OVERVIEW

# Team

The most examined chapter in SE423: both 251 midterms took most of their short questions from it, and both finals have a whole team-based collaboration question. It has three parts. Team development: Tuckman's stages, team objectives and structures, and the team needs pyramid (trust → managed conflict → commitment → accountability → results) with RACI and skills matrices as the accountability tools. Team models: business, democratic, chief-programmer, skunk works, SWAT, and why large teams need hierarchy (n(n−1)/2 channels). Managing people: motivation and personality types, cohesiveness and groupthink, staff selection, working environments, shaping project culture, technical vs. adaptive leadership, and reaching consensus.

The instructor's style: questions ask you to apply an idea to a team situation ("team members disagree: how do you encourage collaboration?"). Saying one word ("negotiation") is not enough; explain what you do, how, and why.

**Team pyramid**

Trust → managed conflict → commitment → accountability → results. Each layer needs the one below; trust is the foundation.

**Tuckman**

Forming → storming (competing for leadership) → norming (agree how to work) → performing → adjourning.

**RACI**

R does the work, A owns the outcome (exactly one per task), C gives input (two-way), I is kept updated (one-way).

**Team models**

Business (tech lead + equals, most common), democratic (whole team decides), chief-programmer (surgical), skunk works, SWAT.

**Leadership**

Technical leadership for problems solvable by known means; adaptive leadership for new or complex ones. Most IT projects need both.

**Consensus**

Not unanimous agreement: everyone feels heard and the group moves forward committed. Persuade, don't dictate.

## HOW THE LECTURE IS ORGANIZED

**Slides 1–9** Team development: influence, the pyramid, Tuckman, team objectives

**Slides 10–24** The pyramid layer by layer: trust, conflict, commitment, accountability (RACI, RAM, skills matrix), results

**Slides 25–34** Team models and team size

**Slides 35–56** Managing people: motivation, personality types, cohesiveness, communication, selection, environment

**Slides 57–70** Project culture, managing thought workers, making people better

**Slides 71–77** Leadership, consensus and key points

## EXAM FOCUS

Midterm 251 (both versions): MCQs on Tuckman's storming stage, the foundation of teamwork and the advantage of a cohesive team; short questions on good-team characteristics, vulnerability-based trust, healthy conflict, the best time to reinforce accountability, business vs. democratic teams, technical vs. adaptive leadership and a RACI table; a case on accountability after an untested release. Finals: commitment, trust and conflict, accountability and transparency (Final 242 Q5), the communication-channel formula (Final 242), a CI/CD consensus decision and a milestone celebration (Final 251 Q5). The instructor's 2026 midterm asked how a PM encourages collaboration when members disagree.

### HOW TO READ THIS PAGE

Every original slide is shown unchanged. Yellow marks on a slide point to a numbered note: hover or tap a mark to jump to its note, and hover a note to see its marks. Questions tagged **Past exam** appeared on a real paper; **Prediction** questions are likely angles, not confirmed. Pick an answer to each question: a wrong pick tells you why it is wrong, so you can try again.

SLIDE 1 / 77 **Slide 1**



# Team

SE423: Software Project Management

1

NOTES

1 The deck has three parts: **team development** and the team needs pyramid (slides 3–24), **team models** (25–34) and **managing people** (35–77). The pyramid (trust → conflict → commitment → accountability → results) is the spine of the chapter and of most exam questions on it.

---

SLIDE 2 / 77 **Outline**

## Outline

- Team Development
- Building Trust
- Managing Conflict
- Embracing Accountability
- Team Models
- Managing People

2

NOTES

1. **Team development** first: how a group becomes a team.
2. Then three of the pyramid layers in order: **building trust**, **managing conflict**, **embracing accountability** (commitment and results are covered between them).
3. Then the **team models** (how teams are organized) and **managing people** (motivation, selection, culture, leadership).

---

SLIDE 3 / 77 **Slide 3**




NOTES

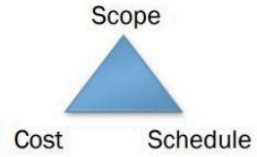
1. Section divider: **Team Development**.

---

 **Team Development - Introduction**

# Team Development - Introduction

- What are your responsibilities as a team member?
- As a Project Manager?
- How does Team Development relate to the Project Triangle?




- What's more important? The strength of individual team members, or the strength of the team?
- What are some characteristics of a good team?

4

## NOTES

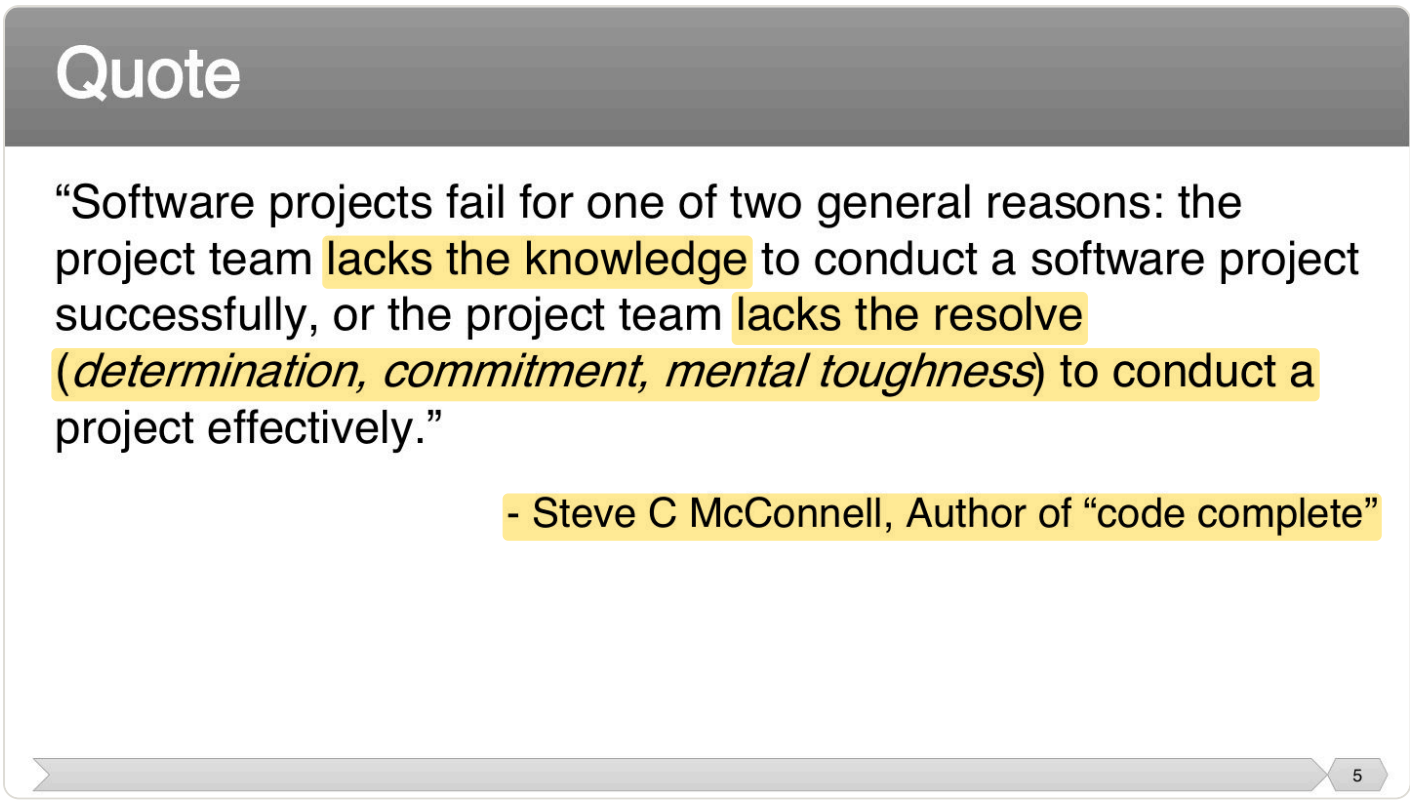
1. Discussion questions to open the topic. As a **team member** you own your tasks and support others; as a **PM** you build the conditions for the team to work: clear goals, roles, trust.
2. The link to the **project triangle**: a weak team costs you on scope, schedule or cost. People are the resource that turns the plan into results.
3. Individuals vs. team: a team of strong individuals who don't trust each other usually loses to a well-functioning team of average people. Team strength wins.
4. "Characteristics of a good team" was a real midterm short question. Answer it with the pyramid, from the bottom up.

## EXAM ANGLE

Past exam · Midterm 251 · V1 · Short Q1 — Q1

What are key characteristics of a good team?
**Answer:** Members trust each other (the foundation), so they speak openly and admit mistakes; they have healthy conflict about ideas rather than people; they commit to decisions with clarity and buy-in; they hold each other accountable; and they put team results ahead of individual credit. A good team also shares a clear goal with clear roles, communicates openly, is cohesive, balances personality types and is the right size (about 4–6 developers plus a lead).

 **Quote**




NOTES

1. McConnell’s two reasons projects fail. **Knowledge**: the team doesn’t know how to run a software project. Training, hiring and process fix this.
2. **Resolve** (determination, commitment, mental toughness): the team knows how but doesn’t push through. This is what team building, motivation and leadership fix, and it is what the rest of the chapter is about.

EXAM ANGLE

Prediction Q1

A skilled team keeps giving up when work gets hard. Which of McConnell's two reasons fits?

**A** A lack of budget

> McConnell's two reasons are knowledge and resolve.

**B** A lack of knowledge

> The team is described as skilled.

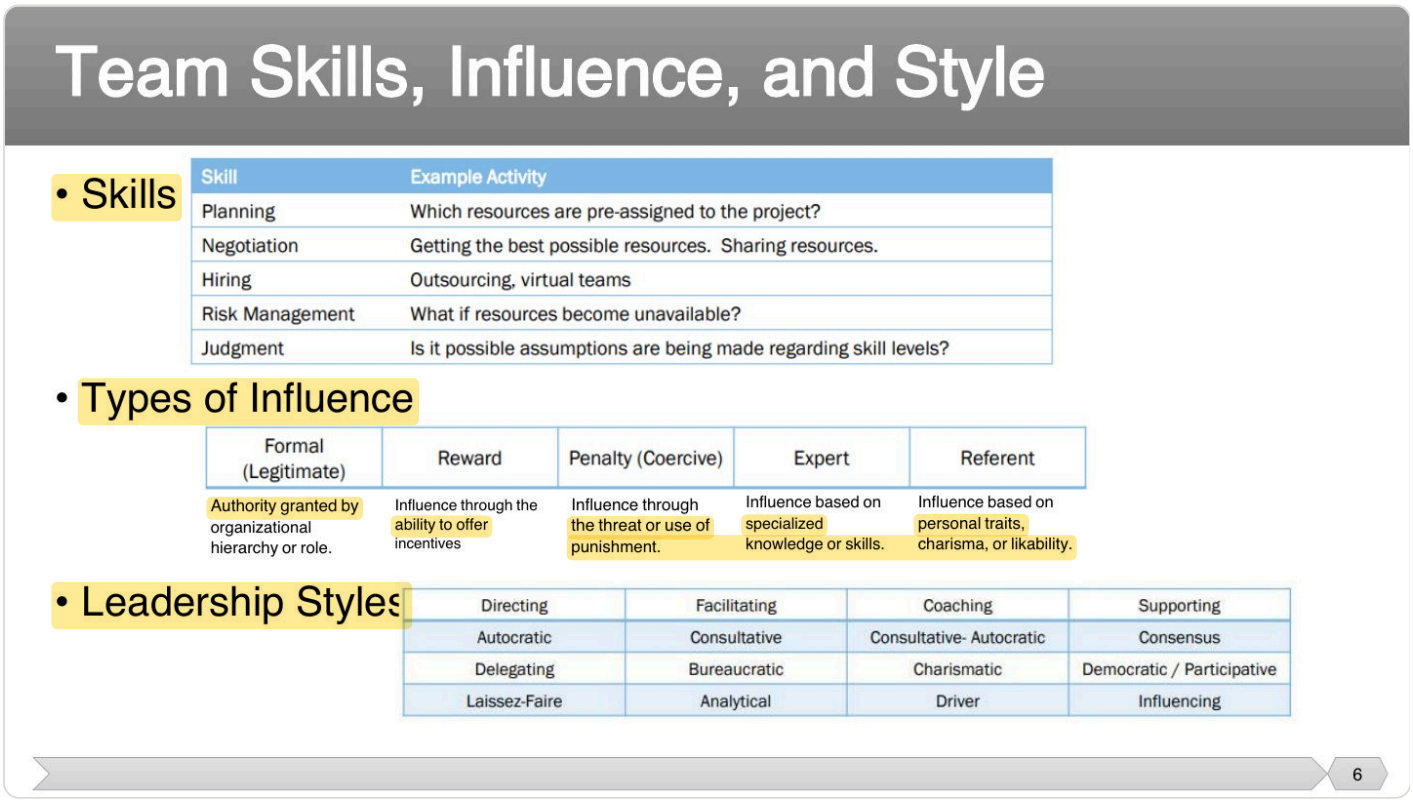
**C** A lack of tools

> That isn't one of the two.

**D** **A lack of resolve**

> They know how, but don't push through.

SLIDE 6 / 77 **Team Skills, Influence, and Style**




NOTES

1 The **types of influence** a PM can use. **Formal (legitimate)**: authority granted by the organizational hierarchy or role. The skills table above lists the PM's team skills (planning, negotiation, hiring, risk management, judgement) and the bottom table lists leadership styles (directing, facilitating, coaching, supporting, etc.).

2. It works because people want the reward, and it stops working if you run out of things to offer. **Reward** also requires that you actually control something they value, such as bonuses or interesting assignments.
3. **Penalty (coercive)**: influence through the threat or use of punishment. It gets compliance, not commitment.
4. People follow you because you know the answer, even without a title. That's why a senior developer can have more real **expert** influence over a team than their manager does.
5. **Referent**: influence based on personal traits, charisma or likability. Formal, reward and penalty come from your **position**; expert and referent come from **you**, and they last longer because people choose to follow.

---

EXAM ANGLE

Prediction Q1

A senior developer persuades the team because everyone respects her deep database knowledge. This is:

A Reward influence

B Formal influence

C **Expert influence**

D Penalty influence

Prediction Q2

Which types of influence come from the person rather than from the position?

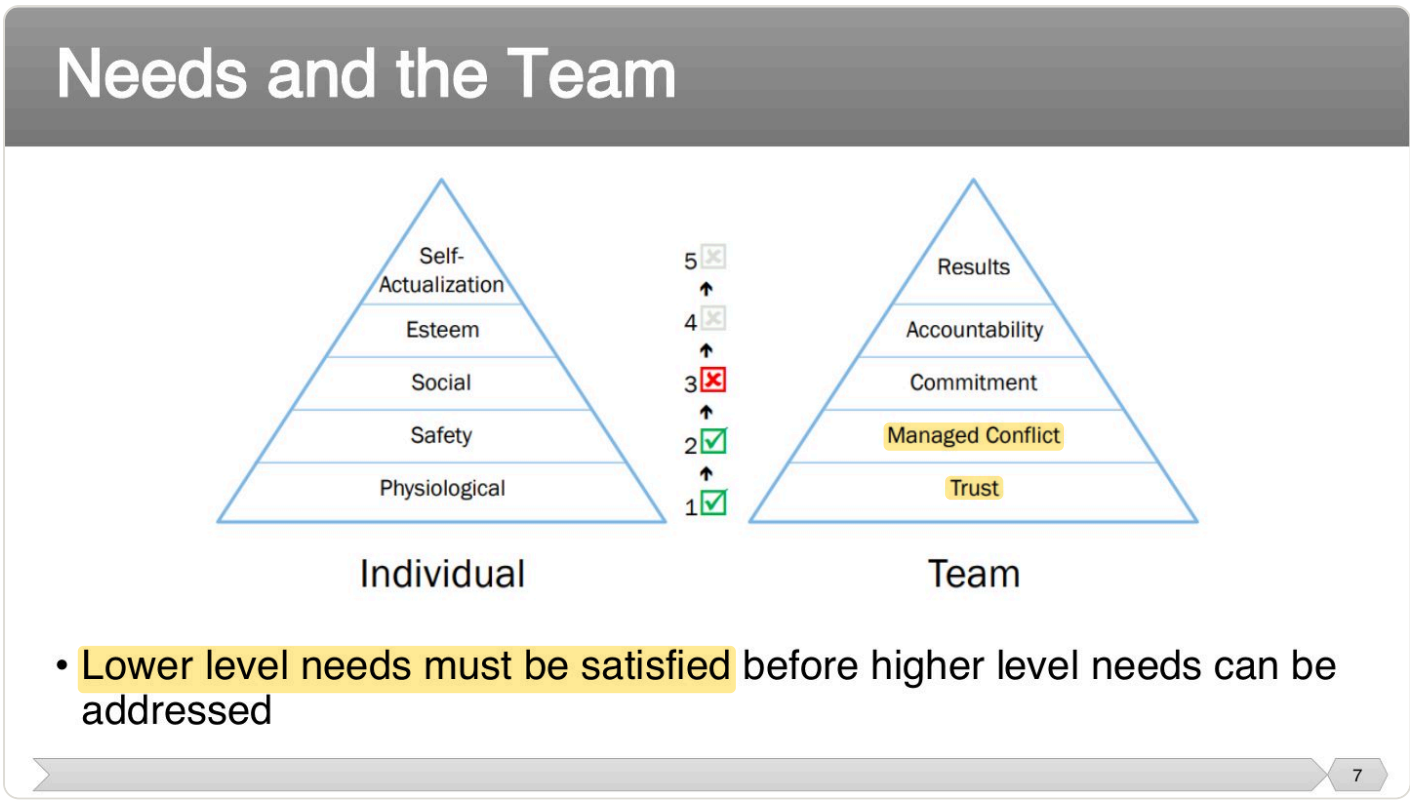
A **Expert and referent**

B Reward and penalty

C Formal and penalty

D Formal and reward

 **Needs and the Team**




NOTES

1. The rule from Maslow: **lower-level needs must be satisfied before higher ones** can be addressed. On the left is Maslow's individual pyramid; on the right the **team needs pyramid** (Lencioni).

2. The team pyramid from the bottom: **trust**, then **managed conflict**, then **commitment**, then **accountability**, then **results**. You can't have healthy conflict without trust, or commitment without having debated the decision.

3. **Results** sit at the top: the team's collective goals. The ticks on the slide show a team that has secured trust and conflict but is stuck at commitment, so the PM works on that layer next.

## EXAM ANGLE

Past exam · Midterm 251 · V1 · MCQ 8 (also V2 MCQ 8) Q1

Which of the following is considered the foundation of teamwork?

A Accountability

**B Trust**

> Trust is the base of the team pyramid: trust, managed conflict, commitment, accountability, results.

C Commitment

D Consensus

Prediction Q2

In the team needs pyramid, which layer sits directly above trust?

A Commitment

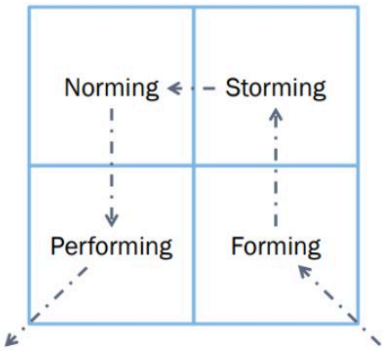
B Accountability

**C Managed conflict**

D Results

 **Tuckman's Model for Team Development**

# Tuckman's Model for Team Development




1. Forming: The group comes together and gets to initially know one another and form as a group.
2. Storming: A chaotic competing for leadership and trials of group processes (compete for influence or control, often without clear structure or authority)
3. Norming: Eventually agreement is reached on how the group operates
4. Performing: The group practices its craft and becomes effective in meeting its objectives
5. Dissolving/Adjourning



NOTES

1. New people are careful and polite, and they look to the leader for direction. **Forming** feels calm, but little gets done yet, because people are still working out what's expected of them.
2. Once people relax, they start to push back and test who has influence. **Storming** feels like something's going wrong, but it's a normal stage. The PM's job here is to surface and manage the conflict, not suppress it.
3. Arguments settle and the group agrees how it will work, such as how decisions get made. **Norming** is when people start saying "we" rather than "I" and accept each other's strengths.
4. People stop spending energy on how to work together and spend it on the work. **Performing** is what you're aiming for, and the team needs little supervision at this point.
5. People may feel loss when a good team breaks up, so a proper ending matters: **adjourning** is when you wrap up and capture what was learned. Order: form → storm → norm → perform → adjourn.

TIPS & TRICKS

- The order matters, and they rhyme: **F**orm, **S**torm, **N**orm, **P**erform, then **A**djourn.

EXAM ANGLE

Past exam · Midterm 251 · V1 · MCQ 7 (also V2 MCQ 7) Q1

According to Tuckman's model of team development, which stage is characterized by members competing for leadership and struggling with group processes?

**A Storming**

> Storming is when members compete for leadership and test group processes; norming comes after, when agreement is reached.

B Norming

C Adjourning

D Performing

Prediction Q2

The stage where the team reaches agreement on how it will operate is:

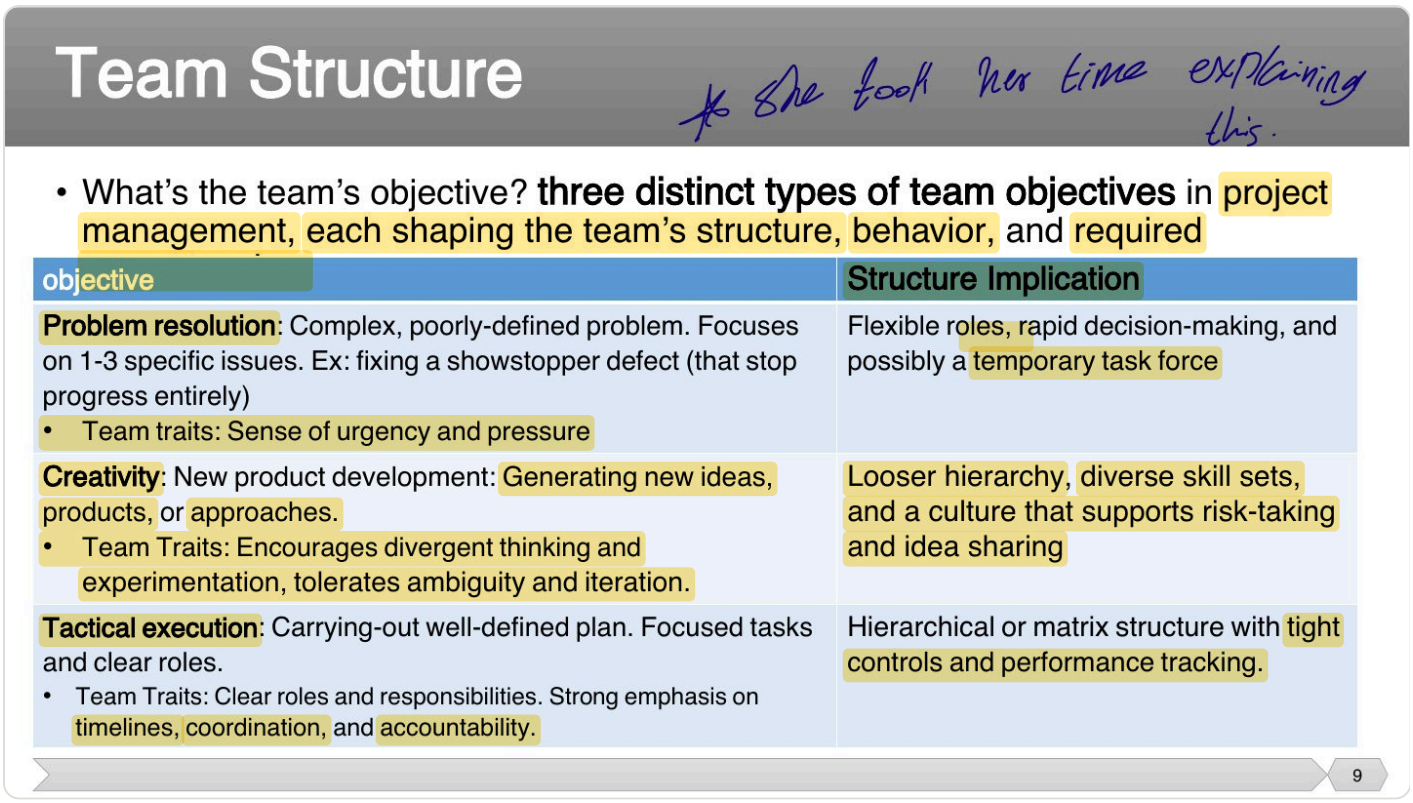
A Performing

**B Norming**

C Forming

D Storming

 **Team Structure**




NOTES

1. A production outage isn't the time for meetings and approvals. **Problem resolution** needs a small group that can decide fast. Once it's fixed, the team usually disbands, which is why it's a **temporary task force**.

2. You can't order someone to have a new idea. **Creativity** needs freedom to try and fail, so a **looser hierarchy** and mixed backgrounds help, and punishing mistakes would kill it.

3. Here the plan already exists, so success depends on precise execution. **Tactical execution** favors clear roles and **tight controls**, and creative freedom would only get in the way. (The instructor noted she spent time explaining this slide.)

## EXAM ANGLE

Past exam · Midterm 251 · V1 · Case Q4 (also V2) Q1

As the project manager, you must establish the team structure for this project. What team structure would you create to ensure smooth development, stakeholder engagement, and integration with Edugate, and why?

**Answer:** A cross-functional Agile team: a Project Manager, a Product Owner from Edugate/IT, developers, a UI/UX designer, QA testers, a faculty advisor representative and a student representative (1.0). This puts every stakeholder's voice inside the team, balances usability against compliance, and supports iterative delivery (0.5).

Prediction Q2

A team must roll out an already-designed module on a fixed timeline. Which objective type and structure fit?

**Answer:** Tactical execution: carrying out a well-defined plan. Use a hierarchical or matrix structure with clear roles, tight controls and performance tracking, because the work is known and the risk is in coordination and deadlines, not in discovering what to build.

