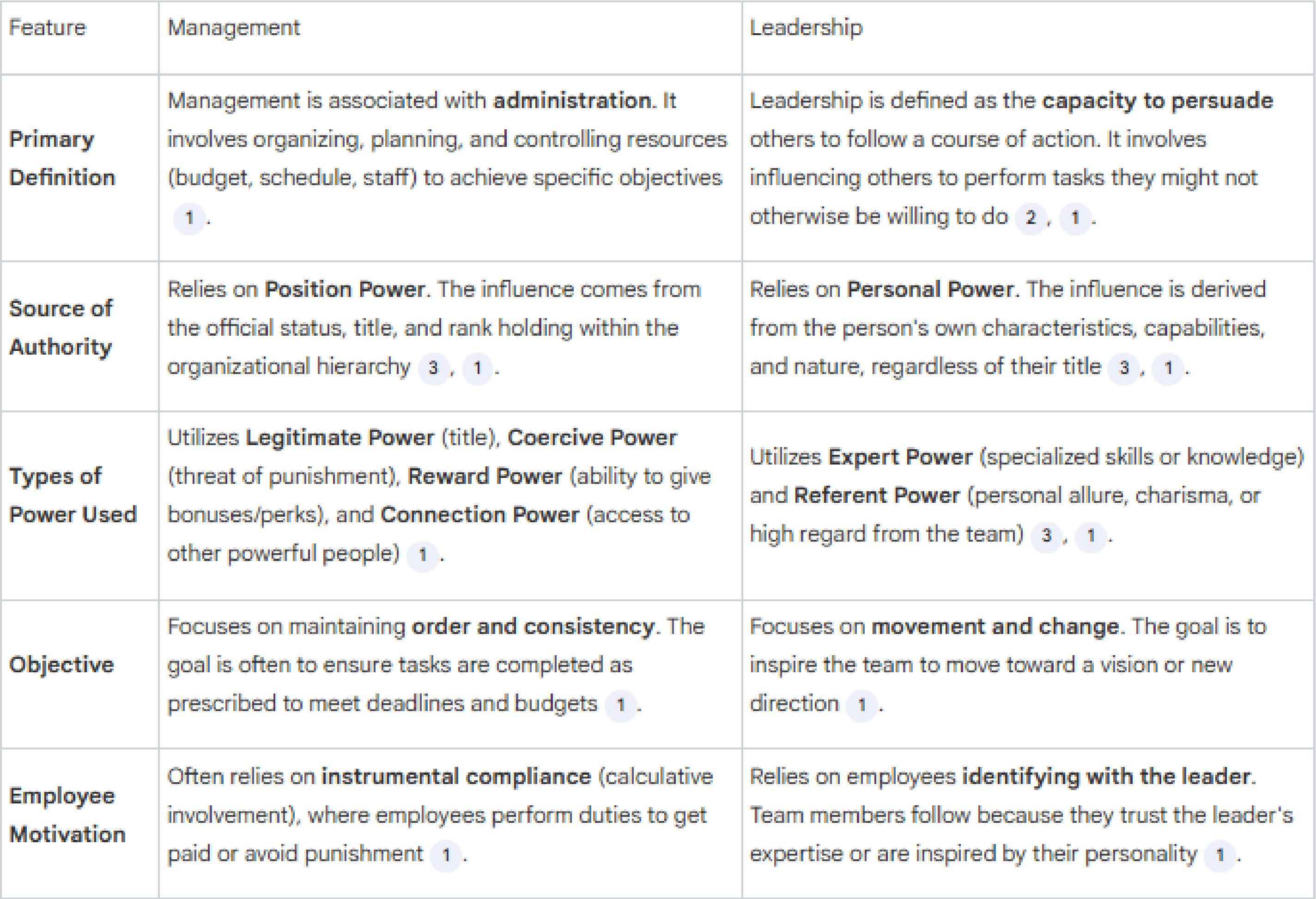
The sources describe leadership styles by evaluating them along two primary axes: **Autocratic vs. Democratic** (decision-making) and **Directive vs. Permissive** (execution supervision).