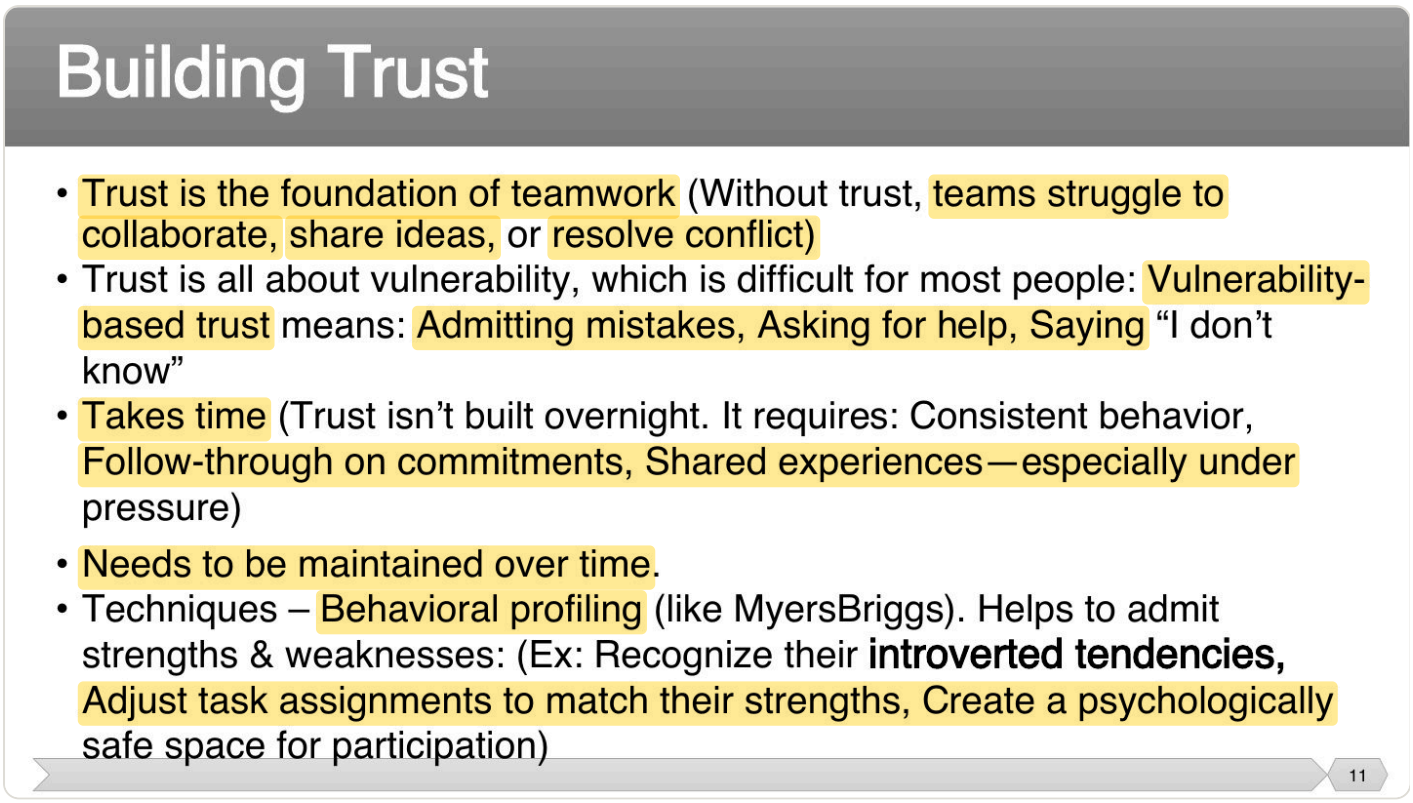
## SLIDE 10 / 77 Slide 10




## NOTES

1 Section divider: **Building Trust**, the base of the pyramid.

 **Building Trust**




NOTES

1. Everything else in the pyramid depends on this. If people don't feel safe, they hide problems and avoid arguments, so **trust** comes first as the foundation.

2. Most people assume that trust means "you're reliable". This kind goes further: it means being able to say "I messed up" and know nobody will use it against you. This list was a real midterm question, so learn **vulnerability-based trust**.

3. Words can't build it fast, as people need to see you keep your word several times. So **it takes time**, and since one broken promise can undo months, it must be **maintained**.

4. Knowing how a teammate tends to work removes a lot of misunderstanding: "she's direct, not rude". **Behavioral profiling** is useful as a conversation starter, not as a label, and it lets the PM give people work that suits them.

EXAM ANGLE

Past exam · Midterm 251 · V1 · Short Q2 (also V2 Short Q1) Q1

What does vulnerability-based trust involve?

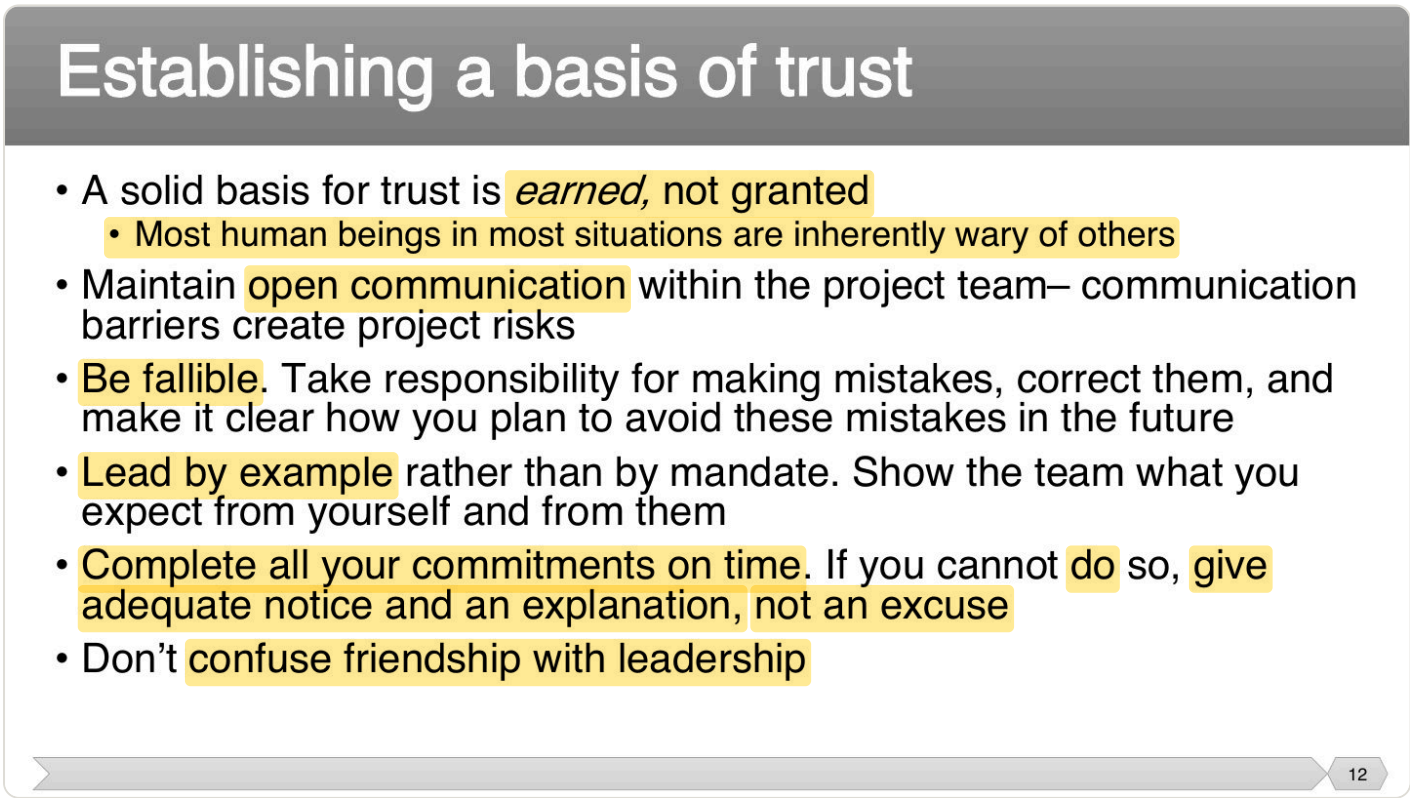
**Answer:** Team members are comfortable being open with each other: admitting mistakes and weaknesses, asking for help, and saying "I don't know", without fear that it will be used against them. It takes time to build through consistent behavior, follow-through and shared experiences under pressure, must be maintained, and is the foundation of the team pyramid.

Past exam · Final 242 · Q5.2 Q2

How does fostering trust and managing conflict within a software project team contribute to improved collaboration and project success?

**Answer:** Trust lets members admit mistakes, share information and ask for help, so problems surface early and people cooperate instead of protecting themselves. Managed (healthy) conflict means ideas are debated openly and respectfully, which gives better technical decisions and avoids groupthink or disputes that turn personal. Together they lead to real commitment and collaboration, which drives quality and project success.

SLIDE 12 / 77 **Establishing a basis of trust**




NOTES

1. A title makes people obey, but it doesn't make them trust you. You have to show them through what you do, so trust is **earned, not granted**, starting from a natural wariness.
2. If people can't tell you things, you find out about problems only when it's too late. So **open communication** is risk management, and not just being friendly.

3. If the PM hides their own mistakes, everyone learns to hide theirs. **Be fallible** by owning an error openly, because that makes it safe for others to do the same.
4. People watch what you do far more than what you say. A PM who asks for punctuality and arrives late has told them what's really expected. That's why you **lead by example**.
5. Nobody expects perfection, but they do expect honesty. Telling people in advance that you'll miss a **commitment** keeps trust, while a surprise and an excuse damages it.
6. **Friendship ≠ leadership**: warmth helps, but leading needs clarity and boundaries (repeated on slide 76).

EXAM ANGLE

Prediction Q1

Which behavior does the slide recommend for establishing trust?

- A **Admit mistakes and explain how you will avoid them**
- B Give excuses when commitments slip
- C Lead by mandate
- D Hide mistakes to protect your authority

SLIDE 13 / 77 **Slide 13**

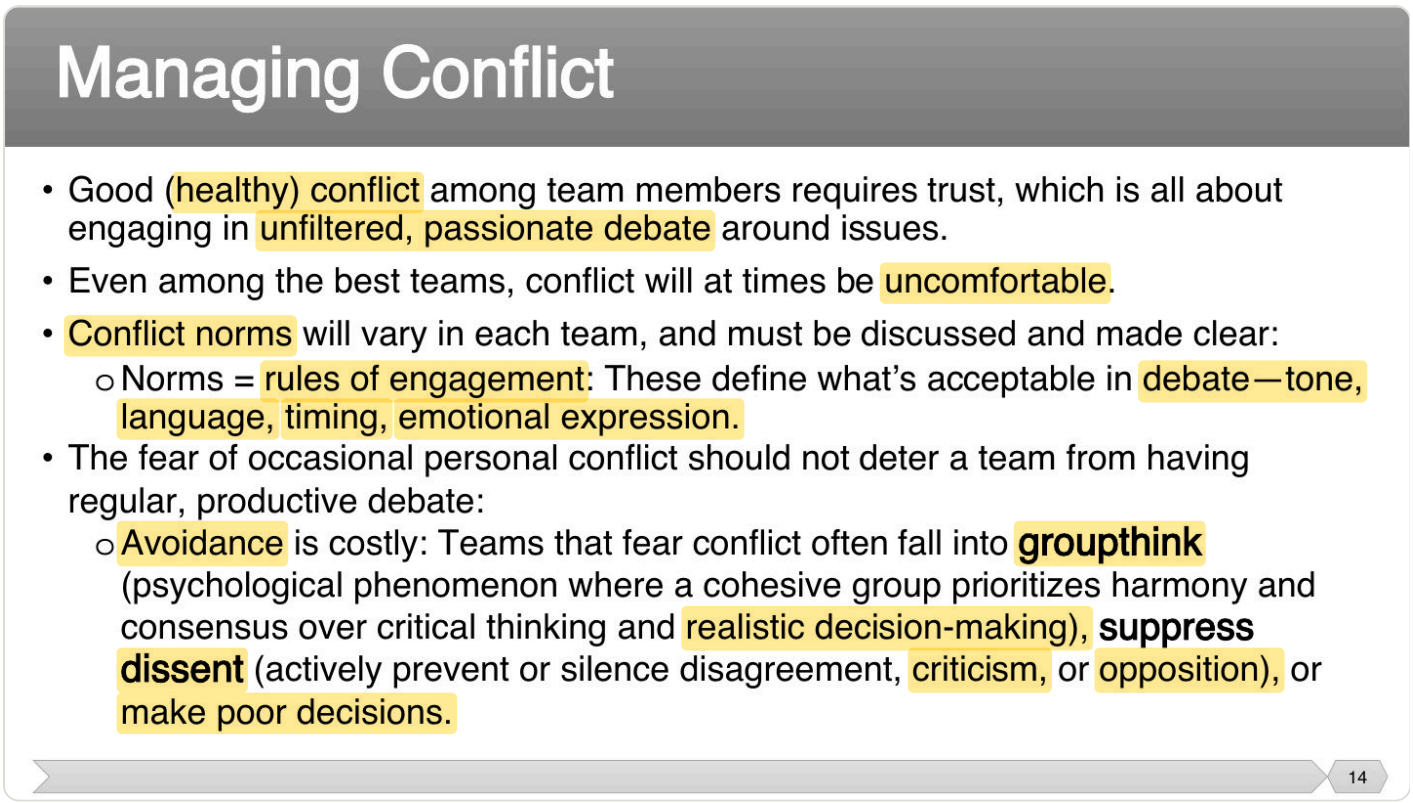


NOTES

1. Section divider: **Managing Conflict**, the second layer.

---

## SLIDE 14 / 77 Managing Conflict




NOTES

1. Think of healthy debate as a fight about ideas where nobody's attacking the person. It only works if you trust that disagreement won't be used against you, which is why **healthy conflict** sits above trust.

2. If everyone is always comfortable, tough topics probably aren't being raised. So some **uncomfortable** moments are a good sign, not something to fix.

3. What's normal in one team feels rude in another. Agreeing the rules up front, such as "challenge the idea, not the person", stops arguments about how to argue. That's what **conflict norms** are for.

4. If nobody challenges a weak design because it would feel awkward, the mistake ships. That's the cost of **avoidance**, and **groupthink** is what it looks like: everyone nods, and nobody really agrees.

EXAM ANGLE

Past exam · Midterm 251 · V1 · Short Q3 (also V2 Short Q3) Q1

Why is healthy conflict important in teams?

**Answer:** Healthy conflict is open debate about ideas, not people. It brings out different perspectives so the team makes better decisions, avoids groupthink and artificial harmony, and surfaces problems early. Because everyone has been heard, members are more willing to commit to the final decision even if they disagreed. It needs trust underneath and agreed conflict norms.

Prediction Q2

A team values harmony so much that it stops critically evaluating ideas. This is:

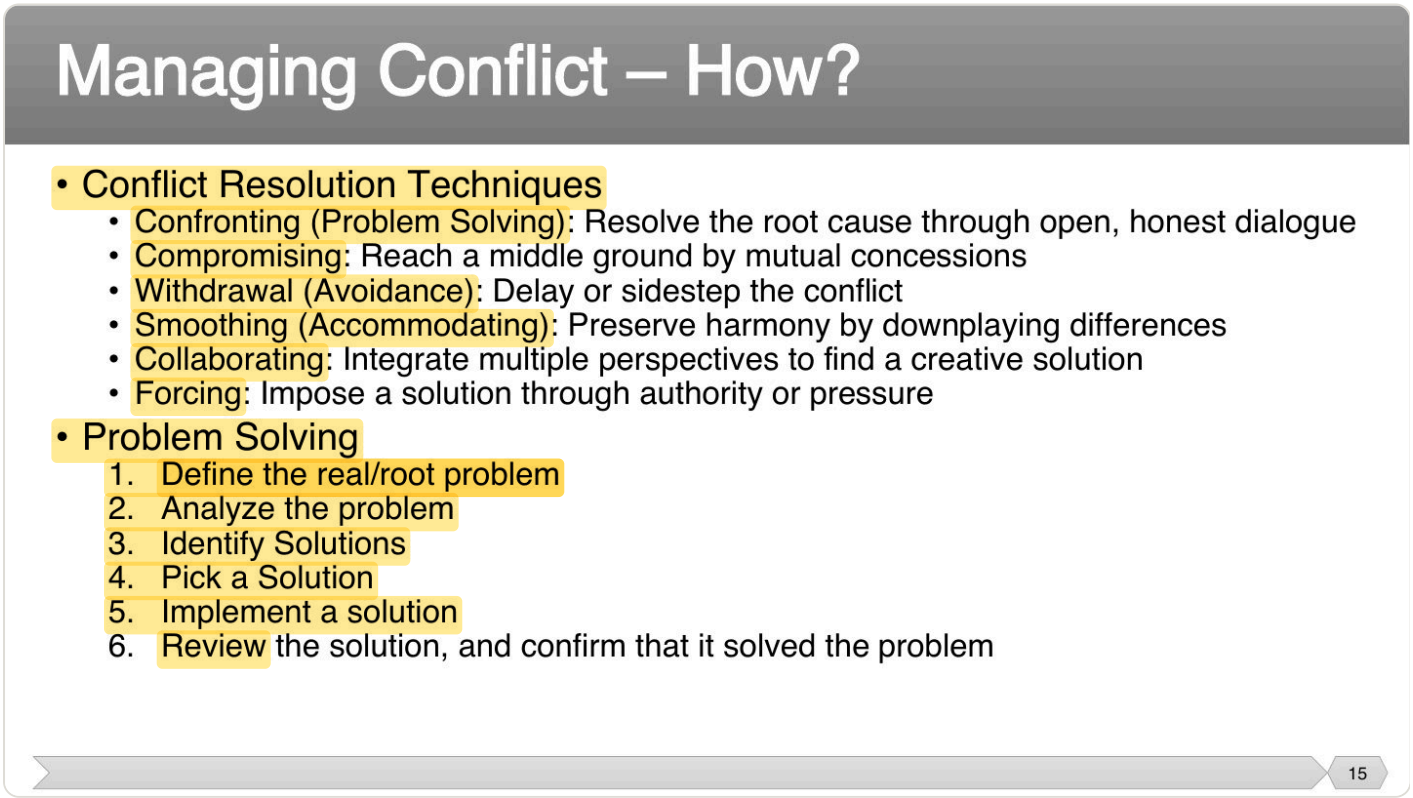
A **Groupthink**

B Accountability

C Storming

D Healthy conflict

SLIDE 15 / 77 **Managing Conflict – How?**




NOTES

1 This asks “what’s really causing this?” and tackles that. It takes longer, but the problem doesn’t come back, which is why **confronting** isn’t the same as being aggressive.

2. **Compromising**: both sides give something up to reach a middle ground. Quick, but nobody gets what they needed.
3. These two feel peaceful, but the cause stays there and tends to return bigger. **Withdrawal** can buy time when emotions are high. **Smoothing** can suit minor issues, or when the relationship matters more than the point.
4. Each side's view contributes something, so you may end up with a better answer than either started with. **Collaborating** takes time, so save it for decisions that matter.
5. Sometimes there's no time to discuss, like during an outage. **Forcing** gets a quick decision, but people who were overruled won't feel committed, so use it rarely.
6. Notice the first step: many teams solve the wrong problem because they jump to answers. The final step is the one often skipped: **review** whether it actually fixed the issue.

TIPS & TRICKS

- Sentence, in slide order: **C**ool **C**ats **W**ill **S**till **C**alm **F**ights. Confronting, Compromising, Withdrawal, Smoothing, Collaborating, Forcing.

## EXAM ANGLE

Past exam · Midterm 261 · Short — Q1

Team members disagree about the best solution to a problem. How can the project manager encourage collaboration? (Saying "negotiation" alone is not enough: explain.)
**Answer:** Treat it as healthy conflict and use collaborating/confronting (problem solving) rather than forcing or smoothing. Bring the people together for an open discussion focused on the problem, not on people; define the real problem and agree the criteria for a good solution; ask each person to present their option with its reasoning and data; weigh the pros and cons of each option together; then converge on the option that best serves the project, or combine ideas into a better one. Work with those who preferred other options so they buy in. This gives a better decision and real commitment because everyone was heard.

Prediction — Q2

A PM ends a debate by declaring "we use my design, end of discussion." Which technique is this?

A Compromising

B Smoothing

C **Forcing**

D Confronting

# Managing and resolving conflict

- Every team encounters conflict: successful teams resolve conflicts, while unsuccessful teams force conflict to the background
- Disagreement is not the same as conflict: conflict involves a hardening of position and associated intractability (insolvability)
- Virtually all conflicts can be traced to a clash between two parties
- To preserve a team's effectiveness, those outside the conflict must assume a neutral stance, even if they themselves agree with one of the parties to the conflict
- One of those outside the conflict must take on the role of conciliator; he or she must mediate between the two parties to defuse the situation
- The goal is to reduce the conflict to a disagreement which can then be addressed through consensus
- Strive for reconciliation between the conflicting parties, if possible

16

## NOTES

1. Conflict that's ignored doesn't go away, as it comes back in other forms, such as people refusing to cooperate. The difference is whether a team **resolves** it.
2. People can disagree and still work things out. It's **conflict** once they've dug in, and each stops listening. Spotting that shift tells you when to step in.
3. Taking sides only spreads the fight and splits the team, so others stay **neutral**. The **conciliator** helps both sides listen again until it's an ordinary disagreement once more.
4. Finally, aim for **reconciliation** between the two parties so the working relationship survives.

## EXAM ANGLE

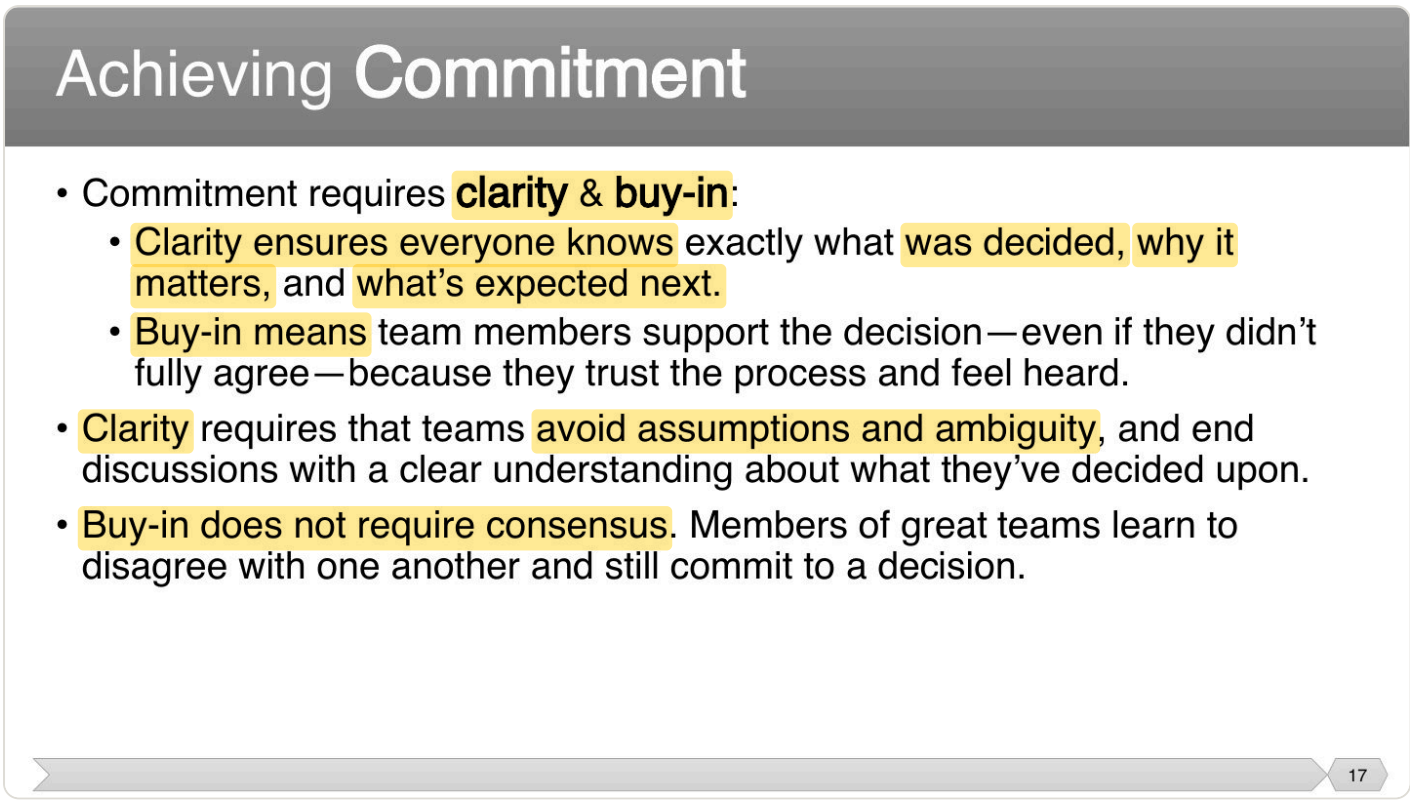
Prediction Q1

Disagreement and conflict are the same thing.

T True

F **False**

> Conflict involves a hardening of position and intractability; the goal is to reduce it back to a disagreement.




NOTES

1. Ever left a meeting and found that everyone understood something different? **Commitment** fails that way, so a decision needs to be both understood and accepted.
2. A simple habit fixes most of this: end each discussion by stating what's been decided and who does what next. Without that **clarity**, people act on their own assumptions.
3. People can accept a decision they argued against, as long as they feel it was fair and their view was heard. That's why **buy-in** comes after debate rather than instead of it.
4. Waiting for everyone to agree can freeze a team. Great teams argue, decide, and then back the decision ("disagree and commit"). A common trap: "commitment requires everyone to agree" is false, because **buy-in does not require consensus**.

TIPS & TRICKS

- **CuB**: Clarity, Buy-in.

## EXAM ANGLE

Past exam · Final 242 · Q5.1 — Q1

In the context of the Team Needs Pyramid, explain how a project manager can foster commitment among team members.

**Answer:** In the pyramid (trust → managed conflict → commitment → accountability → results), commitment sits on trust and healthy conflict: people commit when they trust each other and have had their say. The PM fosters it with clarity (clear decisions, goals, roles and deadlines, confirmed at the end of each meeting, with no assumptions) and buy-in (letting everyone voice their views so they support the decision even if they disagreed; consensus isn't required). Recording decisions and owners (e.g. in a RACI) then lets commitment be followed by accountability.