Based on these axes, four specific leadership approaches are identified:

1. **The Directive Autocrat**
    a. **Decision Making:** This leader makes decisions alone.
    b. **Execution:** They closely monitor the execution of tasks.
    c. **Description:** This approach is highly controlling, where the leader dictates what needs to be done and watches over the staff to ensure it is done exactly as prescribed.
2. **The Permissive Autocrat**
    a. **Decision Making:** This leader makes judgments and decisions alone.
    b. **Execution:** They give discretion to subordinates to carry them out.
    c. **Description:** While the leader sets the goals and makes the choices, they trust the team enough to allow them the freedom to determine *how* to achieve those goals.
3. **The Directive Democrat**
    a. **Decision Making:** This leader participates in decision-making collaboratively with the team.
    b. **Execution:** They closely monitor the execution of tasks.
    c. **Description:** The leader involves the team in deciding what to do, but once the decision is made, they maintain tight control over the implementation process.
4. **The Permissive Democrat**
    a. **Decision Making:** This leader takes decisions collaboratively.
    b. **Execution:** They give discretion to subordinates in carrying them out.
    c. **Description:** This is the most decentralized approach. Decisions are made as a group, and the team is trusted to execute the work with minimal supervision.

The effectiveness of these styles often depends on the team's professional capability and the specific context of the project . For example, highly technical work might benefit from a leader with **Expert Power**, while critical deadlines might require a more **Directive** approach,.

Q4) Differentiate between management and leadership.



Feature	Management	Leadership
Primary Definition	Management is associated with **administration.** It involves organizing, planning, and controlling resources (budget, schedule, staff) to achieve specific objectives [1].	Leadership is defined as the **capacity to persuade** others to follow a course of action. It involves influencing others to perform tasks they might not otherwise be willing to do [2], [1].
Source of Authority	Relies on **Position Power**. The influence comes from the official status, title, and rank holding within the organizational hierarchy [3], [1].	Relies on **Personal Power**. The influence is derived from the person's own characteristics, capabilities, and nature, regardless of their title [3], [1].
Types of Power Used	Utilizes **Legitimate Power** (title), **Coercive Power** (threat of punishment), **Reward Power** (ability to give bonuses/perks), and **Connection Power** (access to other powerful people) [1].	Utilizes **Expert Power** (specialized skills or knowledge) and **Referent Power** (personal allure, charisma, or high regard from the team) [3], [1].
Objective	Focuses on maintaining **order and consistency**. The goal is often to ensure tasks are completed as prescribed to meet deadlines and budgets [1].	Focuses on **movement and change**. The goal is to inspire the team to move toward a vision or new direction [1].
Employee Motivation	Often relies on **instrumental compliance** (calculative involvement), where employees perform duties to get paid or avoid punishment [1].	Relies on employees **identifying with the leader**. Team members follow because they trust the leader's expertise or are inspired by their personality [1].

Here is a detailed differentiation with relevant examples based on the types of power described in the text:

1. Source of Authority: Position Power vs. Personal Power
The most distinct difference lies in where the individual derives their influence.

- **Management (Position Power):** Managers rely on authority granted by their official rank or title within the organization. Their ability to influence comes from the resources they control,.
  - **Example (Legitimate Power):** A Project Manager tells a software developer to prioritize "Module A" over "Module B." The developer complies simply because the manager holds the official title and has the right to assign work
  - **Example (Reward Power):** A manager encourages a team to work overtime by promising a bonus or a paid day off. The team complies to receive the tangible reward.
  - **Example (Coercive Power):** A manager warns an employee that continued tardiness will result in a formal reprimand or termination. The employee changes behavior out of fear of punishment.