Prediction — Q2

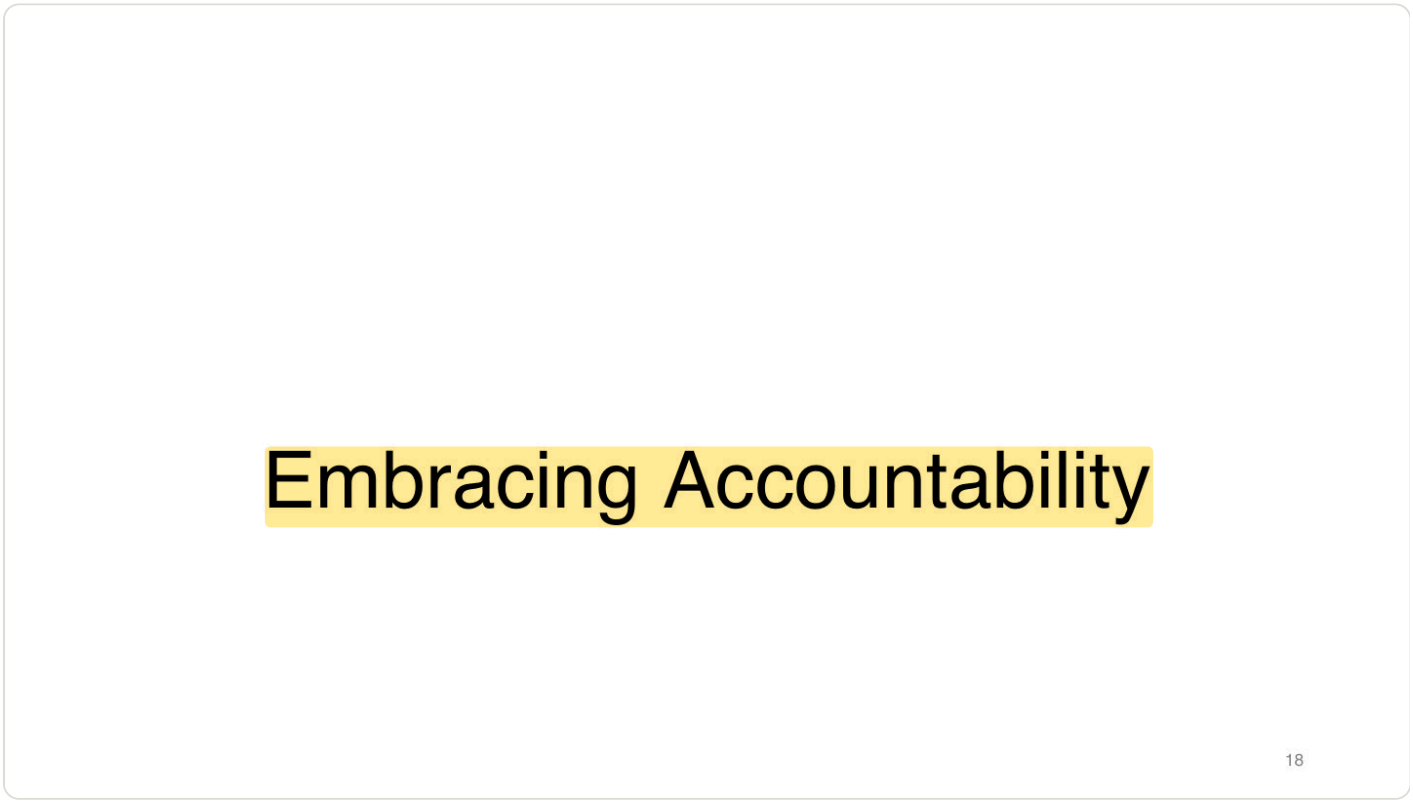
Team buy-in requires that every member agrees with the decision.

T True

**F False**

> Buy-in does not require consensus.

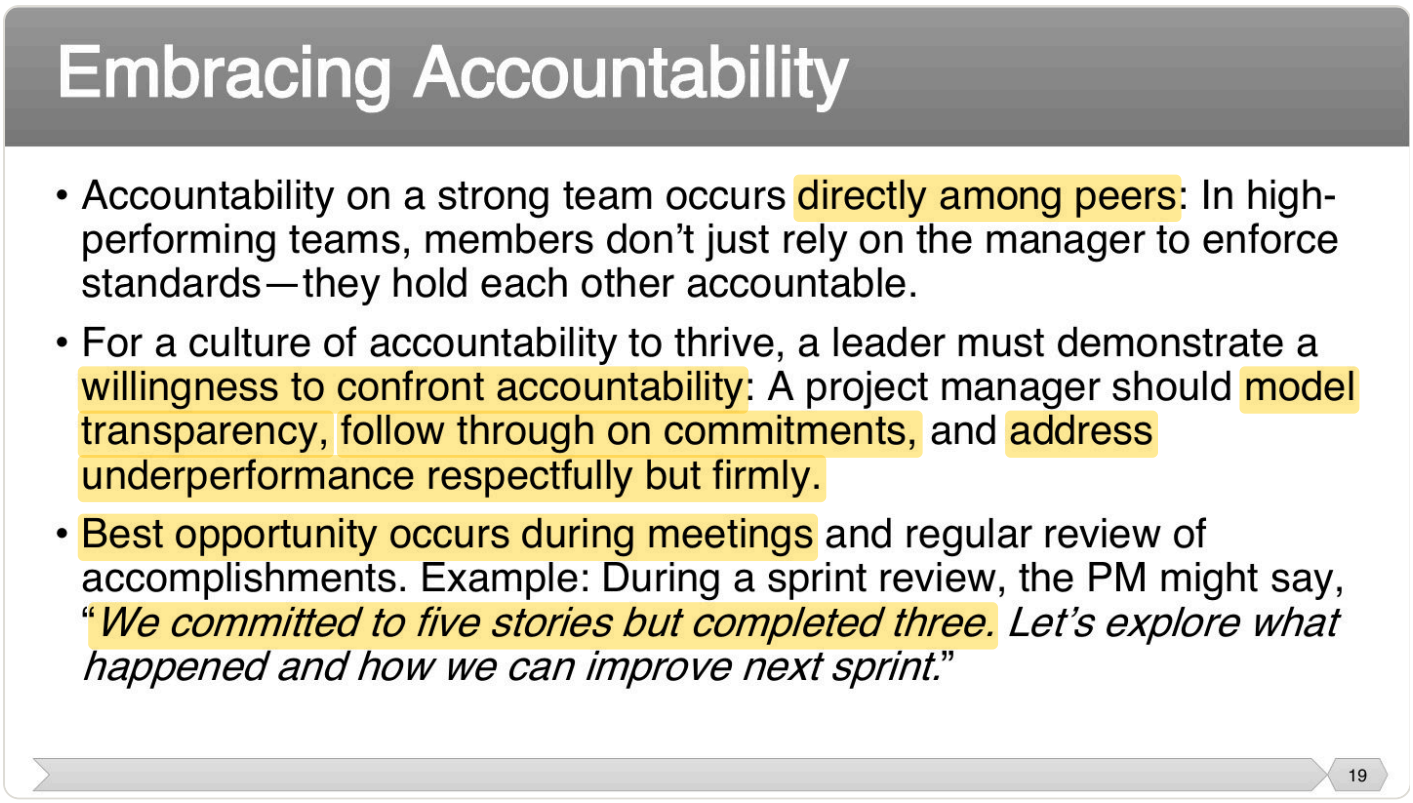
## SLIDE 18 / 77 Slide 18




## NOTES

1. Section divider: **Embracing Accountability**, the fourth layer.

 **Embracing Accountability**




NOTES

1. If only the manager enforces standards, you have one person chasing everyone. When teammates say "this isn't up to what we agreed" to one another, problems get fixed faster. That's accountability **among peers**.
2. If the PM lets poor work slide, the team learns that standards are optional. So the leader needs a **willingness to confront** it, and doing that respectfully shows others how.
3. Reviews work because the facts are in front of everyone. Note the tone in the slide's example, about the commitment and what happened rather than who failed, in **meetings and regular reviews**.

## EXAM ANGLE

**Past exam · Midterm 251 · V2 · Short Q2** Q1

What's the best time to reinforce accountability?
**Answer:** During meetings and regular reviews of accomplishments, such as sprint reviews and status meetings, where what was committed is compared with what was done (e.g. "we committed to five stories but completed three; let's explore why"). Make commitments clear and public at the start and address a missed commitment when it happens rather than waiting for the end of the project, so peers hold each other accountable continuously.

**Past exam · Midterm 251 · V1 · Case Q1 (also V2)** Q2

During a sprint deployment, a key feature was discovered to be not thoroughly tested and caused issues after release. As the project manager, what steps would you follow to address this issue and ensure accountability within the team?
**Answer:** Use a RACI matrix to establish who was Responsible, Accountable, Consulted and Informed for testing that feature (0.5). Hold a retrospective to find the root cause (0.25). Make the responsible roles clear without blaming individuals (0.25). Take corrective action: release a hotfix, redistribute tasks, and add automated testing so it can't recur (0.25). Inform the stakeholders of the recovery plan (0.25).

**Past exam · Final 242 · Q5.3** Q3

Your software project team is experiencing delays due to unclear task ownership and inconsistent updates. As the project manager, how would you apply a team-based approach to improve accountability and transparency within the team? Describe specific steps you would take to enhance collaboration and increase the project's chances of success.
**Answer:** 1) Clarify ownership with a RACI matrix so every task has a Responsible person and exactly one Accountable person. 2) Hold short daily stand-ups (done, next, blocked) so updates are consistent. 3) Put all tasks on a shared board (e.g. Jira or Kanban) with visible goals and deadlines so progress is transparent. 4) Review commitments against results in regular reviews, encourage peer accountability, and run retrospectives to fix what caused the delays.

 **Embracing Accountability**

# Embracing Accountability

- RACI – Responsible, Accountable, (Support), Consulted, Informed

| | Project Manager | Project Sponsor | Developer 1 | Tester 1 |
|---|---|---|---|---|
| Activity 1 | A | I | R | |
| Activity 2 | R | S | | |
| Activity 3 | C | | R | S |
| Activity 4 | RA | | | C |
| Activity 5 | A | R | S | |

Note: "Support" is not part of the original RACI acronym but is sometimes ad
in variants like RASCI to reflect collaborative roles more accurately.

20

NOTES

1. **RACI**: **R**esponsible does the work; **A**ccountable owns the outcome and signs off (exactly **one** A per activity); **C**onsulted gives input before or during the work (two-way); **I**nformed is kept updated (one-way). Several R's on one activity are fine.
2. It's worth a mention because some companies use it, so you won't be confused if you see **RASCI**. It's for people who help out on a task without being responsible for it.
3. In the example, Activity 4 has the project manager as **RA**: one person can be both Responsible and Accountable for a task they do and own.

EXAM ANGLE

Past exam · Midterm 251 · V1 · Short Q6 (V2 asked T2 and T5) Q1

Your team has been assigned to develop a mobile app for a local fitness center. The app will allow users to book classes, track workouts, and receive personalized fitness tips. The project is expected to be completed in 8 weeks. Fill in the RACI assignments for T1 and T6 only.

**Answer:** These cells come from the instructor's key. The person who does the work is R (the client representative for requirements, the UI/UX designer for the interface, the lead developer for building and launching, the marketing specialist for promotion). Whoever's expertise is needed is C, and everyone else is I. The key makes the PM Accountable (A) for requirements and launch, the two tasks that commit the project; on the other tasks the PM is kept Informed.

Prediction Q2

In RACI, who owns the outcome and signs off on the task?

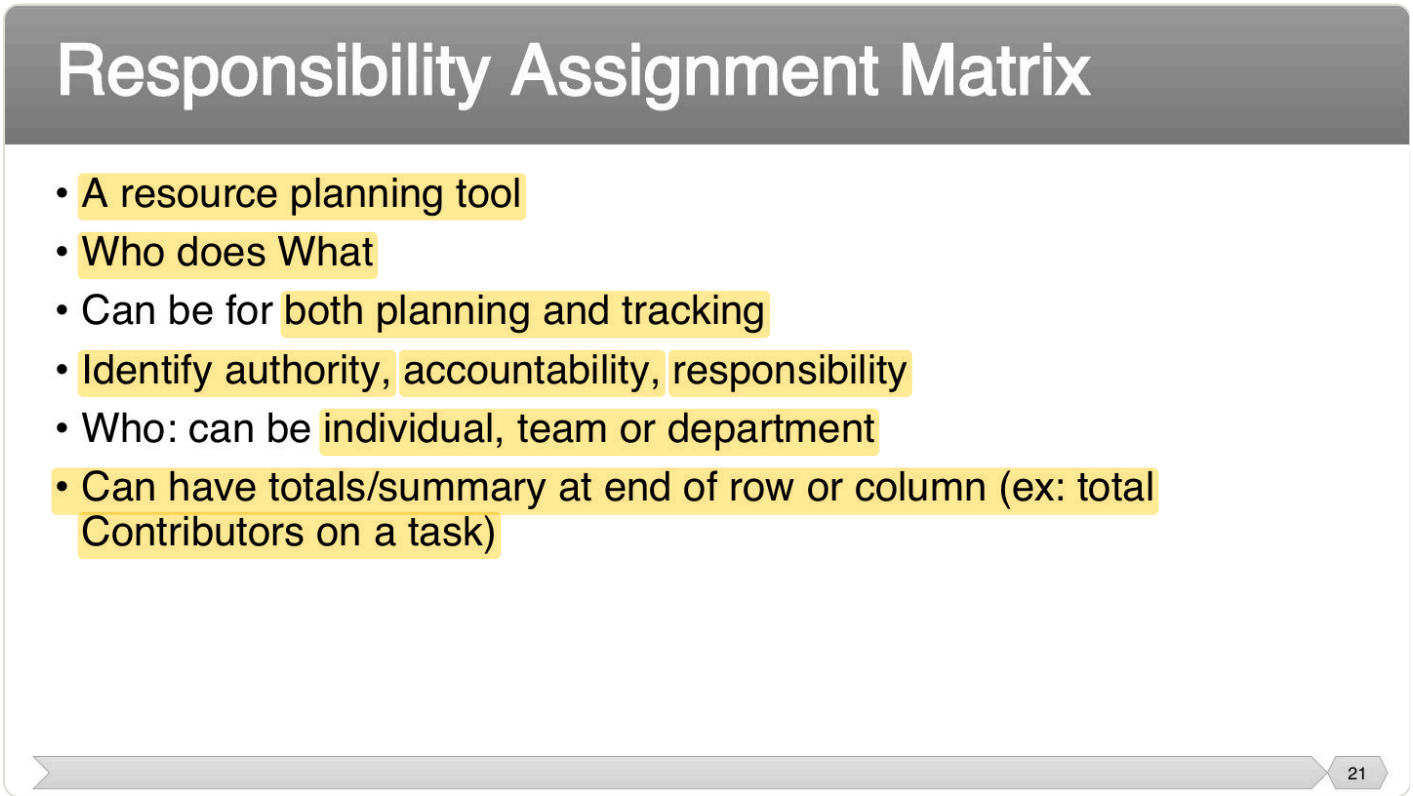
A Consulted

B Informed

C Responsible

D **Accountable**

SLIDE 21 / 77 **Responsibility Assignment Matrix**



## NOTES

1. If nobody clearly owns a task, it tends to fall through the cracks, and two people may each assume the other has it. A **RAM** makes that visible, and **RACI** is just one way of filling one in.
2. During **planning** you assign the work, and during **tracking** you can see where a task has no owner. So it keeps its value after the first week.
3. On a large project you may not assign to named people yet, so you can start with a **team or department**. Row totals show if one person or task is overloaded.

## EXAM ANGLE

Prediction Q1

What does a Responsibility Assignment Matrix show?

A Which risks are most severe

> That's a risk matrix.

B How far the project is behind

> That's schedule tracking.

C How much each task costs

> That's a cost estimate.

**D Who does what**

> It's a resource planning tool for this.

Prediction Q2

A RAM can be used only during planning.

T True

> The slide names both uses.

**F False**

> It is used for both planning and tracking.

 **Simple RAM**

# Simple RAM



| ROLE / Project Deliverable (or Activity) | Executive Sponsor | Project Sponsor | Steering Committee | Advisory Committee | Role #5 | Project Manager | Tech Lead | Functional Lead | SME | Project Team Member | Developer | Administrative Support | Business Analyst | Role #4 | Role #5 | Consultant | PMO | Role #3 | Role #4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Project Leadership | | | | | Project Team Members | | | | Project Sub-Teams | | | | | | External Resources | | | |
| **Initiate Phase Activities** | | | | | | | | | | | | | | | | | | | |
| Request Review by PMO | A/C | R/A | | | | R/A | A/C | | C | | | | | | | | | | |
| Submit Project Request | | | | | | R | | | | | | | | | | | A | | |
| Research Solution | I | | | | | R/A | A/C | A/C | C | | | | C | | | C | | | |
| Develop Business Case | I | A/C | I | I | | R/A | C | C | C | | | | C | | | C | C | | |
| **Plan Phase Activities** | | | | | | | | | | | | | | | | | | | |
| Create Project Charter | C | C | | | | R/A | C | C | C | | | | C | | | C | | | |
| Create Schedule | I | I | I | I | | R/A | C | C | C | C | C | C | C | | | C | I | | |
| Create Additional Plans as Required | I | I | I | | | R/A | | | | I | I | I | I | | | C | I | | |
| **Execute Phase Activities** | | | | | | | | | | | | | | | | | | | |
| Build Deliverables | C/I | C/I | C/I | C/I | | | R/A | R/A | R/A | R/A | R/A | | | | | A/C | | | |
| Create Status Report | I | I | I | I | | R/A | R/A | R/A | R/A | | | | | | | C | I | | |
| **Control Phase Activities** | | | | | | | | | | | | | | | | | | | |
| Perform Change Management | | C | C | C | | R | A | A | A | | | | | | | C | I | | |
| **Close Phase Activities** | | | | | | | | | | | | | | | | | | | |
| Create Lessons Learned | C | C | C | C | | R/A | C | C | C | C | C | C | C | | | C | C | | |
| Create Project Closure Report | I | I | I | I | | R/A | I | I | I | I | I | I | I | | | | I | | |

RACI – Responsible, Accountable, (Support), Consulted, Informed

22

## NOTES

1. A real RAM: project deliverables grouped by phase (initiate, plan, execute, control, close) down the side, and roles grouped as project leadership, team members, sub-teams and external resources across the top. Each cell holds a RACI letter, so you can read across a row to see who is involved in a deliverable, or down a column to see one role's workload.

## EXAM ANGLE

**Prediction** Q1

How do you see one role's workload on a RAM?

**A** Look at the risk register

> That's a different document.

**B** Add up the budget

> The RAM holds RACI letters, not money.

**C** Read across a deliverable's row

> That shows who's involved in one deliverable.

**D** **Read down that role's column**

> Each column is a role.

 **Skills Matrix**

# Skills Matrix

- Another resource planning tool
- Resources on one axis, skills on other
- Skills can be high level or very specific
- Cells can be X's or numeric (ex: level, # yrs.)






## NOTES

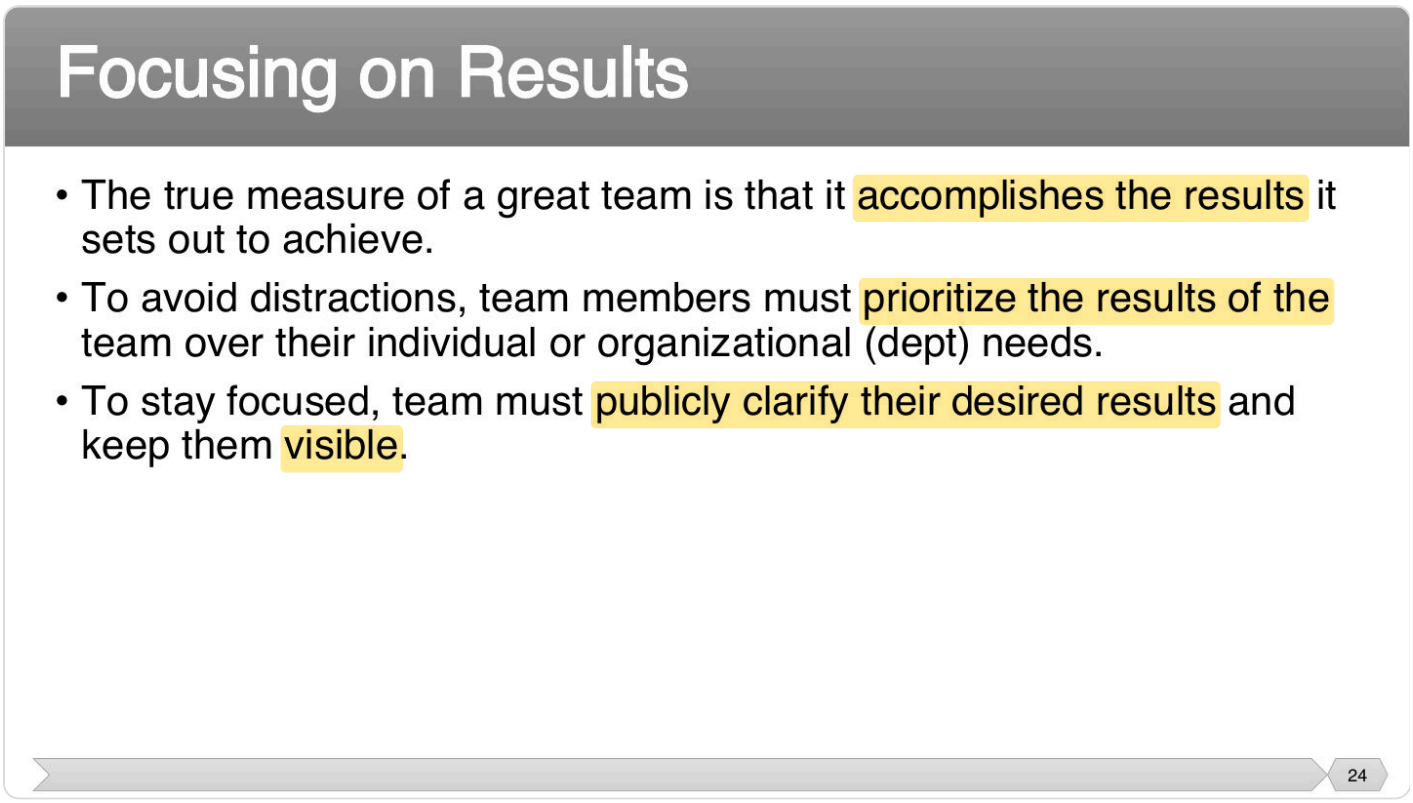
1. A RAM says who is doing what, while a **skills matrix** says who is able to. Use it before assigning work, so you aren't guessing about abilities.
2. Skills can be high level or very specific (PHP, MySQL, React); cells can be X's or **numeric** levels (the legend: expert, practitioner, novice). Use it to assign tasks to the right people and to spot gaps, e.g. only one person is an expert in a key technology, which is a key-person risk.

## EXAM ANGLE

Past exam · Final 242 · Q2.3 — Q1

You are responsible for managing a diverse team with varying availability and skill sets. Describe two strategies you would use to allocate your team's resources efficiently across tasks. How would you adjust your resource plan if one hardware specialist becomes unavailable for 4 weeks?

**Answer:** 1) Build a skills matrix and assign tasks to the people whose skills match, recording owners in a RAM/RACI. 2) Check availability per period and level the workload so nobody is overallocated, using float on non-critical tasks. If the hardware specialist is out for 4 weeks: move their non-critical tasks later within their float, reassign what others can do (using the skills matrix to find the next-best person), bring in a contractor or borrow a specialist for critical work, and update the schedule and inform stakeholders if the critical path moves.




NOTES

1. Everything below this exists for one reason. A team can have trust and good meetings and still have failed if it doesn't deliver, so it's judged by **results**.
2. A developer might want a clever feature for their CV, and a department might want its own goal. **Team results** come first, or everyone wins individually and the project loses.
3. If a goal lives in a document nobody opens, it can't guide daily decisions. A public, **visible** goal also makes it awkward for anyone to quietly drift from it.

EXAM ANGLE

Prediction Q1

A developer pushes a pet feature that helps their CV but delays the team goal. Which pyramid layer is the problem?

A Skills matrix

> That's a planning tool.

B Forming

> That's a Tuckman stage, not a pyramid layer.

**C Focus on results**

> Team results must come before individual needs.

D Trust

> The issue here is what's being prioritized.

Prediction Q2

How can a team stay focused on results?

A Stop reviewing progress

> That makes drifting more likely.

B Let each department set its own results

> Team results come first.

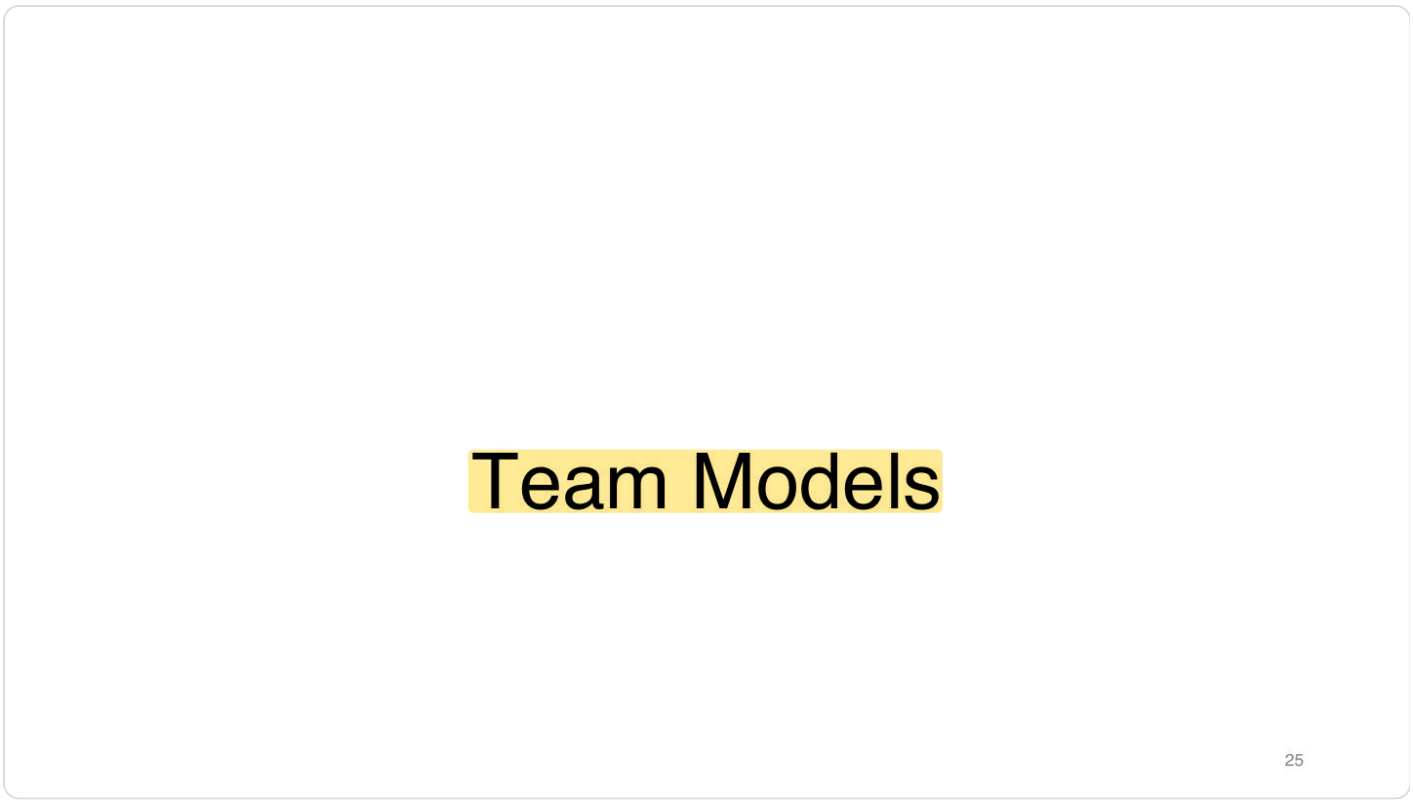
C Keep goals private

> That's the opposite.

**D Publicly clarify its desired results and keep them visible**

> A visible goal is hard to drift from.

SLIDE 25 / 77 **Slide 25**

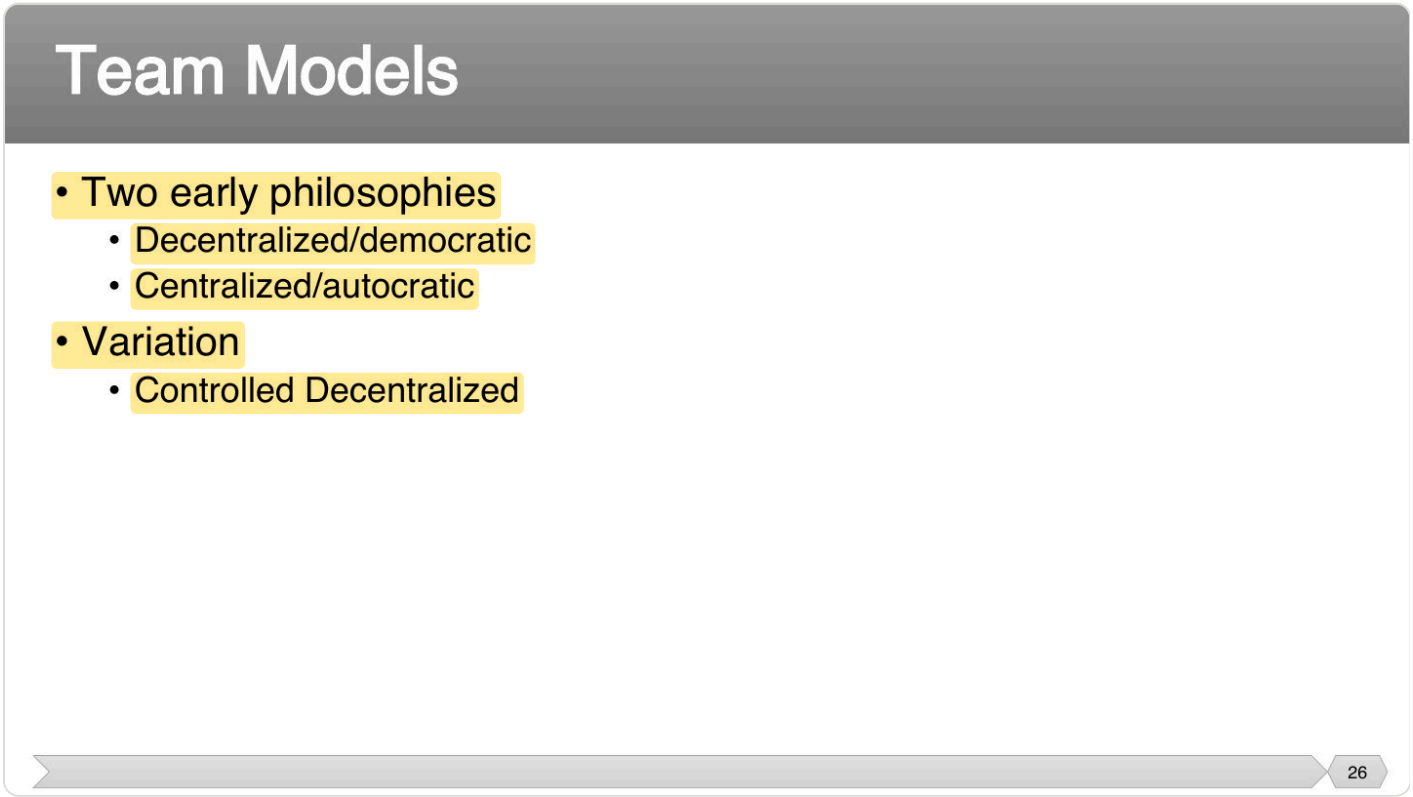



NOTES

1 Section divider: **Team Models**, the ways a software team can be organized.

---

SLIDE 26 / 77 **Team Models**



## NOTES

1. These are the two extremes. A **democratic** team gets better buy-in but decides slowly, while an **autocratic** one decides fast but depends on one person's judgment.
2. The variation in between, **controlled decentralized**, has a leader who coordinates while sub-groups decide within their own areas.

## EXAM ANGLE

Prediction

Q1

In which philosophy does one person decide?

A Decentralized or democratic

> There the team decides together.

**B Centralized or autocratic**

> One person makes the decisions.

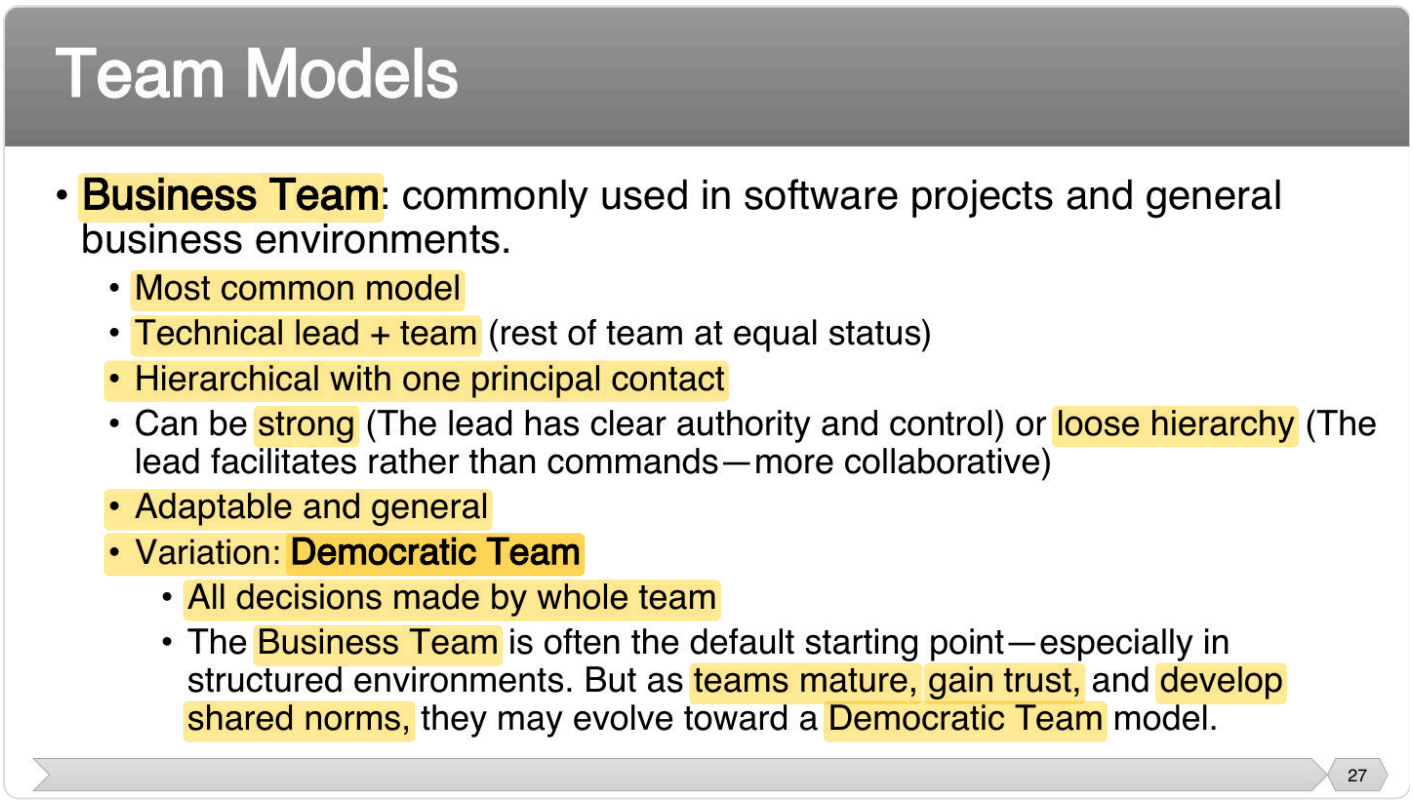
C Self-organizing

> That's the team deciding.

D Controlled decentralized

> A leader coordinates, while sub-groups decide.

 **Team Models**




NOTES

1. Why is it so common? It's simple, so everyone knows who to ask and who answers to outsiders. If a question doesn't mention anything special, the **business team** is the safe assumption.
2. One lead keeps responsibility clear, while the rest of the team stay equal. **Strong** suits a new or inexperienced team, and **loose** suits a skilled one. It's this flexibility that makes it general.
3. Trust has to come first. A new team asked to decide everything together would stall, so moving to a **democratic team** is something a team grows into.

TIPS & TRICKS

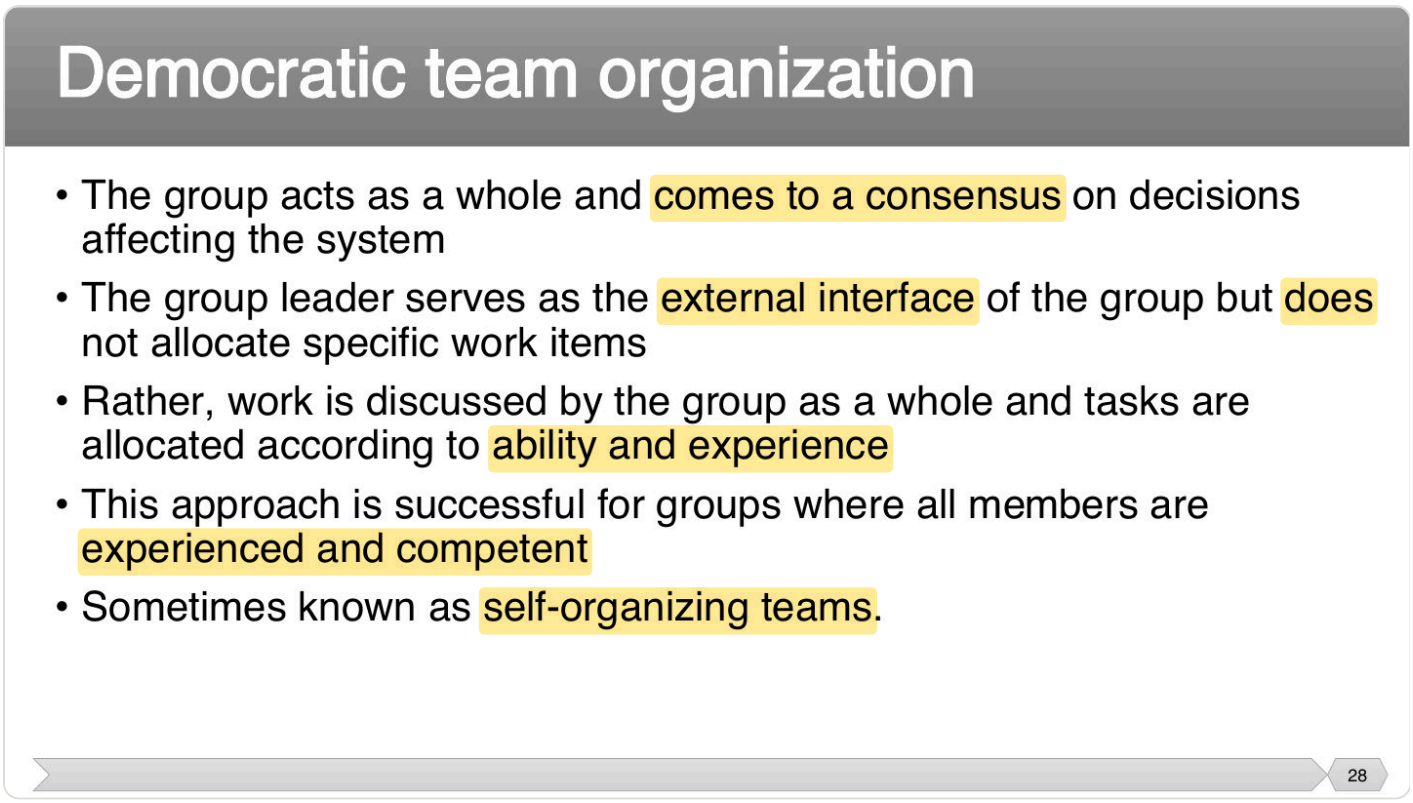
- Team models, in slide order: **B**usy **D**evelopers **C**ode **S**uper **S**oftware. Business, Democratic, Chief-programmer, Skunk works, SWAT.

EXAM ANGLE

Past exam · Midterm 251 · V1 · Short Q4 (also V2 Short Q4) Q1

What's the difference between a Business Team and a Democratic Team?
**Answer:** A business team has a technical lead who is the principal contact and makes or facilitates decisions, with the other members at equal status under the lead (a strong or loose hierarchy); it is the most common model. A democratic team acts as a group: decisions are made by consensus, the leader is only the external interface and does not allocate work, tasks are allocated by ability and experience, and it works best with experienced, competent members. Teams often move from business to democratic as trust and norms develop.

 **Democratic team organization**




NOTES

1. Nobody hands down a decision, so people commit to it, and it takes longer. That's the trade-off behind reaching **consensus**: strong buy-in in exchange for speed.
2. Someone still has to talk to the outside world, but that person isn't the boss. As an **external interface**, they pass on information, and the team decides who does what.
3. People take the work they're best at, rather than getting tasks by rank. That only works when everyone knows one another's **ability and experience**, as you can't allocate fairly otherwise.
4. If some members are beginners, they can't judge decisions well and will either stay quiet or slow things down. So this is the condition to remember: **all members experienced and competent**.

EXAM ANGLE

Prediction Q1

In a democratic team, what does the leader do?

A Removes the need for consensus

> The group reaches consensus.

B Assigns all the tasks

> That’s what the leader doesn’t do.

**C Acts as the external interface, without allocating specific work**

> Work gets allocated by the group.

D Writes all the core code

> That’s a chief programmer.

Prediction Q2

A team of beginners tries a democratic model and stalls. Why?

A It only works for hardware

> Nothing says so.

B Democratic teams must be large

> Size isn’t the condition.

C Democratic teams need a surgeon

> That’s the chief-programmer model.

**D It works when all members are experienced and competent**

> That’s the condition the slide gives.

# Agile programming groups

# Agile programming groups

- Agile programming groups are variants of democratic organization
- In Agile programming groups, some ‘management’ decisions are delegated to group members
- In XP, Programmers work in pairs and take a collective responsibility for code that is developed



NOTES

1. Agile teams estimate and assign their own work, because they’re closest to it. That’s why they count as **democratic** variants, though someone like a product owner still sets priorities.
2. In **XP**, programmers work in pairs and take **collective responsibility** for the code: anyone can change any part, and nobody “owns” a module.

EXAM ANGLE

Prediction Q1

In XP, who is responsible for the code?

A The customer

> The customer sets priorities, not code ownership.

B Only the person who wrote it

> Nobody owns a module alone.

C Only the tester

> Everyone shares it.

**D The whole team, collectively**

> Programmers take collective responsibility.

 **Team Models**

# Team Models

**Chief-Programmer Team** a.k.a. 'surgical team: From IBM in 70's: Puts a superstar at the top: Just as a surgeon performs the critical operation while being supported by anesthetists, nurses, and technicians, the **chief programmer** writes the core code while being supported by a carefully selected team of specialists.

- Backup Programmer/co-pilot (Equally skilled—ready to step in if needed, often reviews or shadows the chief.)
- Program Clerk (Manages documentation, version control, and technical records.)
- Administrator (Handles administrative tasks, freeing technical staff to focus on development.)
- Tool-smith (Builds and maintains development tools and environments.)
- "Language lawyer" (Deep expert in the programming language(s) used—consulted for tricky syntax or semantics.)

- Issues: Difficult to achieve, Ego issues: superstar and/or team
- Can be appropriate for creative projects or tactical execution

30

NOTES

1. Why would you build a team this way? Some people are far more productive, so the idea is to free your best person from everything except the hardest code, like the **surgeon** in an operating room.
2. If the chief is ill or quits, everything depends on them. The **backup** exists to remove that risk, and also gives the chief someone able to catch their mistakes.
3. Each role removes one kind of distraction from the chief: paperwork, admin, tools, language questions. You don't need to memorize details beyond the fact that they're **supporting roles**, though the list is fair for a listing question.
4. Real superstars are rare, and those around them may resent the imbalance. So despite the appeal, it's **difficult to achieve**. It can suit creative projects or tactical execution.

TIPS & TRICKS

- **BAT CLiP**: Backup programmer, Administrator, Tool-smith, Chief programmer, Language lawyer, Program clerk.

EXAM ANGLE

Prediction Q1

In a chief-programmer team, who is equally skilled and ready to step in for the chief?

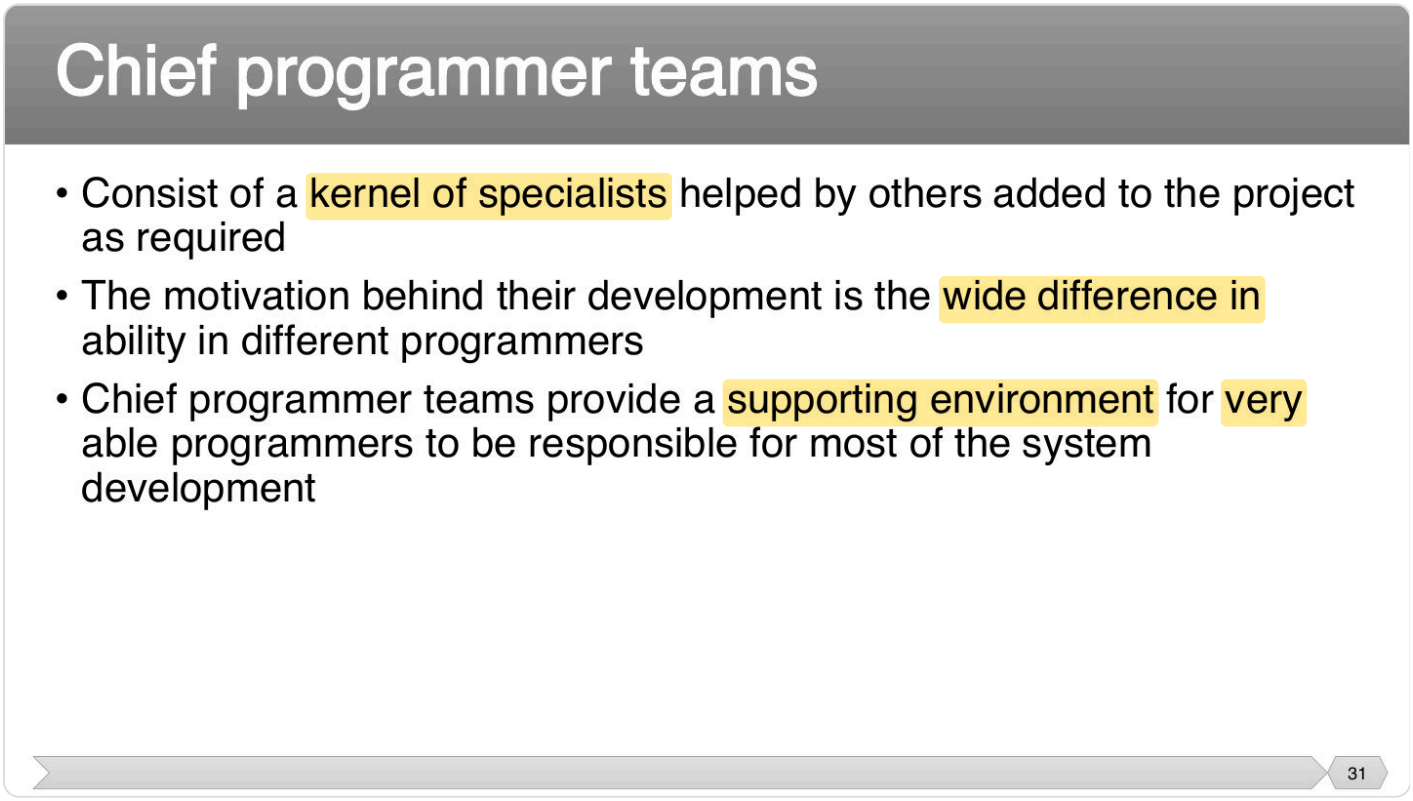
A The language lawyer

B The tool-smith

C The program clerk

D **The backup programmer (co-pilot)**

SLIDE 31 / 77 **Chief programmer teams**




NOTES

1. Think of a core group that stays, with others joining for particular parts. That **kernel** keeps the design consistent, as a few minds decide the system's shape.
2. The motivation: the **wide difference in ability** between programmers. If the best are many times more productive, build the team around them.
3. If someone can do the work of several people, the efficient choice is to remove obstacles from them. That's the logic of a **supporting environment**: everyone else exists to help them produce.

EXAM ANGLE

Prediction Q1

Why were chief-programmer teams developed?

A To remove all specialists

> It relies on specialists.

B **Programmers differ widely in ability**

> Build the team around your best people.

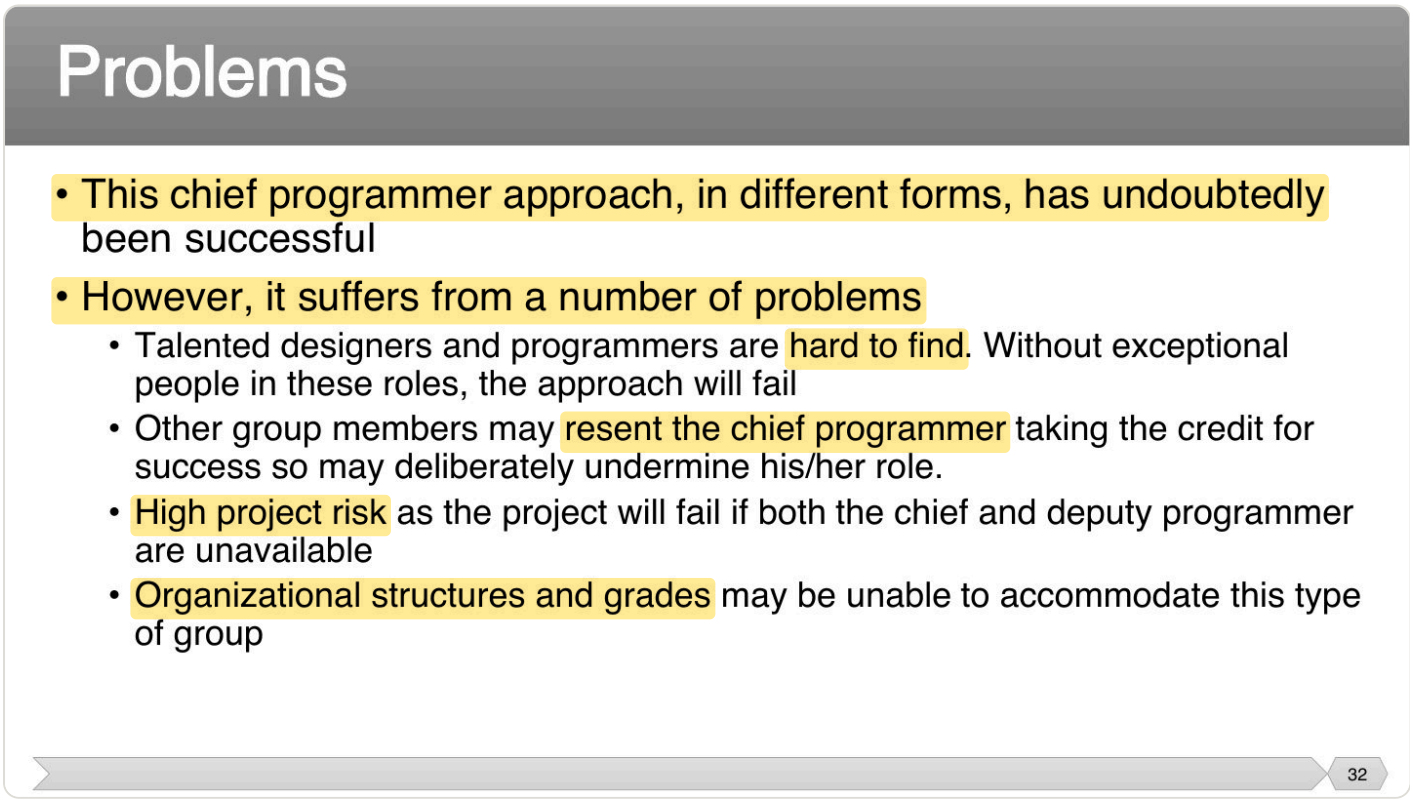
C All programmers perform equally

> That's the opposite.

D To make everyone a manager

> That isn't the aim.

SLIDE 32 / 77 **Problems**




NOTES

1 The whole model assumes you have a superstar. An average lead in this role becomes a bottleneck, so if exceptional people are **hard to find**, it fails.

2. If one person gets the praise for everyone's work, the rest may stop caring or even work against them. That's why **resent** is on this list: the human side can sink the team.
3. So much depends on two people that losing both, even temporarily, stops the project. This is **key-person risk**: the same thing a skills matrix would warn you about.
4. The organization's **structures and pay grades** may not accommodate a team where one person is far above the rest.