- **Leadership (Personal Power):** Leaders derive influence from their own character, skills, and relationships, regardless of their official title. They can be influential even without being in charge,.
  - **Example (Expert Power):** A senior programmer with no management title suggests a complex architectural change. The entire team adopts the change not because they *have* to, but because they respect the programmer's deep technical knowledge and believe it is the best solution.
  - **Example (Referent Power):** A team lead is charismatic, fair, and supportive. When a crisis hits, the team volunteers to stay late to fix it, not because of a bonus, but because they value their relationship with the lead and want to support them.

2. Nature of Influence: Compliance vs. Persuasion

- **Management (Administration):** Management is concerned with organizing, planning, and staffing. It focuses on maintaining order and consistency. The interaction is often transactional.
  - **Example:** A manager creates a Gantt chart, assigns resources to tasks, and ensures timesheets are filled out correctly to keep the project within budget. This is an administrative function ensuring the "machinery" of the project runs smoothly.
- **Leadership (Persuasion):** Leadership is concerned with movement and change. It involves persuading people to align with a vision, often requiring them to go beyond their basic job description.
  - **Example:** A project creates a stressful environment due to a client change. A leader (who could be the manager or a team

member) rallies the team, reframes the challenge as an opportunity for innovation, and keeps morale high. This is an emotiona l and persuasive function.

3. Decision-Making Styles

While managers often have the authority to make decisions unilaterally, the text highlights that leadership styles vary in ho w they involve the team:

• **Management Scenario (Directive Autocrat):** A manager decides solely on the implementation strategy for a new feature and closely monitors the team to ensure they follow the instructions exactly. This prioritizes control and execution.

• **Leadership Scenario (Permissive Democrat):** A leader gathers the team, explains the problem, and allows the group to decide the solution by consensus. The leader then trusts the team to execute the solution with minimal interference, fostering a sense of ownership and autonomy.

---

q5) How to select a right person for the job? Explain the recruitment process in detail. **[9]**

selecting the right person aligns with **Frederick Taylor's** management philosophy. Taylor emphasized the necessity of the "first-class man," suggesting that productivity is maximized when the person's skills and the job's requirements are perfectly matched. If a team member is not recruited correctly, it can lead to meaningful disparities in productivity compared to their colleagues.

1. Create a Job Specification

The first step is to clearly define the role.

Definition: This involves writing down and agreeing upon the needs of the job, including the specific kinds of tasks to be performed.

Legal Consideration: Advice is frequently required when drafting this because a formal document could have legal ramifications regarding discrimination or labor laws.

Outcome: A formal document describing the duties and responsibilities of the role,.

2. Create a Profile for the Job Holder

Once the job is defined, the manager must define the type of person required to fill it.

- **Profiling:** The individual required to perform the job is profiled using the job specification.
- **Details:** This profile lists the necessary characteristics, credentials, training, and experience required.
- **Goal:** To create a clear benchmark against which applicants can be measured,.

3. Recruit Applicants

This stage involves attracting a pool of potential candidates.

Advertising: An advertisement is published, either internally within the company or outside in the local or trade press.

Strategy: The goal is to determine the media most likely to reach the largest number of possible applicants at the lowest cost. The job holder profile would be thoroughly evaluated to determine the best placement for the ad,.

4. Examine Resumes/CVs (Screening)

The objective here is to filter the applicants to a manageable number.

Self-Elimination: A key rule in advertising is to include enough details in the advertisement to permit "self-elimination." Candidates who know they do not fit the criteria should be discouraged from applying.

Filtering: The applicant pool should be reduced to those who have realistic qualifications by providing income location, job description, and prerequisites in the ad. This saves time during the interview phase.

5. Interviews and Testing

This is the core assessment phase to evaluate the candidate's fit.

Methods: Common selection approaches include aptitude exams, personality tests, and the scrutiny of work samples. These are used to test certain traits listed in the job holder profile.

The Interview: The method most frequently utilized is the interview. Its primary purpose is to maximize the chance of discussion and follow-up questions.

  - **Structure:** Ideally, there should be **two interviewers** for each session. This allows for a more balanced evaluation.