EXAM ANGLE

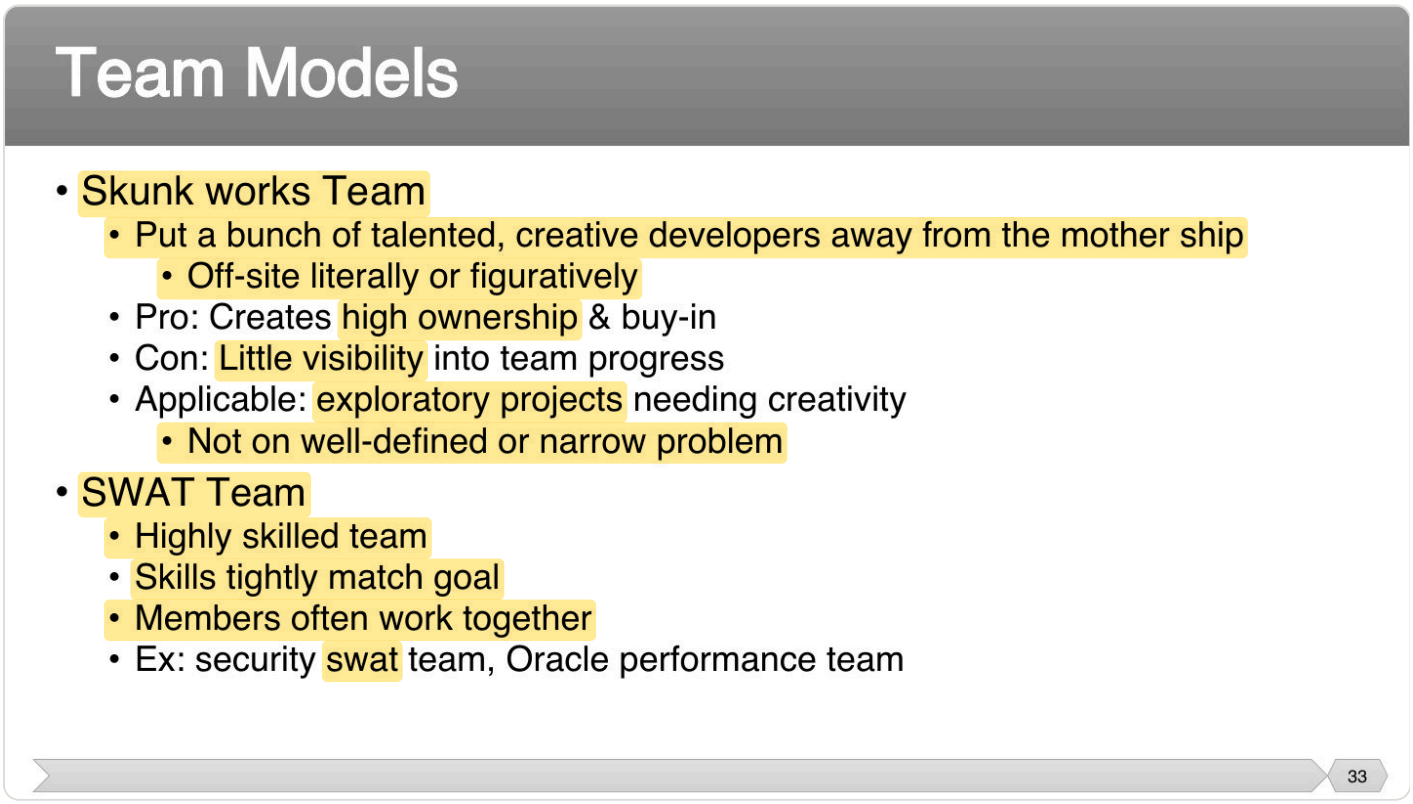
Prediction Q1

A chief-programmer team carries high project risk because the project fails if both the chief and the deputy are unavailable.

T **True**

F False

SLIDE 33 / 77 **Team Models**




NOTES

1. Distance from the main organization removes interruptions and rules, and the team feels the result is really theirs. That's the appeal of **skunk works**, and the reason for **high ownership**.
2. If they're off on their own, nobody can easily tell how it's going. Fine for exploring a new idea, but a narrow, well-defined job doesn't need this freedom, and **little visibility** would just be a risk.

3 Compare the two: skunk works is about freedom to explore, while **SWAT** is about precise skills for a known goal. They already work well together, so they can be dropped into a problem and start immediately.

EXAM ANGLE

Prediction Q1

A team of database experts is brought in to fix Oracle performance problems. This is a:

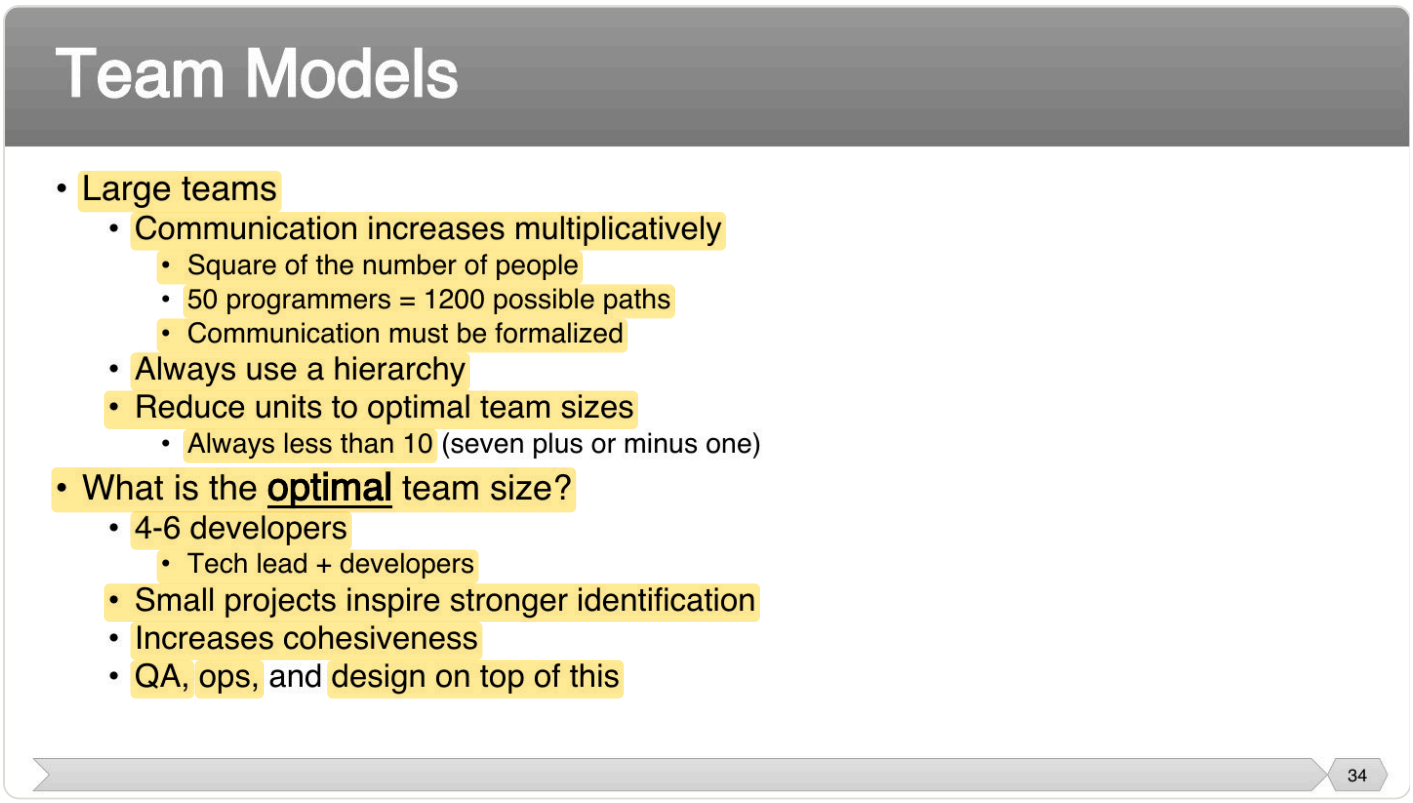
- **A SWAT team**
- B Skunk works team
- C Chief-programmer team
- D Democratic team

Prediction Q2

A company sends a few creative developers off-site to explore a new product idea with little reporting. This is a:

- A SWAT team
- B Business team
- C Chief-programmer team
- **D Skunk works team**

 **Team Models**




## NOTES

1. In **large teams** communication paths grow with roughly the square of the team size. The exact formula is **n(n – 1) / 2**: 50 people → 1,225 paths (the slide rounds to 1,200); 7 people → 21.
2. With that many paths, informal chat can't keep everybody informed. A **hierarchy** reduces the paths, since people talk within small groups and leads talk to each other.
3. Why so small? In a small team, everyone knows what's going on and feels their work matters, and that's what builds **cohesiveness**. The other roles sit around this core so it stays small.

## EXAM ANGLE

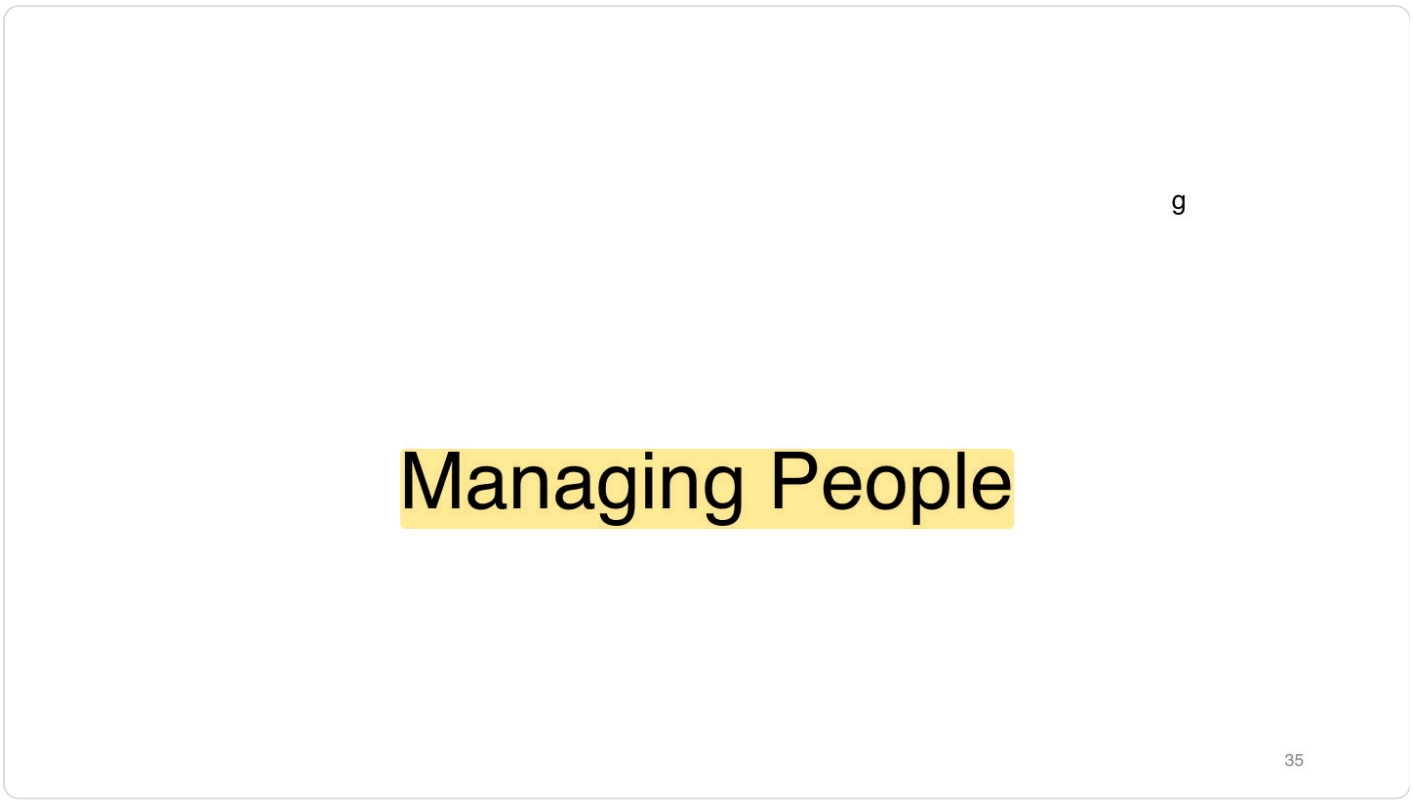
**Past exam · Final 242 · Q3.2** Q1

Calculate the number of communication channels for a project team of 7 members.
**Answer:** n(n – 1) / 2 = 7 × 6 / 2 = 21 channels.

**Prediction** Q2

Your team grows from 5 to 10 people. By how much do the communication channels increase?
**Answer:** From 5 × 4 / 2 = 10 to 10 × 9 / 2 = 45 channels: the team doubled but channels grew 4.5 times. That is why large teams need hierarchy and should be split into sub-teams under 10.

## SLIDE 35 / 77 Managing People

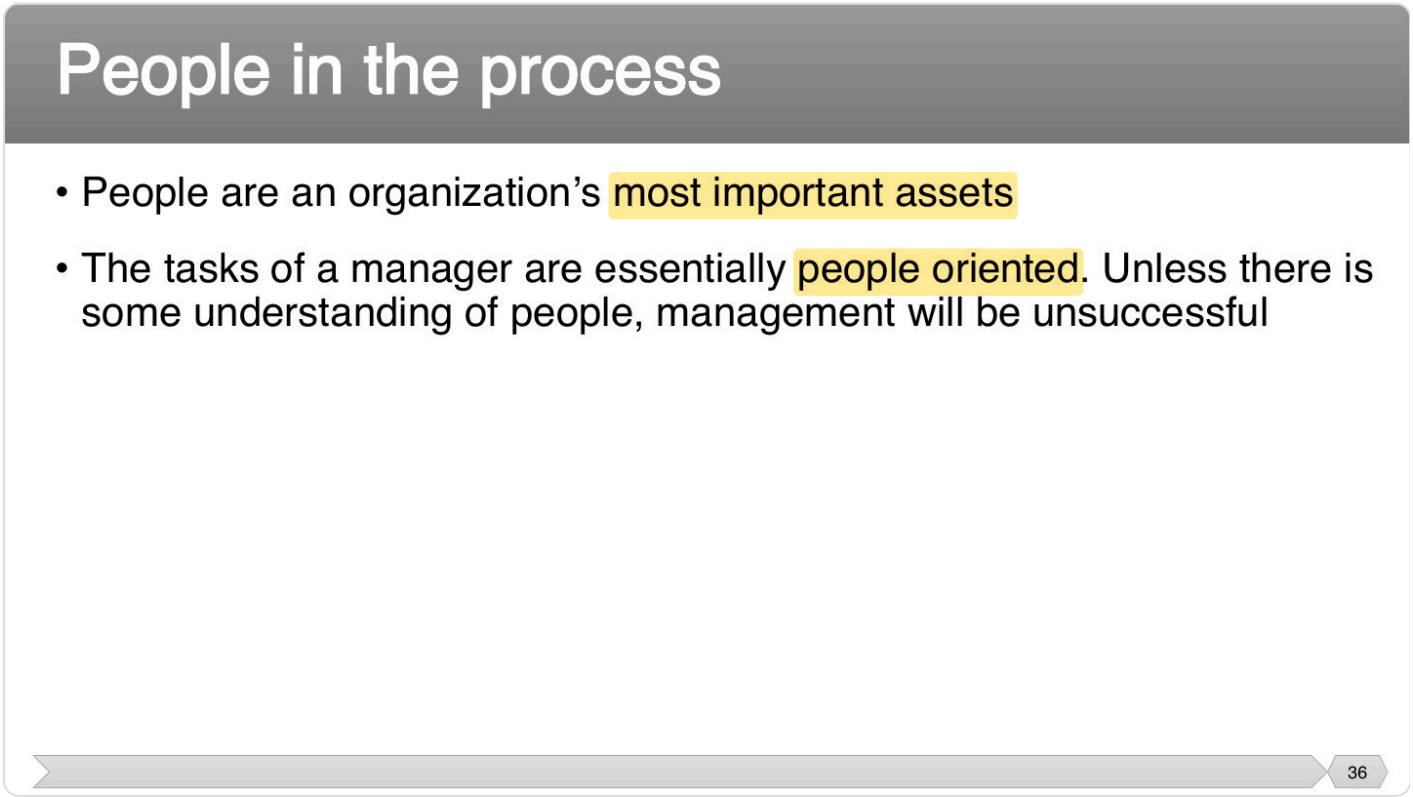



NOTES

1 Section divider: **Managing People**.

---

## SLIDE 36 / 77 People in the process



## NOTES

1. Equipment can be replaced, but people carry the knowledge and make the decisions. Calling them the **most important assets** means a PM should plan for people as carefully as for tasks.
2. A manager's tasks are essentially **people-oriented**; without understanding people, management fails however good the technical plan is.

## EXAM ANGLE

Prediction

Q1

Why does a manager need to understand people?

A Only HR deals with people

> The slide says the manager's tasks are people-oriented.

B People never affect the plan

> They're the ones doing it.

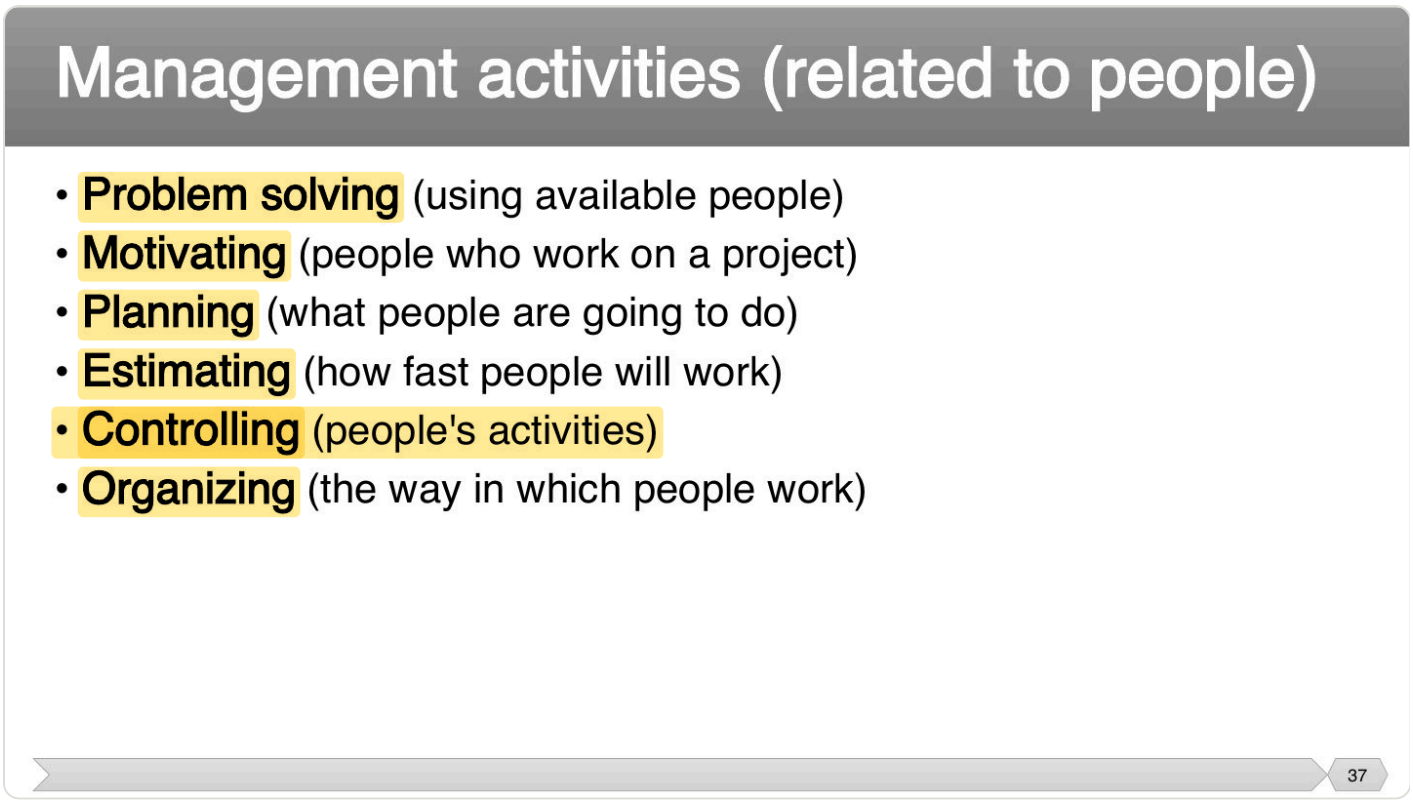
C Tools matter more than people

> People are called the most important assets.

**D A manager's tasks are essentially people-oriented, so management fails otherwise**

> Technical plans don't carry themselves out.

 **Management activities (related to people)**




NOTES

1. Notice that “using available people” is a real limit: you rarely get to choose your ideal team. And **motivating** comes up here because a plan only happens if people want to do it.
2. Plans and estimates can look purely technical, but “how fast” depends on who’s doing it. The same task can take one developer a day and another a week, so **estimating** is about people too.
3. **Controlling** doesn’t mean micromanaging; it means knowing whether the work is on track. **Organizing** is about who works with whom and how, which shapes how well they do.

TIPS & TRICKS

- Sentence, in slide order: **P**roject **M**anagers **P**lan, **E**stimate, **C**ontrol, **O**rganize. Problem solving, Motivating, Planning, Estimating, Controlling, Organizing.

EXAM ANGLE

Prediction Q1

"How fast will people work?" belongs to which management activity?

A Organizing

> That's about the way they work.

B Motivating

> That's about getting them to want to.

C Controlling

> That's about people's activities.

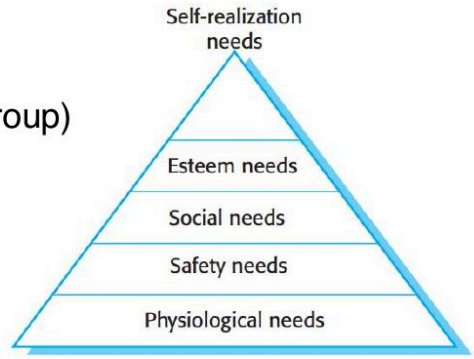
D **Estimating**

> The slide defines it that way.

SLIDE 38 / 77 **Motivation**

# Motivation

- An important role of a manager is to motivate the people working on a project (motivation is needs-based)
- Motivation is a complex issue but it there are different types of motivation based on
  - Basic needs (e.g. food, sleep, etc.)
  - Personal needs (e.g. respect, self-esteem)
  - Social needs (e.g. to be accepted as part of a group)
- Maslow's hierarchy of needs




38

NOTES

1 **Needs-based** means that to motivate someone, you first ask what they're lacking. The same reward can motivate one person and do nothing for another whose need is different.

2. Think of the groups as moving outward: staying alive, feeling valued, then belonging. A manager has most influence on **personal** and **social** needs, since those arise at work.

3. **Maslow's hierarchy**: physiological → safety → social → esteem → self-realization (self-actualization).

TIPS & TRICKS

- Maslow from the bottom up, where the order matters: **P**lease **S**tudents, **S**tudy **E**very **S**ession. Physiological, Safety, Social, Esteem, Self-realization.

## EXAM ANGLE

Prediction Q1

A teammate wants to be accepted as part of a group. Which type of need is this?

A Technical

> That isn't a type on the slide.

B Basic

> That's food and sleep.

**C Social**

> Being accepted is the slide's example.

D Personal

> That's respect and self-esteem.

Prediction Q2

Motivation is based on needs.

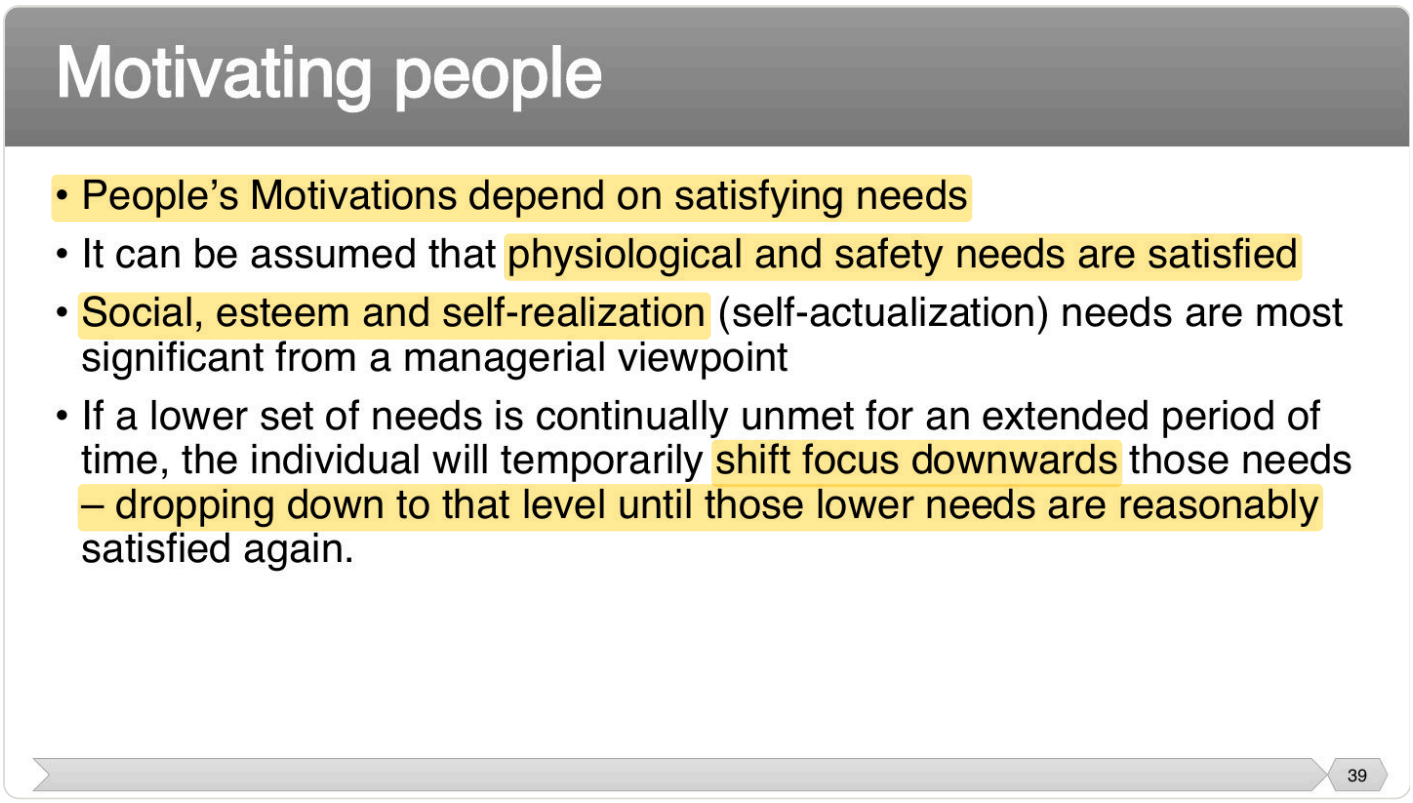
**T True**

> People act to satisfy needs.

F False

> The slide says it's needs-based.