    - **Scoring:** A formal scoring system should be developed where interviewers decide on individual scores that are then compared to ensure fairness.
    - **Content:** An interview could be of a general or technical nature. When the candidate's practical expertise is evaluated, a significant portion of the interview may include assessing and correlating information from the CV.

6. Other Techniques

Final checks are performed before an offer is made.
References: References provided by the candidate will be contacted to verify history and character.
Medical Checkup: A medical checkup can be required to ensure the candidate is fit for the role.

---

q6) Explain five fundamental stages of development. [9]
The five fundamental stages of development in **software project management** refer to **Tuckman's Stages of Group Development**, which describe the predictable social evolution of a project team. Understanding these stages allows the Project Manager to anticipate team dynamics and apply appropriate leadership to maximize productivity.

These stages are: **Forming, Storming, Norming, Performing, and Adjourning.**

1. Forming 🤝

This is the stage where the team is initially **assembled** and begins to orient itself to the project and each other.

- **Characteristics:** Team members are typically polite, cautious, and non-committal. They focus on understanding the project **goals, scope, and rules**.
- **Challenge for PM:** High uncertainty and dependence on the Project Manager (PM) for clear direction and initial task assignments. The team lacks established trust.
- **Example:** In a software project kickoff meeting, the PM introduces the developers, defines the architecture goals, and assigns the initial setup tasks like environment configuration and repository access.

2. Storming 💥

Conflict and competition emerge as team members begin to assert their individual opinions and establish their place within the team hierarchy.

- **Characteristics: Disagreements** over project methods (e.g., choice of framework, coding standards), roles, and even leadership. Tension and frustration are common. This is the **most critical** stage.
- **Challenge for PM:** Mediating conflicts constructively, maintaining focus on the goals, and ensuring disagreements are about ideas, not personalities. Teams can get stuck here if conflict is avoided or handled poorly.
- **Example:** Two senior developers strongly disagree on whether to use a microservices approach or a monolithic architecture, leading to heated discussions in sprint planning until the PM steps in to facilitate a data-driven decision.

3. Norming ✅

The team resolves its conflicts, establishes consensus, and develops an agreed-upon way of working.

- **Characteristics: Cohesion** increases, roles and responsibilities become clear, and **group norms** (e.g., meeting schedule, code review process, communication channels) are accepted. Team members begin to trust each other.
- **Challenge for PM:** Reinforcing positive behavior, stepping back from day-to-day decisions, and encouraging the team to take more responsibility for problem-solving.
- **Example:** The software team agrees to use a specific pull request template for code reviews and establishes a shared document detailing all decision rules and conventions, leading to a much smoother workflow.

4. Performing 🚀

The team is fully functional, operating with autonomy, high efficiency, and a shared commitment to achieving the project goal .

- **Characteristics:** High **productivity** and seamless collaboration. The team can resolve internal issues independently and focus all energy on delivering high-quality software. Roles are flexible and interchangeable as needed.

- **Challenge for PM:** The PM's role shifts to a **"servant leader"** or **"gateway,"** primarily removing external obstacles, communicating with stakeholders, and celebrating successes.
- **Example:** The team consistently meets or exceeds its sprint goals, automatically detects and fixes integration errors, and proactively suggests process improvements without needing PM intervention.

5. Adjourning 🎉

This is the final stage, when the project is completed, and the team members transition to new assignments.

- **Characteristics:** The emphasis shifts to **final wrap-up activities**, such as knowledge transfer, documentation, and the final project retrospective (lessons learned).
- **Challenge for PM:** Recognizing and rewarding the team's efforts, celebrating the project's success, and addressing the emotional impact of separation (sometimes called *mourning*), especially for high-performing teams.
- **Example:** The PM hosts a final launch party, conducts a thorough post-mortem to capture what went well and what didn't, and ensures all code is properly archived before disbanding the development team.

---

q7) Explain briefly the FOUR (4) types of teams in an organization. [9]

1. Hierarchical Team Organization

In this traditional structure, control and decision-making flow from the top down.

- **Structure:** It consists of levels of authority where top-level managers make decisions for the entire organization. Middle-level managers act as a liaison between the top and the bottom, translating strategies into tasks.
- **Function:** Lower-level managers and employees execute the tasks. The primary focus is on control, motivation, and guiding lower-level managers to achieve targets and goals.

2. Chief Programmer Team Organization

This structure is highly centralized and specialized, often compared to a surgical team.

- **Structure:** The "Chief Programmer" is the central figure (like a surgeon) who is directly involved in the system development and can exercise better control function. They are supported by a "Backup Programmer" and other support staff (like a librarian or tool smith).
- **Function:** The chief programmer defines the specification, design, and code. This approach allows for highly skilled execution but poses a risk if the chief programmer leaves or is overwhelmed, as the project's survival depends heavily on one person.

3. Matrix Team Organization

This structure organizes staff based on their specialization rather than just the project they are working on.

- **Structure:** Staff members are divided into specialist groups (e.g., a database group, a testing group). Each group has a manager.
- **Function:** When a project needs specific skills, members from these specialist groups are assigned to the project. This allows for the efficient handling of problems by experts in that specific field, though it requires careful management of resources across different projects.

4. Egoless Team Organization

Based on the philosophy of "egoless programming" (popularized by Gerald Weinberg), this team structure encourages a collaborative environment where work is viewed as a collective asset.

- **Structure:** There is no single "king" or authoritarian leader. The team works together towards a common goal.
- **Function:** Code and designs are considered "common property" rather than personal creations. The team emphasizes peer reviews and finding faults in the work (the program) rather than the person, fostering an environment where group members can accept criticism and work together to improve the product.

Q8) Discuss the different theories of organizational behaviour and their implications for software project management. Explain how the different factors that influence individual and team behaviour in software projects, such as motivation, personality, and group dynamics, can impact project performance. **[9]**

1. Theories of Organizational Behaviour and Motivation

Organizational Behaviour (OB) involves the study of individual and group behavior in software development environments. The sources highlight several key theories that guide how project managers should treat and motivate their staff.

- **The Taylorist Model (Theory X):**
  - **Theory:** Based on Frederick Taylor's scientific management, this model views human behavior as mechanistic. It assumes the "average human has an innate dislike of work" and requires coercion, direction, and control. It relies on financial incentives ("piece-rates") as the primary motivator,,.
  - **Implications for SPM:** In software projects, this approach manifests as strict control, micro-management, and regarding developers as interchangeable resources. While it aims for efficiency, it often ignores the creative nature of software development and can lead to frustration among skilled professionals.
- **McGregor's Theory Y:**
  - **Theory:** Donald McGregor proposed that work is as natural as play or rest. People can exercise self-direction and self-control if they are committed to objectives. They not only accept but seek responsibility,.
  - **Implications for SPM:** This supports a participative management style. Software managers should provide autonomy and trust, allowing developers to make technical decisions, which fosters higher commitment and innovation.
- **Maslow's Hierarchy of Needs:**
  - **Theory:** Humans have a hierarchy of needs ranging from basic physiological needs to safety, social acceptance, esteem, and finally self-actualization,.
  - **Implications for SPM:** A project manager must understand that once a developer's basic needs (salary/security) are met, more money may not be an effective motivator. To improve performance, the manager must offer opportunities for esteem (recognition) and self-actualization (challenging technical work).
- **Herzberg's Two-Factor Theory:**
  - **Theory:** This theory distinguishes between **Hygiene Factors** (e.g., salary, working conditions), which prevent dissatisfaction but do not motivate, and **Motivators** (e.g., achievement, recognition, the work itself), which actually drive performance,.
  - **Implications for SPM:** A software manager must ensure good "hygiene" (fair pay, good tools) to avoid turnover, but to actually increase project performance, they must design jobs that provide a sense of achievement and personal growth.
- **Vroom's Expectancy Theory:**
  - **Theory:** Motivation is determined by three factors: **Expectancy** (belief that effort leads to performance), **Instrumentality** (belief that performance leads to a reward), and **Valence** (the value of the reward to the individual),.
  - **Implications for SPM:** A manager must ensure developers believe the project goals are achievable (Expectancy) and that they will actually receive a valued reward (bonus, promotion, or time off) upon success.
- **Oldham-Hackman Job Characteristics Model:**
  - **Theory:** This model suggests that job satisfaction comes from five core dimensions: Skill Variety, Task Identity, Task Significance, Autonomy, and Feedback,.
  - **Implications for SPM:** Software tasks should be designed to be "meaningful." Instead of assigning a developer a small, disconnected fragment of code, they should own a feature from start to finish (Task Identity) and understand its impact on the user (Task Significance) to maximize internal motivation.