 **Motivating people**




NOTES

1. A paying job with a safe office usually covers the basics. So a manager can start higher up, which is why **physiological and safety needs** aren't the usual focus.
2. Once the basics are covered, what makes people try harder is belonging, feeling respected and growing. These are the needs a manager can really act on, so **social, esteem and self-realization** matter most.
3. But if a lower need stays unmet for a long time (e.g. job security during layoffs), people **shift focus downwards** to it until it is satisfied again, and higher motivators stop working.

EXAM ANGLE

Prediction Q1

During layoffs, recognition stops motivating someone who fears for their job. Why?

A Recognition is never useful

> It normally works.

B Needs don’t affect work

> They do.

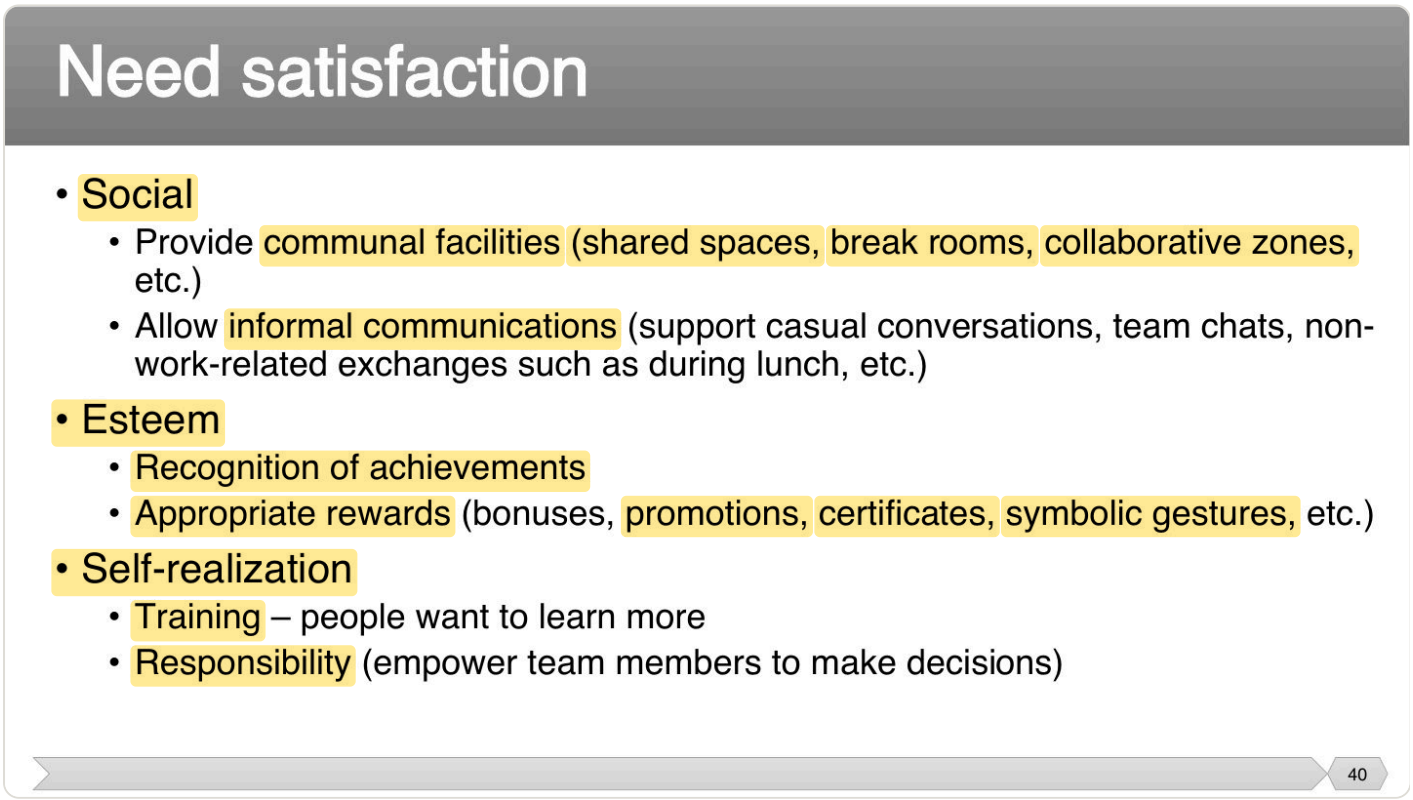
C Self-realization always comes first

> Lower needs come first.

**D An unmet lower need draws their focus down until it’s satisfied**

> That’s what the slide describes.

SLIDE 40 / 77 **Need satisfaction**




NOTES

1 People work better with colleagues they know. Break rooms and casual chats look like wasted time, but they’re how **social** needs get met and how information spreads.

2. People want to feel their effort was noticed. A specific “thank you” in front of the team can matter as much as a bonus, so **recognition** doesn’t have to be expensive.

3. Here people want to grow and be trusted. So **training** and real **responsibility** motivate strong staff, while doing the same simple task for years drives them away.

## EXAM ANGLE

Past exam · Final 251 · Q5.3 — Q1

Your team has just completed a major milestone in the project development process. How would you celebrate this achievement collaboratively with your team members? How would you recognize individual contributions to the team’s success? What actions would you take to ensure that team morale remains high as you move forward in the project?
**Answer:** Celebrate as a team (social need), e.g. a team lunch and an announcement of the milestone to the stakeholders. Recognize individuals specifically and publicly (esteem): name what each person contributed in the meeting, send thank-you notes to their managers, and give certificates or small rewards. Keep morale high (self-realization): hold a retrospective that captures what went well, set clear and achievable next goals, balance the workload, and offer training and more responsibility to those who want to grow.

Prediction — Q2

Sending a developer on a cloud-architecture course mainly satisfies which need?

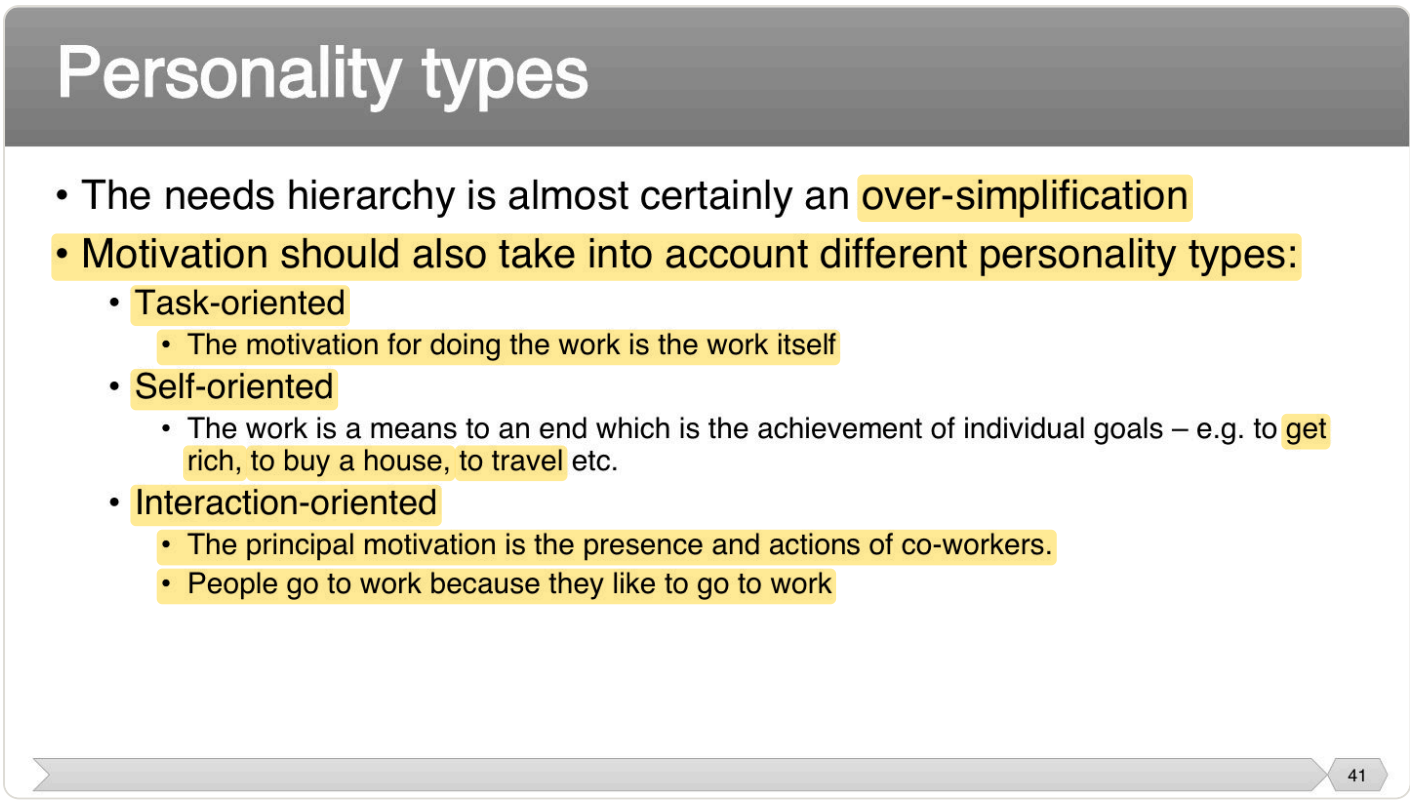
A Safety

B **Self-realization**

C Physiological

D Social

 **Personality types**




NOTES

1. Maslow suggests everybody wants the same things in the same order, and real people aren't like that. Two teammates at the same level can want very different things, hence **personality type**.
2. **Task-oriented**: the motivation is the work itself (they love solving the problem).
3. The same project might be a means to a promotion or to buying a house. **Self-oriented** isn't a criticism, but it tells you these people respond to personal benefit.
4. For them, an isolated task with no contact is demotivating, even if it's interesting. **Interaction-oriented** people keep a team talking, and that has real value.

TIPS & TRICKS

- **SIT**: Self-oriented, Interaction-oriented, Task-oriented.

EXAM ANGLE

Prediction Q1

A developer works mainly to save for a house. This personality type is:

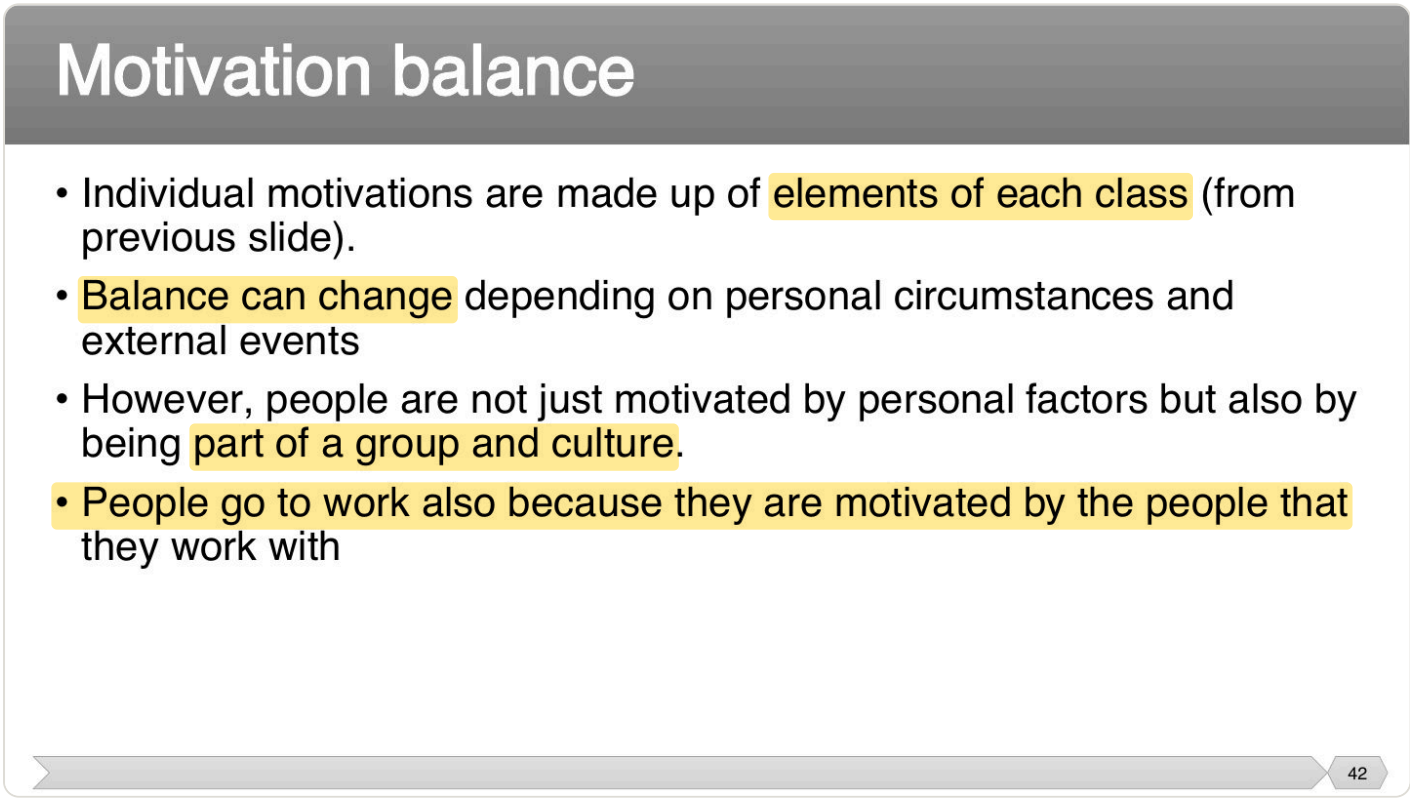
A Interaction-oriented

B Democratic

C Task-oriented

D **Self-oriented**

SLIDE 42 / 77 **Motivation balance**




NOTES

1. Real people have **elements of each class**, and the balance changes with personal circumstances (a new baby makes someone more self-oriented for a while).
2. Even someone working mainly for money may stay because they like the team. So building a good **group and culture** is a motivation tool that doesn't cost much.

EXAM ANGLE

Prediction Q1

A new parent becomes more focused on income for a while. What does this show?

**A** Only salary motivates people

> Group and culture matter too.

**B** People never change types

> The slide says the balance changes.

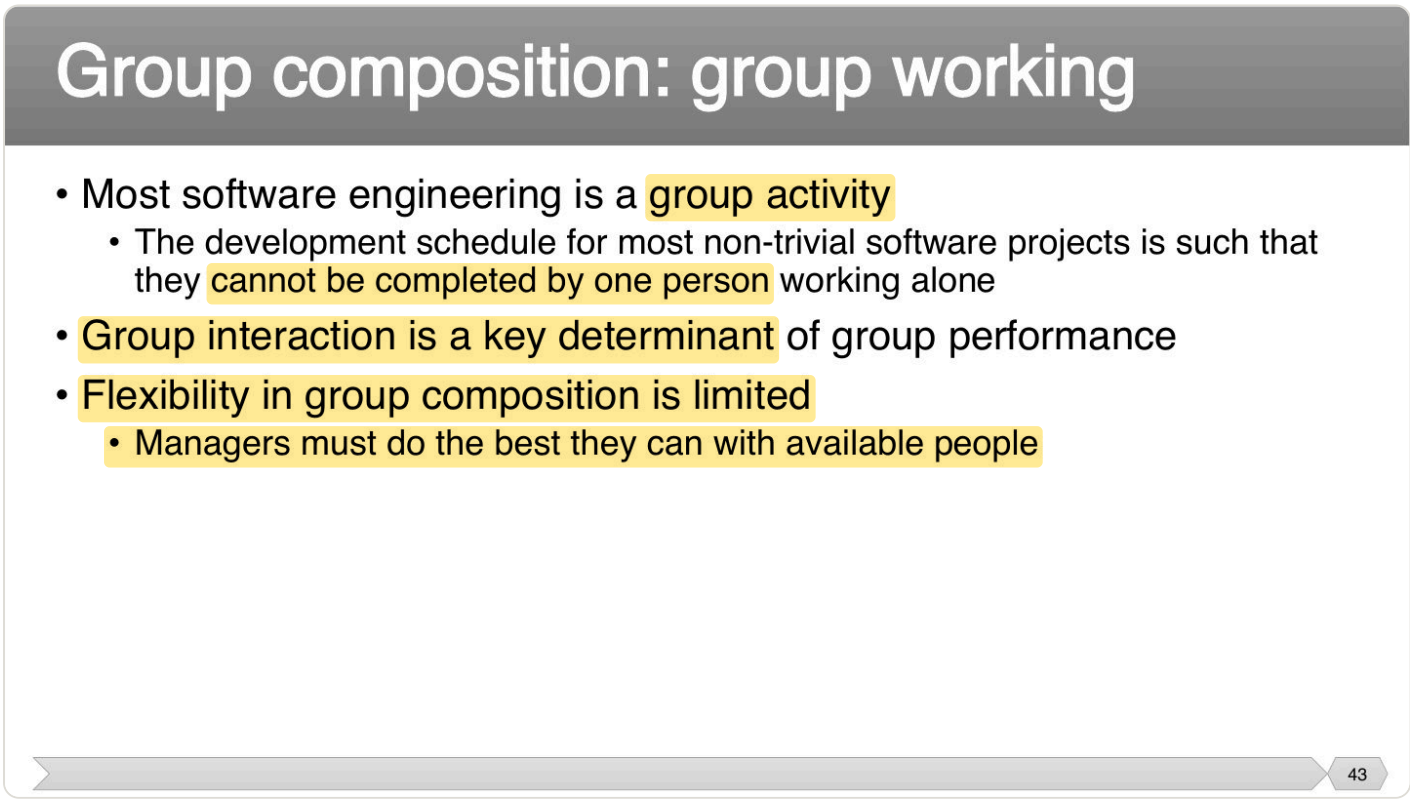
**C** They're now interaction-oriented

> Income focus is self-oriented.

**D** **The balance of motivations changes with personal circumstances**

> People aren't fixed in one type.

SLIDE 43 / 77 **Group composition: group working**




NOTES

1 Real software is too big for one person, however talented. Working out how people fit together is therefore central, because this is a **group activity** by necessity.

2 **Group interaction** is a key determinant of performance, so how the group works together matters as much as individual skill.

3 You rarely get to hand-pick a dream team, so plan around who you actually have. That's why **group composition** is listed as limited: adapt the work to people's strengths.

EXAM ANGLE

Prediction Q1

Why can't a manager just choose the ideal group?

A Rules ban choosing people

> That isn't the reason.

B Groups aren't needed in software

> Most software is a group activity.

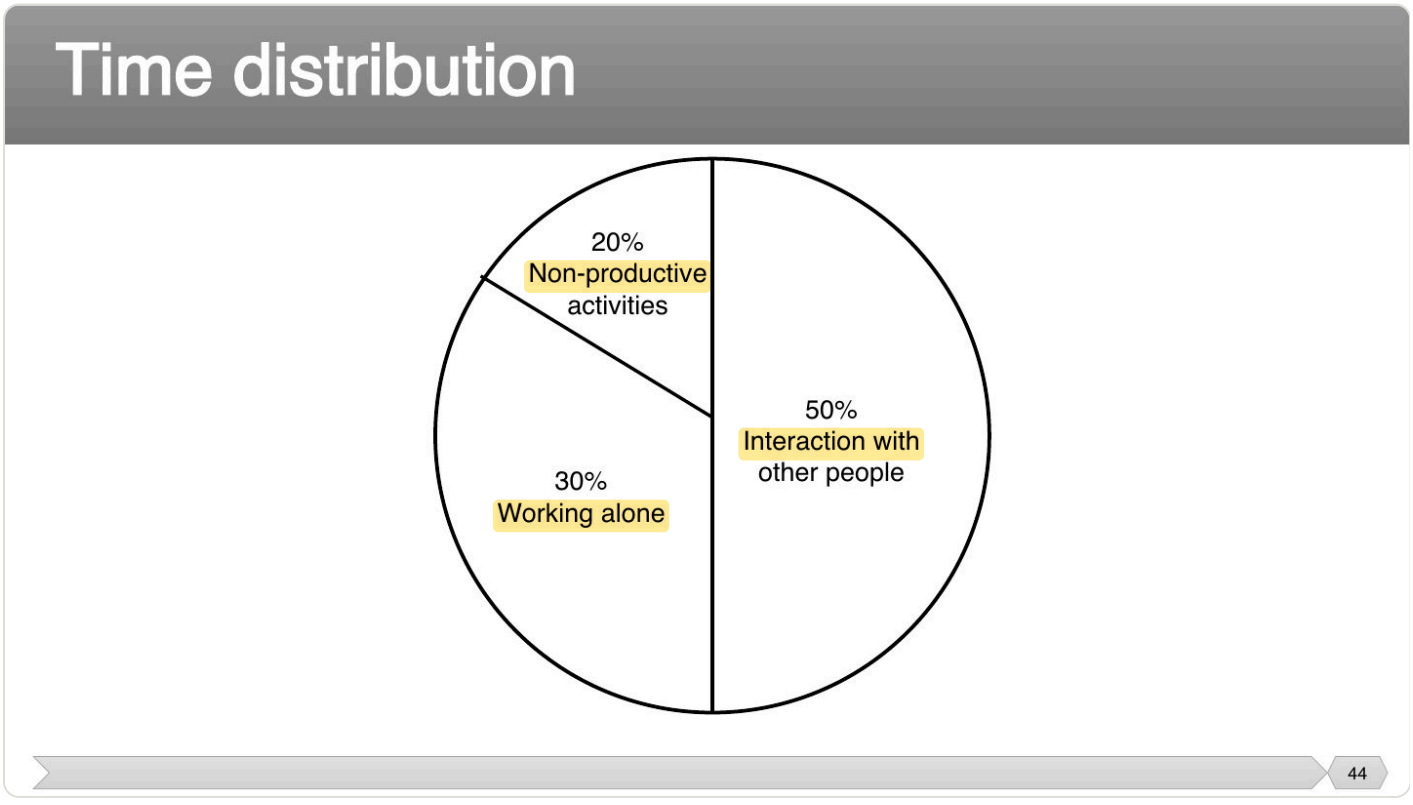
C Interaction doesn't affect performance

> It's a key determinant.

D **Flexibility in group composition is limited**

> They must use the people available.

SLIDE 44 / 77 **Time distribution**



NOTES

1. About **50%** of an engineer's time is spent interacting with other people (meetings, reviews, discussions), so communication skills matter far more than people expect.
2. Don't plan a person's time as if they code all day. With only **30%** alone, an estimate that assumes eight hours of solo work will be far too optimistic.

TIPS & TRICKS

- **WIN**: Working alone, Interaction with others, Non-productive.

EXAM ANGLE

Prediction Q1

According to the time-distribution chart, roughly how much of an engineer's time is spent interacting with other people?

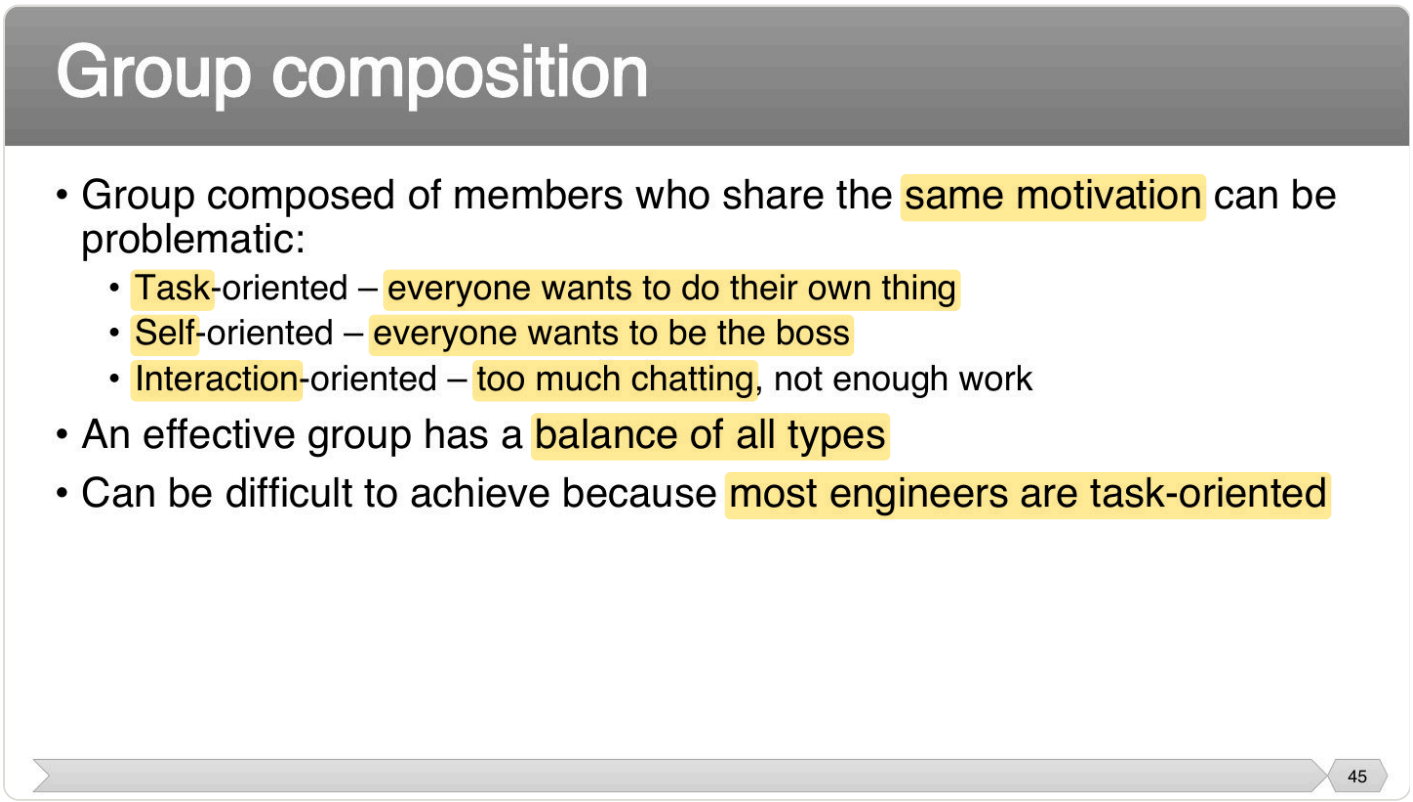
A 80%

B 20%

C 30%

D **50%**

SLIDE 45 / 77 **Group composition**



NOTES

1. Mixing types is deliberate. People with identical motivations tend to clash or leave gaps, just like a team of only goalkeepers, which is why having the **same motivation** is a problem.
2. Each extreme has a weakness on its own: nobody coordinates, everybody competes, or nobody finishes. That's why you look at the mix, not only at individuals: **task**, **self**, **interaction**.
3. Since most engineers enjoy the work itself, you'll often lack people who keep communication going. So value the **interaction-oriented** member, even if they write less code.