2. Factors Influencing Behaviour and Project Performance

The sources identify specific factors regarding personality and group dynamics that directly impact the success or failure of a software project.

A. Personality and Team Roles (Belbin)

Individual personality traits determine how team members interact. ***Meredith Belbin*** *identified that successful teams require a balance of specific roles, not just technical skills,,.*

- **Key Roles:** A team needs a **Plant** (creative problem solver), a **Resource Investigator** (explorer of opportunities), a **Coordinator** (mature chairperson), a **Shaper** (dynamic drive), a **Monitor Evaluator** (strategic judge), and a **Completer-Finisher** (anxious to get the job done correctly),.
- **Impact on Performance:** If a software team consists only of "Plants" (creative types), they may generate great ideas but never finish the code. If comprised only of "Implementers," they may work hard but lack strategic direction. Managers must select a balanced mix of personalities to ensure all aspects of the project (innovation, execution, completion) are covered.

B. Group Dynamics and Development Stages (Tuckman)

Groups are not static; they evolve. The sources describe ***Tuckman's model*** *of team development,:*

1. **Forming:** Members get to know each other; reliance on the leader.
2. **Storming:** Conflict arises as members resist constraints and vie for leadership.
3. **Norming:** Conflicts are resolved, and a sense of identity/cooperation emerges.
4. **Performing:** The team becomes fully functional and focuses on the task.
5. **Adjourning:** The group disbands after project completion.

- **Impact on Performance:** A project manager must recognize that conflict (Storming) is a natural phase. If managed poorly, the team may never reach the "Performing" stage, leading to project failure. The manager's leadership style may need to shift from directive (in Forming) to permissive (in Performing).

C. Group Decision Making

Group dynamics significantly influence decision quality.

- **Obstacles:** The sources warn of **"Risky Shift"** (groups taking riskier decisions than individuals) and **Groupthink** (pressure to conform suppressing dissent).
- **Impact on Performance:** To maintain performance, managers must structure decision-making to encourage diverse views and prevent the suppression of critical feedback, ensuring that technical risks are properly evaluated.

D. Recruitment and Selection

Selecting the "right person for the job" is the foundation of individual behavior. Following Taylor's view, productivity is maximized when a person's skills match the job profile. Poor recruitment leads to low productivity and team friction. The process involves defining a job specification, creating a job holder profile, and using interviews/testing to ensure a psychological and technical fit,.

---

q9) List the factors that are involved in making a team. Explain the characteristics.

1. Selection (The "Right Person for the Job")

The foundation of any team is recruitment. Following Frederick Taylor's management philosophy, productivity is maximized when the person's skills match the job requirements perfectly.

Characteristics:

- **Job Specification:** The process begins by clearly defining the tasks, duties, and responsibilities of the role,.
- **Job Holder Profile:** Managers create a profile of the ideal candidate, including education, experience, and psychological traits, to serve as a benchmark for applicants,.
- **Recruitment & Screening:** Methods such as aptitude tests, personality tests, and interviews are used to ensure both technical competence and cultural fit.

2. Balanced Team Roles (Belbin's Management Teams)

Technical skills alone do not guarantee a successful team. **Meredith Belbin** identified that teams must have a balance of psychological roles to function effectively. A team full of creative geniuses (Plants) might fail because no one attends to details.