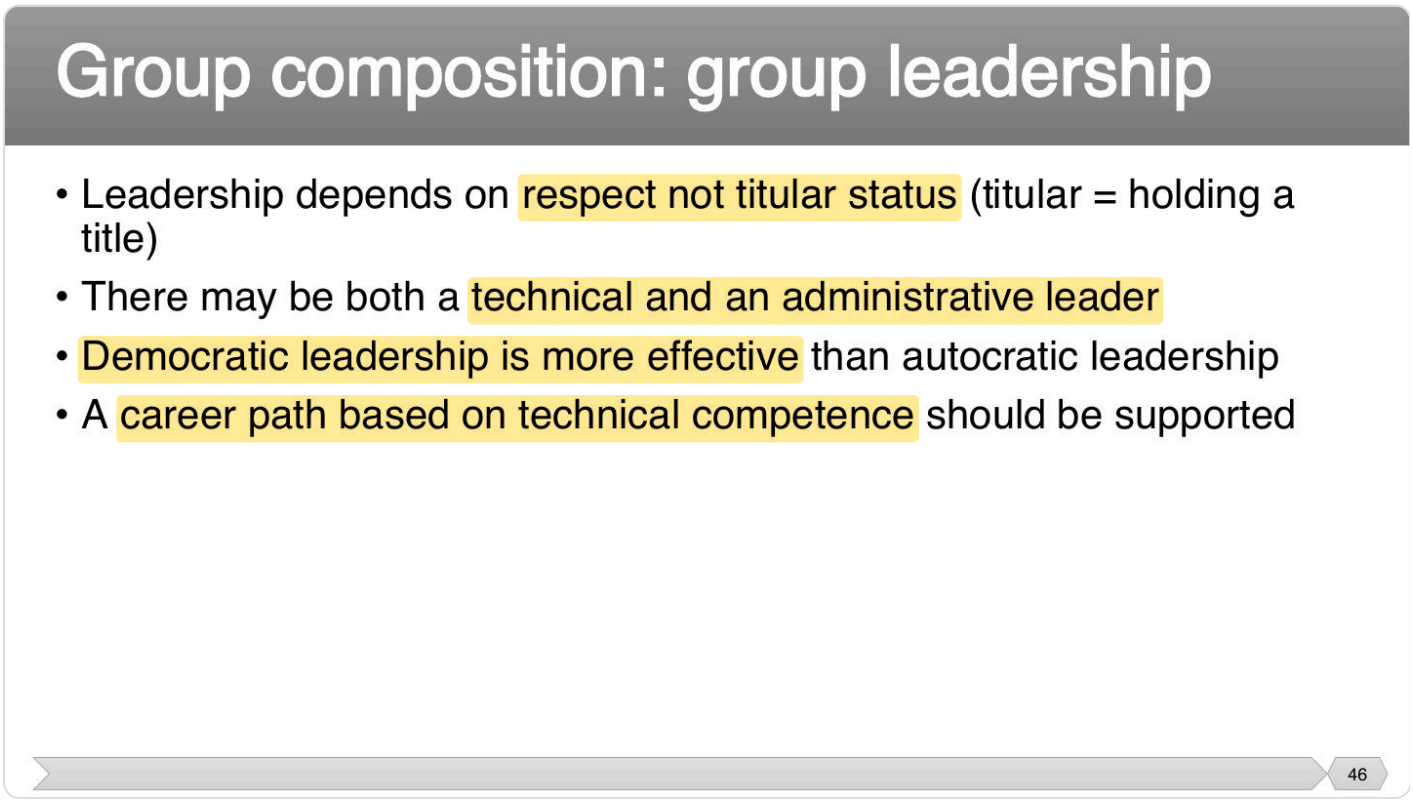
EXAM ANGLE

Prediction Q1

A team where "everyone wants to be the boss" is dominated by which personality type?

- A Interaction-oriented
- B Task-oriented
- C **Self-oriented**
- D Balanced

SLIDE 46 / 77 **Group composition: group leadership**



NOTES

1. A manager with a title can be ignored in practice if nobody respects them, while a respected senior developer gets followed. So **respect** is what actually makes a leader.
2. Good engineers aren't always good at budgets and reports, and vice versa. Splitting the role into **technical** and **administrative** leaders lets each person do what they're good at.
3. Skilled people often know more than their manager about the problem. Letting them contribute to decisions gets better answers and more commitment. That's why **democratic leadership** works in software.
4. Support a **technical career path** so good engineers don't have to become managers to progress.

## EXAM ANGLE

Prediction

Q1

Why might a group have both a technical and an administrative leader?

A Two bosses are always required by law

> Nothing says so.

**B Different skills are involved, so leadership can be split**

> The slide allows for both.

C Leadership depends only on title

> It depends on respect.

D Technical people can’t be leaders

> The slide supports technical careers.

Prediction

Q2

What does leadership depend on, according to the slide?

A Title, not respect

> That’s the reverse.

B Seniority alone

> Respect is what’s named.

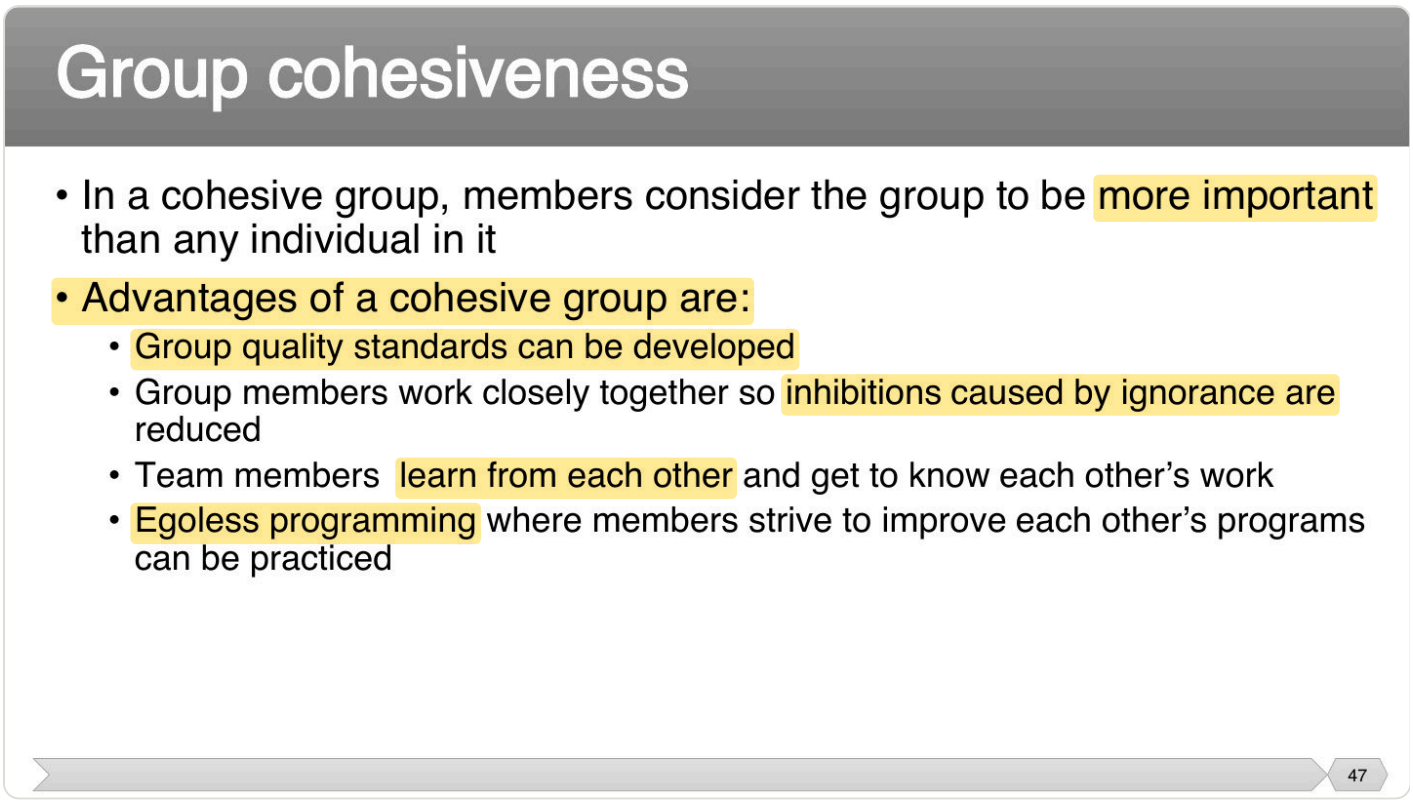
**C Respect, not title**

> That’s the first point.

D Pay grade

> That isn’t named.

 **Group cohesiveness**




NOTES

1. When people care about the group, they help each other and protect its standards, rather than only their own task. That's what **cohesive** means in practice.

2. Standards the group sets for itself get followed, while rules imposed from outside are often dodged. Working closely also means people quickly learn what they don't know, rather than hiding it, which is how **ignorance** shrinks.

3. If only one person understands a module, you have a risk. When people **learn from each other**, the team can cope with someone being away.

4. **Egoless programming**: members try to improve each other's programs and don't treat code as personal property.

EXAM ANGLE

Past exam · Midterm 251 · V1 · MCQ 10 (also V2 MCQ 9) Q1

In a cohesive team, which of the following is an advantage?

A Members avoid challenging each other's work to reduce conflict.

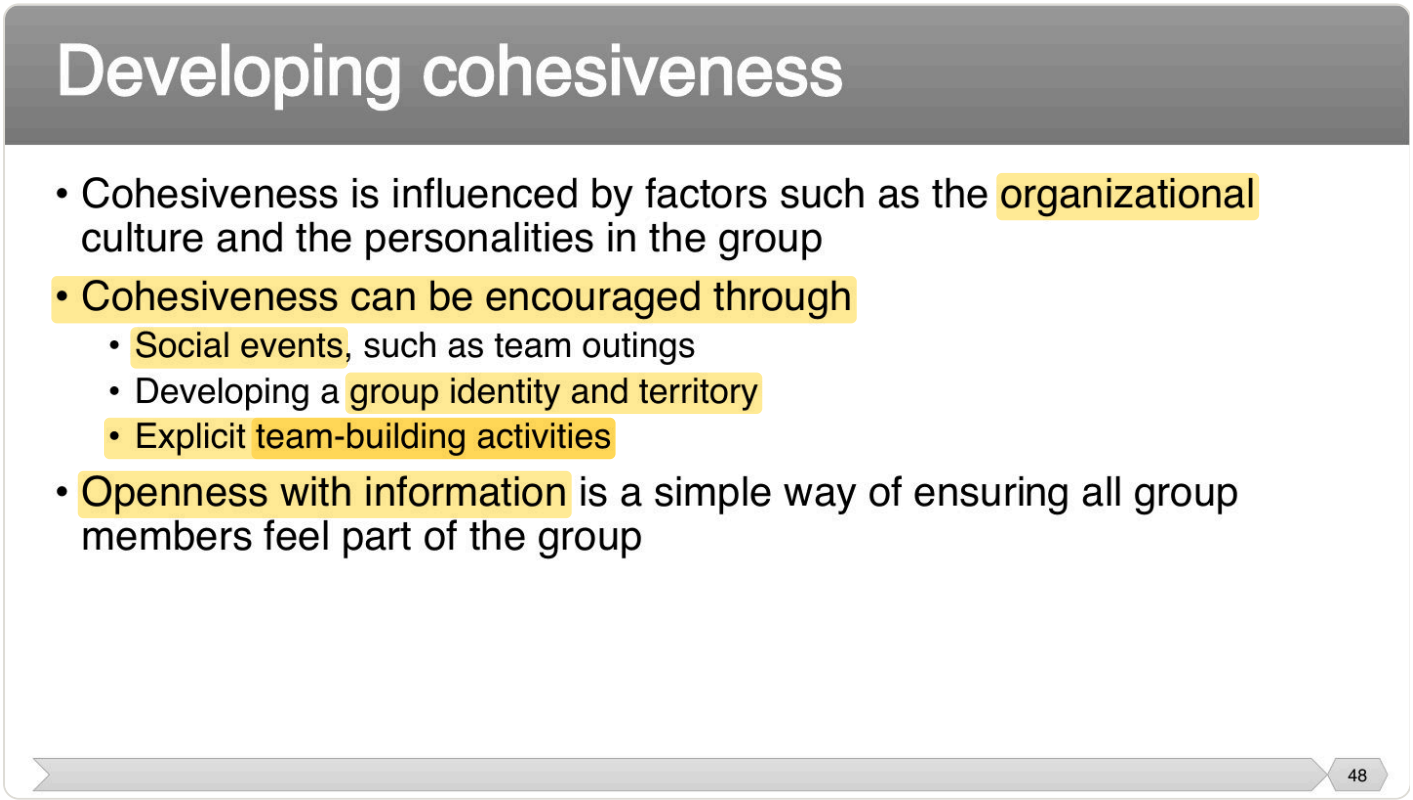
B **Group quality standards can be developed and maintained collectively.**

> Cohesive groups can set and keep shared quality standards, learn from each other and practise egoless programming. Avoiding challenge is groupthink, a risk of cohesion.

C Each member works independently without needing oversight.

D Decision-making is always faster due to consensus.

SLIDE 48 / 77 **Developing cohesiveness**




NOTES

1. You can't simply order people to bond. A blaming **organizational culture** works against it, and some personalities need more time, so treat it as something to encourage.
2. A shared name or space sounds trivial, but it gives people something to belong to. **Group identity** is what turns individuals into "us".
3. **Openness with information** is the simplest tool: when everyone knows what is going on, everyone feels part of the group.

EXAM ANGLE

Prediction Q1

Which is the simplest way to help everyone feel part of the group?

A Having only formal meetings

> That's not what it names.

**B Being open with information**

> The slide calls this a simple way.

C Reducing team size to one

> That removes the group.

D Keeping plans secret

> That does the opposite.

Prediction Q2

A team picks a name and has its own space. What's being developed?

A Groupthink

> Identity alone isn't groupthink.

B A cost baseline

> That's about money.

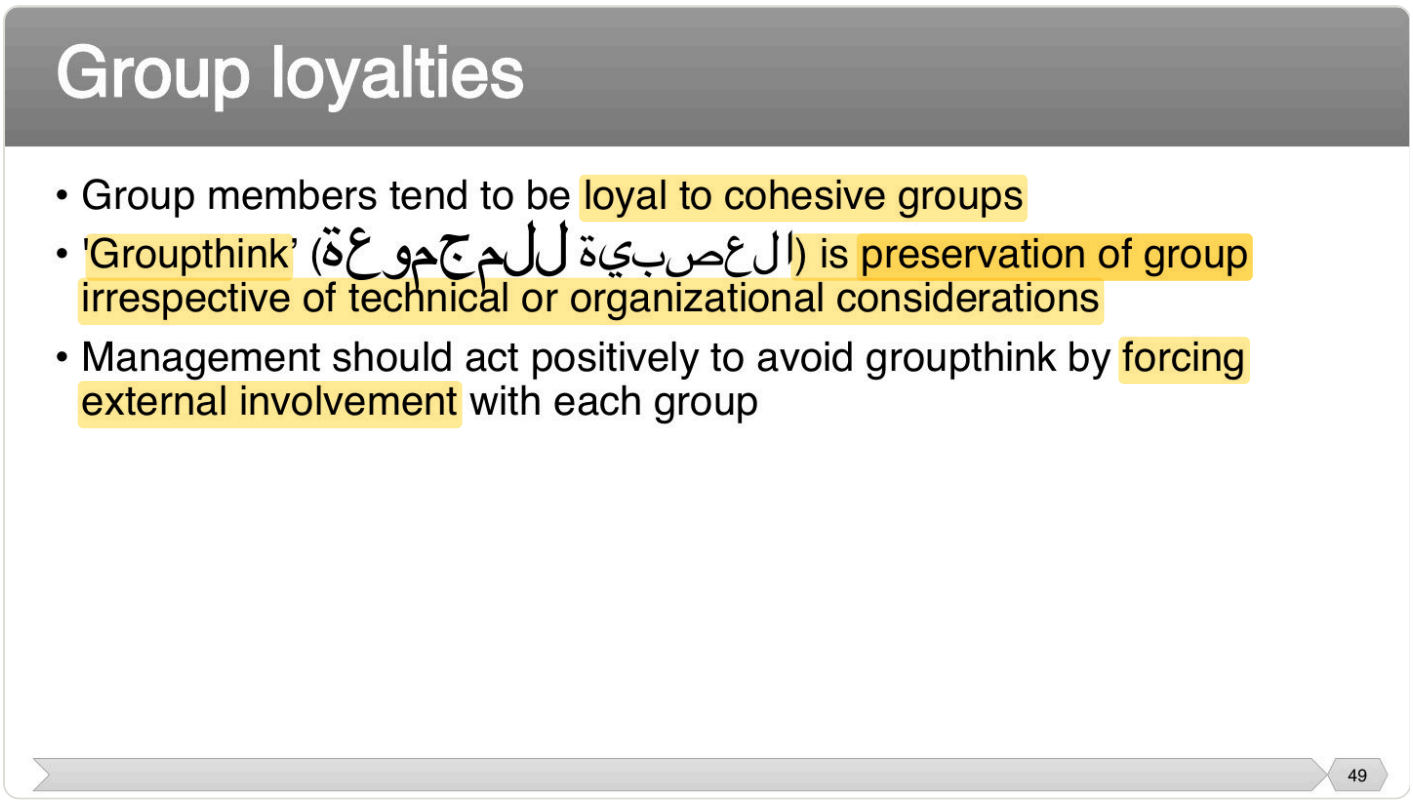
**C A group identity and territory**

> It's a way of encouraging cohesiveness.

D A skills matrix

> That's a planning tool.

 **Group loyalties**




NOTES

1. A close team defends its members, which helps morale. But **loyalty** can also stop people from speaking up when the group is wrong.
2. **Groupthink**: preserving the group regardless of technical or organizational considerations (defending a bad design because the group made it).
3. A team can't easily see its own blind spots, so bring in someone who isn't invested. That's what **external involvement** does: outsiders can ask the awkward questions.

EXAM ANGLE

Prediction Q1

How should management counter groupthink?

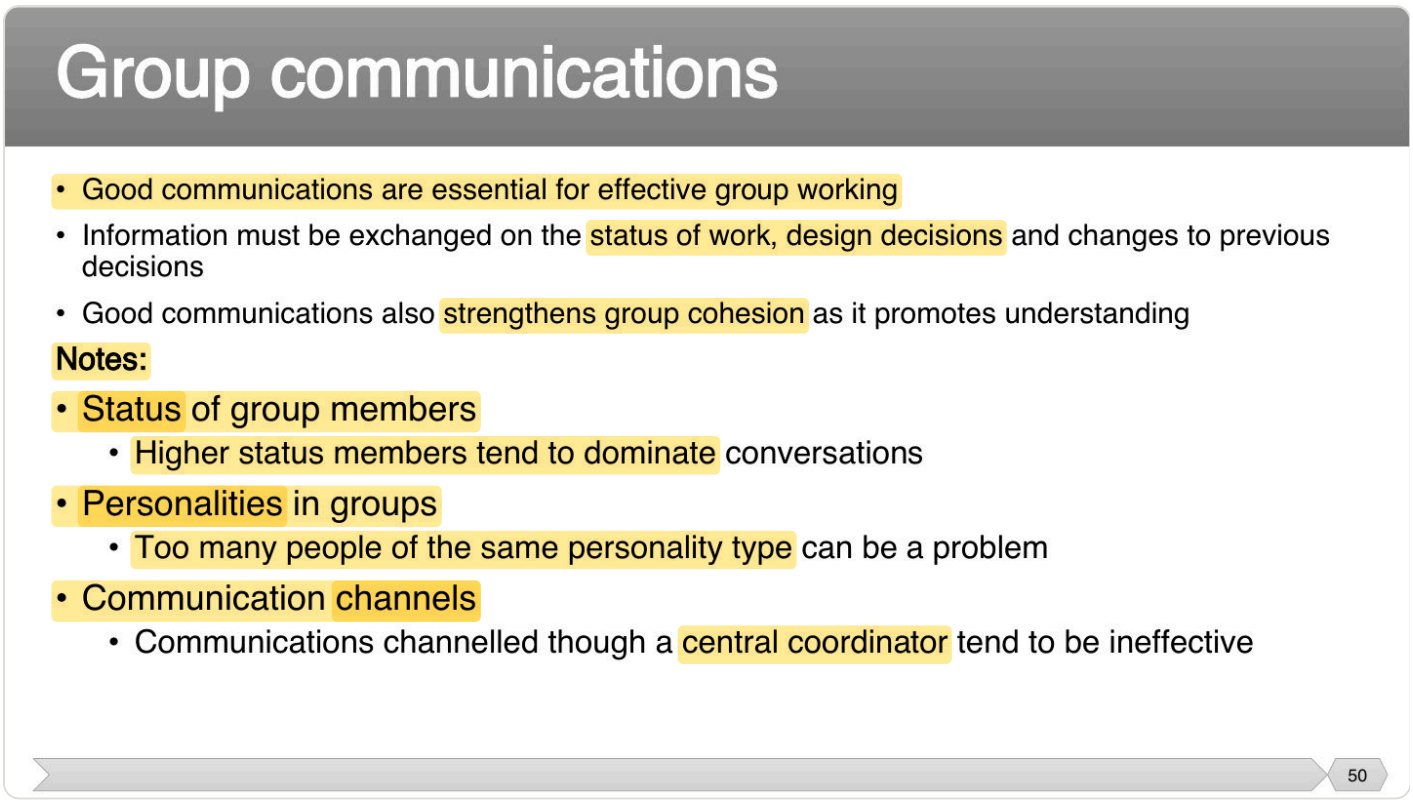
A Reward unanimous agreement

B **Force external involvement with the group**

C Isolate the group further

D Remove all conflict

 **Group communications**




NOTES

1. If one developer changes a design and nobody hears about it, others build on something that no longer exists. So **status** and **design decisions** must be shared, and sharing them also makes people feel included.

2. Juniors may have the answer but stay quiet. A good PM deliberately asks them first, so the **status** of a senior person doesn't shut the discussion down.

3. If everyone is quiet, nothing gets raised, and if everyone is dominant, nobody listens. A mix of **personalities** helps conversations work, the same point as slide 45.

4. If everything must pass through one person, they become a traffic jam, and messages get distorted along the way. Direct **channels** are usually better.

EXAM ANGLE

Prediction

Q1

Juniors stay quiet while a senior person dominates a discussion. Which factor is this?

A Team size only

> The slide names status for this.

B Financial incentives

> They aren't involved.

C Communication channels

> That's about the route information takes.

**D Status of group members**

> Higher-status members tend to dominate.

Prediction

Q2

All questions must go through one coordinator, and answers are slow. What's the problem?

A Too much natural light

> That's unrelated.

**B Communication through a central coordinator tends to be ineffective**

> That person becomes a bottleneck.

C Too many personality types

> That's a different factor.

D The group is too cohesive

> That isn't the cause here.

 **Group organization**

# Group organization

- Software engineering group sizes should be relatively small (< 10 members) optimal is less than seven (see previous slides)
- Break big projects down into multiple smaller projects
- Small teams may be organized in an informal, democratic way
- Chief programmer teams try to make the most effective use of skills and experience



NOTES

1. More people mean many more communication paths, so past a certain size you spend more time coordinating than working. That's why you **break big projects down** instead.
2. With so few people, formal roles and procedures would only slow things down. An **informal, democratic** setup works because people can simply talk to each other...
3. ...while a **chief-programmer** team takes the opposite view: if skills are scarce, arrange the team so your strongest people do the critical work.

EXAM ANGLE

Prediction Q1

What should you do with a big project, according to the slide?

A Remove all leads

> That isn't suggested.

B **Break it down into multiple smaller projects**

> Groups should stay small.

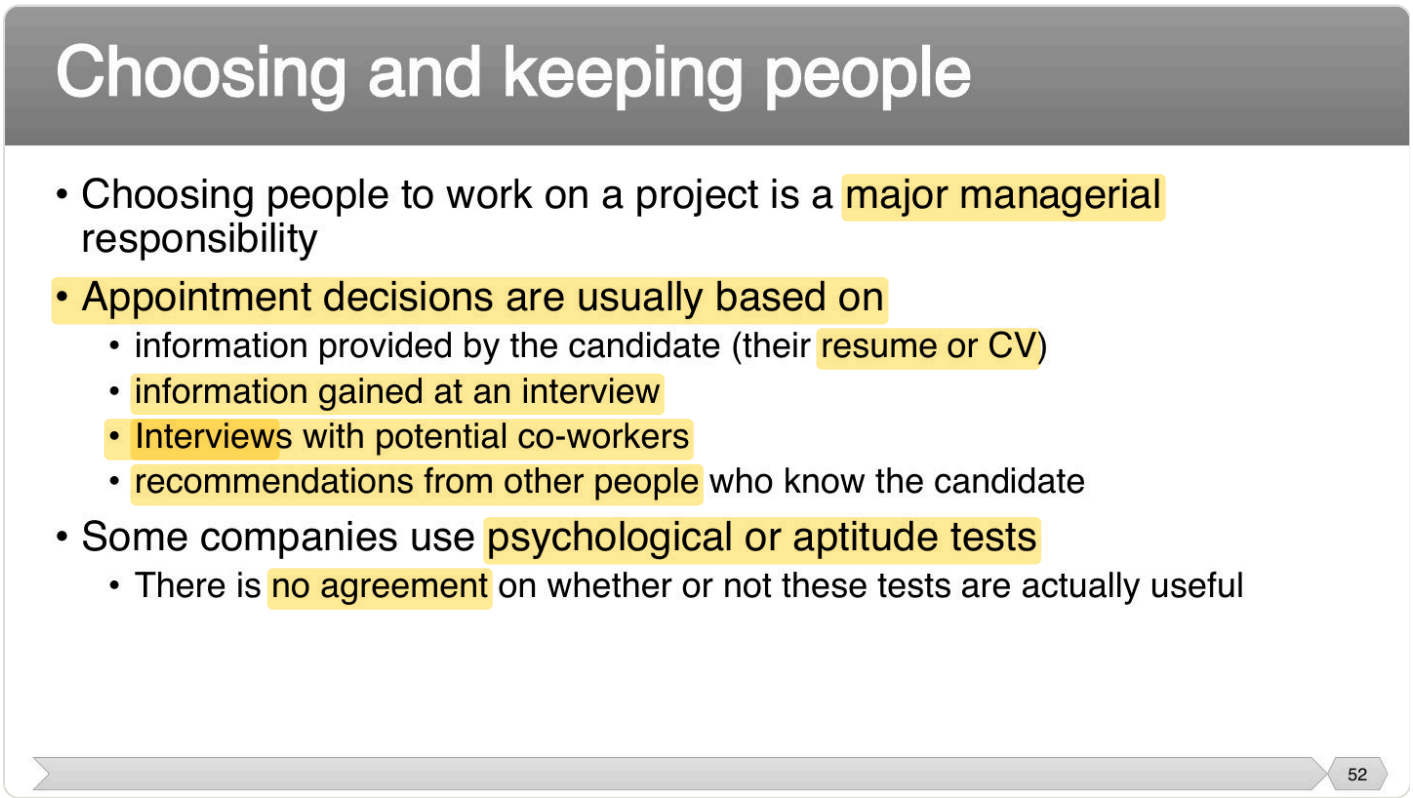
C Avoid communicating

> That can't help.