- **Characteristics (Key Roles):**
    - **The Chair/Coordinator:** Mature and confident; clarifies goals and promotes decision-making,.
    - **The Plant:** Creative and imaginative; solves difficult problems but may ignore practical details,.
    - **The Monitor Evaluator:** Sober and strategic; judges options accurately,.
    - **The Shaper:** Dynamic and challenging; finds ways around obstacles,.
    - **The Completer-Finisher:** Anxious and conscientious; searches for errors and ensures the job is finished on time,.
    - **The Team Worker:** Cooperative and diplomatic; averts friction within the group,.

3. Team Structure and Organization

How the team is structured determines communication flow and decision-making authority. The sources identify several distinct structures:

- **Characteristics:**
    - **Hierarchical/Departmental:** Standardization is high, but responsiveness can be slow. Communication flows through management layers.
    - **Chief Programmer Team:** Highly centralized. The "Chief" makes all critical decisions, supported by a backup and administrative staff. This is efficient for specific tasks but relies heavily on one individual,.
    - **Egoless (Democratic) Team:** Decentralized. Code is considered "common property." The group makes decisions by consensus. This fosters high quality and innovation but can be slower due to endless meetings,.
    - **Matrix Structure:** Staff are organized by specialty (e.g., programming, testing) and assigned to projects as needed. This allows for efficient use of specialized skills but can cause conflict between project managers and functional managers.

4. Team Development Stages (Tuckman Model)

Teams are not instant; they evolve through specific stages of development. Understanding this lifecycle is a factor in managing expectations and behavior.

- **Characteristics:**
    - **Forming:** The team meets; members are anxious and rely on the leader for guidance,.
    - **Storming:** Conflict arises as members vie for positions and resist constraints. This is the most difficult stage,.
    - **Norming:** Conflicts are resolved. The team establishes "norms" (rules of behavior) and cohesion develops,.
    - **Performing:** The team is fully functional and focused on the task at hand,.
    - **Adjourning:** The project ends, and the group disbands,.

5. Decision-Making and Group Dynamics

The way a team makes decisions affects its performance. Managers must navigate obstacles that arise from group dynamics.

- **Characteristics:**
    - **Groupthink:** A phenomenon where the desire for harmony overrides the realistic appraisal of alternatives, leading to bad decisions.
    - **Risky Shift:** The tendency for groups to take riskier decisions than any single individual would take alone.
    - **Methods to Improve:** Techniques like the **Delphi Method** (collecting anonymous expert opinions) can help avoid these pitfalls by reducing peer pressure.

6. Communication and Coordination

Teams require effective coordination mechanisms to handle dependencies between tasks.

- **Characteristics:**

- **Shared Resources:** Managing conflicts when different tasks require the same equipment or staff.
- **Producer-Consumer Relationships:** Ensuring that the output of one task (producer) meets the needs of the next task (consumer).
- **Communication Genres:** Utilizing the right mode of communication (e.g., formal meetings vs. instant messaging) based on the constraints of time and place.

----

q10)What are the methods used to improve motivation? [9]

Methods Used to Improve Motivation

To enhance motivation within a software project team, managers can employ specific techniques related to goal setting and job design:

• **Set Specific Goals:** These goals need to be demanding and yet acceptable to the staff. Involving staff in the setting of goals helps gain acceptance for them. Goals provide a clear target for the team to aim for,.

• **Provide Feedback:** Feedback is necessary not only to correct errors but also to provide positive reinforcement. It helps team members understand their progress and adjust their efforts accordingly. Feedback should be regular,.

• **Consider Job Design:** Jobs can be altered to make them more interesting and give staff more feeling of responsibility. Two primary methods are used:

◦ **Job Enlargement:** This involves widening the range of tasks (horizontal expansion) or the trend of specialization. It attempts to reverse the issue where a person performs a very small, repetitive part of the process,.

◦ **Job Enrichment:** This involves giving the job holder more authority and control over their work (vertical expansion). It allows the employee to manage their own tasks, making the work more meaningful and satisfying,.