D Put everyone in one group

> Groups should stay under 10.

SLIDE 52 / 77 **Choosing and keeping people**




NOTES

1 Hiring mistakes are expensive and hard to undo, as a poor fit slows down everybody. Good selection is a **major responsibility** that prevents problems no process can fix later.

2. Each source has a weakness: a **CV** is written to impress, and an **interview** is a short performance. That's why you use several, and why meeting co-workers matters: they're the ones who'll work with this person.

3. Don't assume a test score predicts good work, since experts disagree. If an option says these tests are proven to work, it's wrong: the slide says there's **no agreement**.

---

EXAM ANGLE

Prediction Q1

Psychological tests are proven to predict how well someone will do.

T True

> The slide says this is unresolved.

**F False**

> There's no agreement on whether they're useful.

Prediction Q2

Why is meeting potential co-workers valuable?

**A It's one of the sources of information for appointment decisions**

> It's on the list, along with CVs and recommendations.

B It replaces all other information

> It's one source of several.

C It removes the need for interviews

> It's a kind of interview.

D It's a legal requirement

> That isn't stated.

---

 **Staff selection factors**

# Staff selection factors

| Factor | Explanation |
|---|---|
| Application domain experience | For a project to develop a successful system, the developers must understand the application domain |
| Platform experience | May be significant if low-level programming is involved. Otherwise, not usually a critical attribute. |
| Programming language experience | Normally only significant for short duration projects where there is insufficient time to learn a new language. |
| Educational background | May provide an indicator of the basic fundamentals which the candidate should know and of their ability to learn. This factor becomes increasingly irrelevant as engineers gain experience across a range of projects. |
| Communication ability | Very important because of need for project staff to communicate orally and in writing with other engineers, managers, and customers. |
| Adaptability | Adaptability may be judged by looking at the different types of experience that candidates have had. This is an important attribute as it indicates an ability to learn. |
| Attitude | Project staff should have a positive attitude towards their work and should be willing to learn new skills. This is an important attribute but often very difficult to assess. |
| Personality | Again, an important attribute but difficult to assess. Candidates must be reasonably compatible with other team members. No particular type of personality is more or less suited to software engineering. |



## NOTES

1. A developer who doesn't understand banking can write clean code that does the wrong thing. That's why **application domain** knowledge often matters more than knowing a specific language.
2. Good programmers pick up a new language fairly quickly, so it's only critical when there's no time to learn. **Platform** knowledge counts when you're working close to the hardware. A common trap: it is not the most critical factor.
3. A degree tells you what someone has probably been taught, and how well they can learn. As real work builds up, it says less, which is why **education** counts for less over time.
4. Software is a people job, and half of it is interaction (slide 44). Technology also changes constantly, so **communication** and **adaptability** can outlast any one technical skill.
5. You can't measure these from a CV, so they're hard to judge, but a bad fit can harm a whole team. There's no ideal "programmer **personality**"; what matters is whether they fit this team.

## TIPS & TRICKS

- Sentence, in slide order: **A**ll **P**rogrammers **P**refer **E**asy **C**ode **A**nd **A**wesome **P**izza. Application domain, Platform, Programming language, Educational background, Communication ability, Adaptability, Attitude, Personality.

EXAM ANGLE

Prediction Q1

According to the table, programming language experience is significant mainly when:

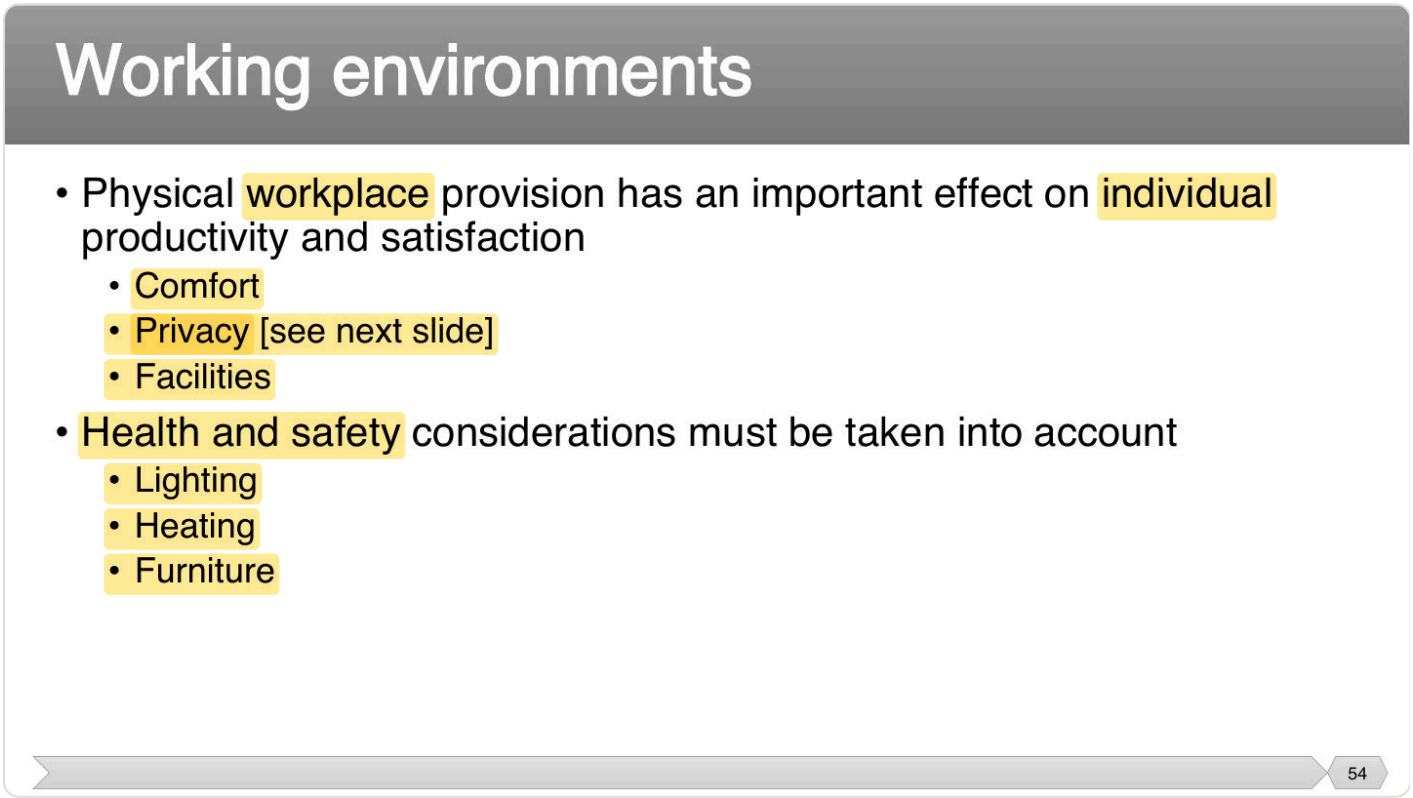
A The project is fully agile

**B The project is short with no time to learn a new language**

C The project is long

D The candidate has a postgraduate degree

SLIDE 54 / 77 **Working environments**




NOTES

1. People who are too hot, or who can't see properly, work worse and get tired faster. So the **workplace** is a real productivity lever, and cheap compared with salaries.
2. These aren't luxuries, since people concentrate better when they're comfortable and undisturbed. **Privacy** is singled out because of how much focused work depends on it.
3. This one isn't optional, because it's about people's wellbeing and is often a legal duty. **Health and safety** covers the basic conditions a workplace must meet.

TIPS & TRICKS

- **PaCiFy**: Privacy, Comfort, Facilities.

EXAM ANGLE

Prediction Q1

Which are health and safety considerations in the workplace?

**A** **Lighting, heating and furniture**

> That's the slide's list.

**B** Salary, bonuses and promotion

> These are rewards.

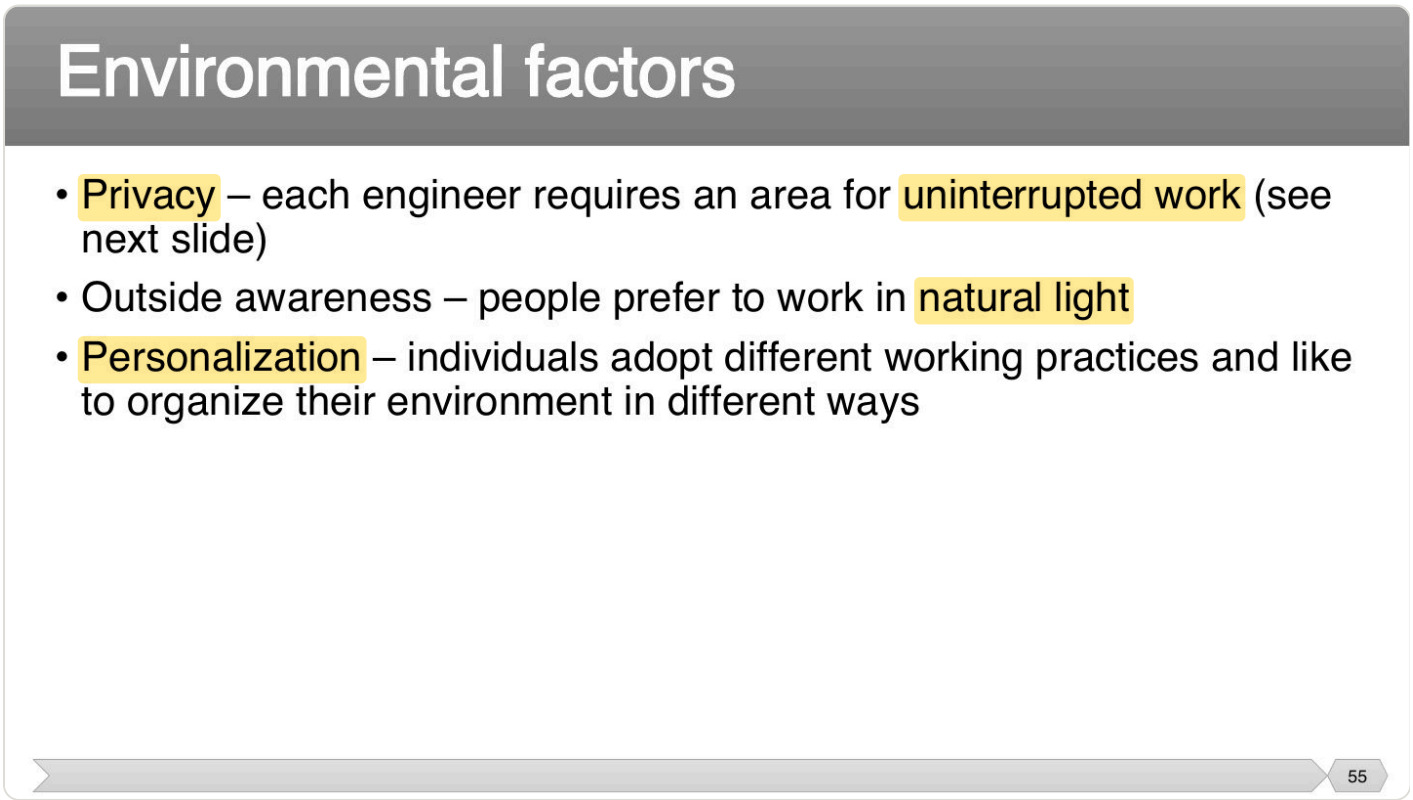
**C** Scope, time and cost

> These are constraints.

**D** Stand-ups, sprints and backlogs

> These are Scrum items.

SLIDE 55 / 77 **Environmental factors**



NOTES

1. **Privacy**: each engineer needs an area for **uninterrupted work**, because deep work is destroyed by interruptions.
2. Windowless rooms affect mood over time. It seems a small thing, yet **natural light** comes up often in research on satisfaction at work, and it costs little to consider.
3. Allowing people to arrange their own space and way of working tells them they're trusted. **Personalization** helps them feel at home and work the way that suits them.

TIPS & TRICKS

- **POP**: Privacy, Outside awareness, Personalization.

EXAM ANGLE

Prediction Q1

Which factor is about letting people arrange their environment their own way?

A Privacy

> That's about uninterrupted work.

B Outside awareness

> That's about natural light.

C **Personalization**

> That's the definition.

D Egoless programming

> That's about code ownership.

# Workspace organization

# Workspace organization

- Workspaces should provide private spaces where people can work without interruption
  - Providing individual offices for staff has been shown to increase productivity
- However, teams working together also require spaces where formal and informal meetings can be held



NOTES

1. Each interruption costs more than the interruption itself, since it takes time to get back into a hard problem. That's why **individual offices** raise productivity, even with their cost.
2. But teams also need shared spaces for **formal and informal meetings**. Balance focus and collaboration.

EXAM ANGLE

Prediction Q1

Which is true of workspaces?

**A** **People need private spaces and also spaces to meet**

> The slide calls for both.

**B** Only open spaces are productive

> Individual offices are shown to raise productivity.

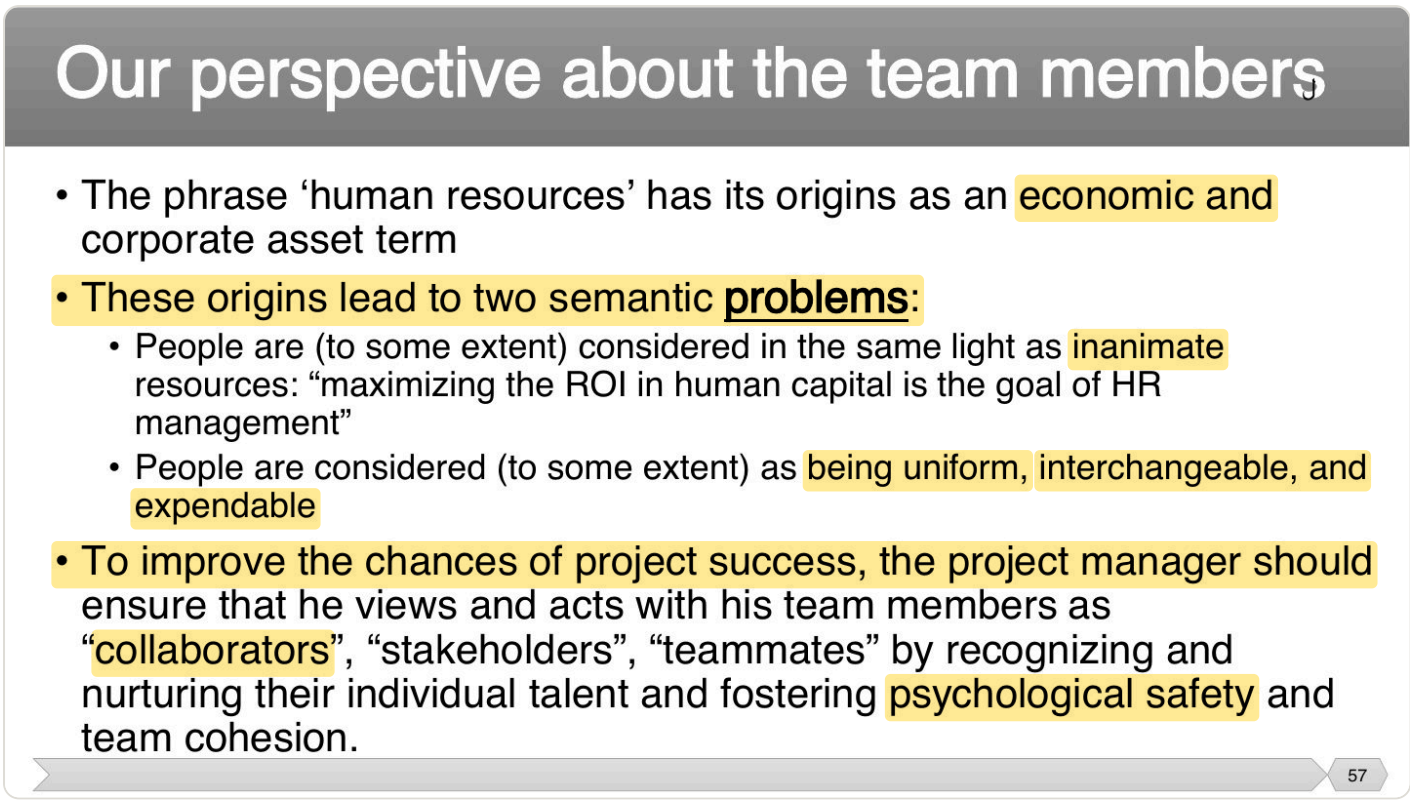
**C** Meeting spaces are unnecessary

> Teams need them.

**D** Offices always lower productivity

> That's the opposite.

 **Our perspective about the team members**




NOTES

1. Words shape how we think. Calling people “resources” puts them in the same category as desks and servers, which comes from the **economic** origin of the term.

2. If you see people as parts, you assume one programmer can simply replace another, and that’s rarely true. Treating them as **interchangeable** also makes them feel disposable, so the good ones leave.

3. Treating people as partners changes how you act: you ask instead of telling, and develop people instead of using them up. **Psychological safety** then lets them speak up, which protects the project.

EXAM ANGLE

Prediction Q1

What problem comes from the phrase “human resources”?

A It makes people too expensive

> That isn’t stated.

**B People can get treated as uniform, interchangeable and expendable**

> It’s one of the two problems named.

C It removes the need for a PM

> It doesn’t.

D It improves psychological safety

> It works against this.

Prediction Q2

How should a PM view team members instead?

A As costs to minimize

> That’s the same problem.

B As replaceable parts

> Also the problem.

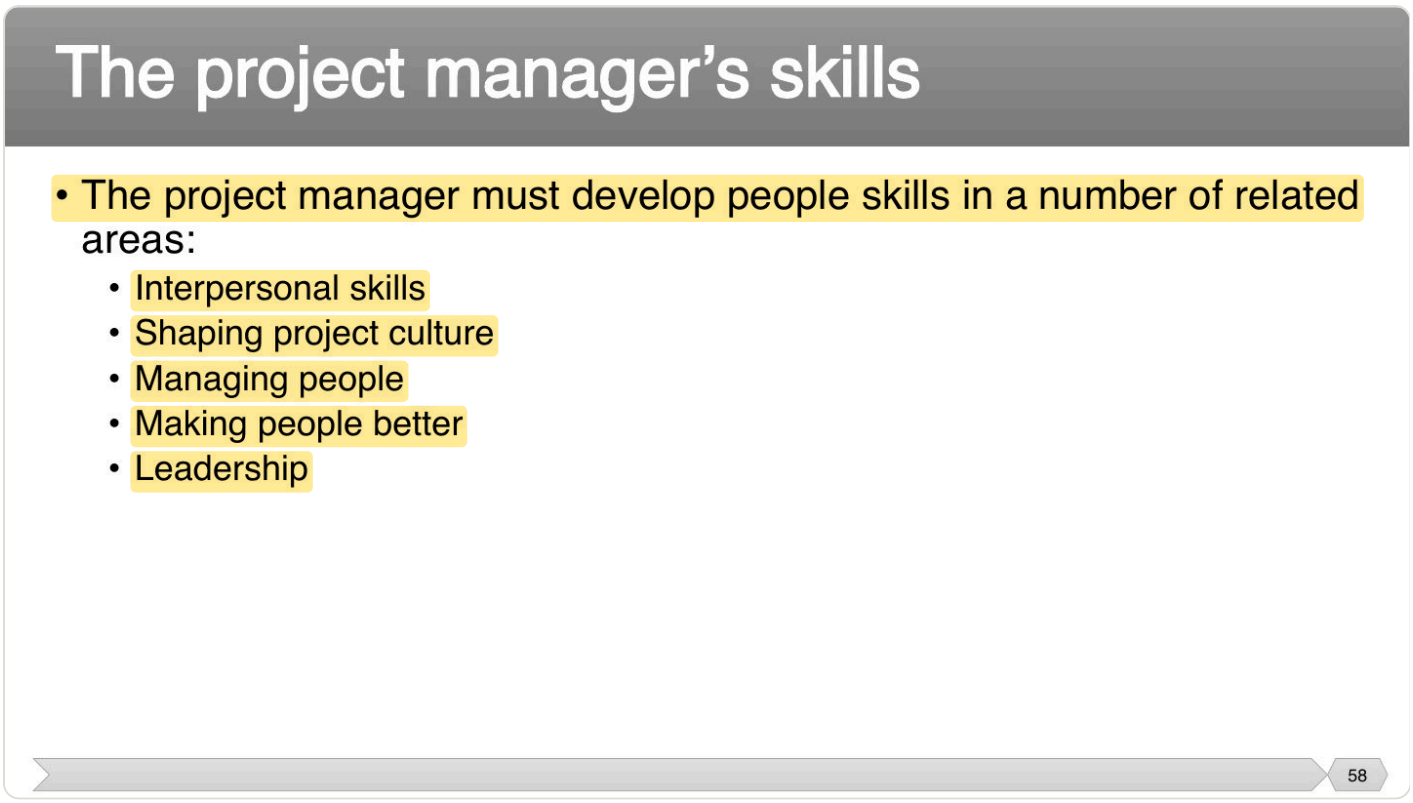
**C As collaborators, whose talent is nurtured**

> That’s the alternative offered.

D As inanimate resources

> That’s the problem.

 **The project manager's skills**




NOTES

1. Think of these as different jobs: **interpersonal skills** are how you deal with one person, and **shaping culture** is how you influence how the whole team behaves...
2. ...**managing** is about getting today's work done, while **making people better** is about what they'll be able to do next year...
3. ...and **leadership** ties these together, by giving the team a direction they want to follow.

TIPS & TRICKS

- **SLiMMIng**: Shaping project culture, Leadership, Managing people, Making people better, Interpersonal skills.

EXAM ANGLE

Prediction Q1

Which is NOT one of the PM's five people-skill areas?

**A Procurement**

That isn't on the list.

B Leadership

This is on the list.

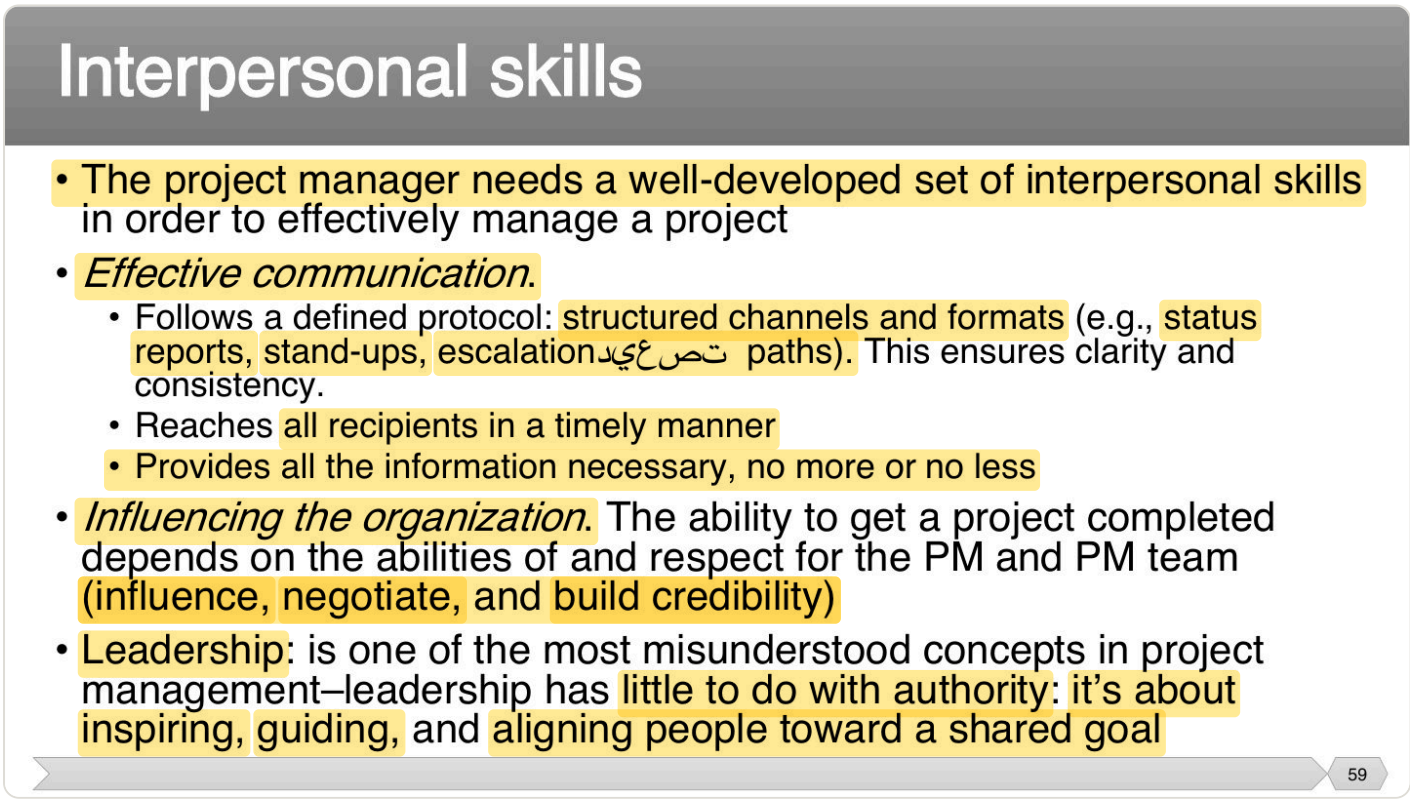
C Making people better

This is on the list.

D Shaping project culture

This is on the list.

SLIDE 59 / 77 **Interpersonal skills**




NOTES

1 Too little and people work on outdated information; too much and they stop reading. **Effective communication** means a known way of sharing, so everyone gets what they need, on time.

2. A PM often needs help from people who don't report to them. They can't order it, so **influencing the organization** depends on credibility built earlier.

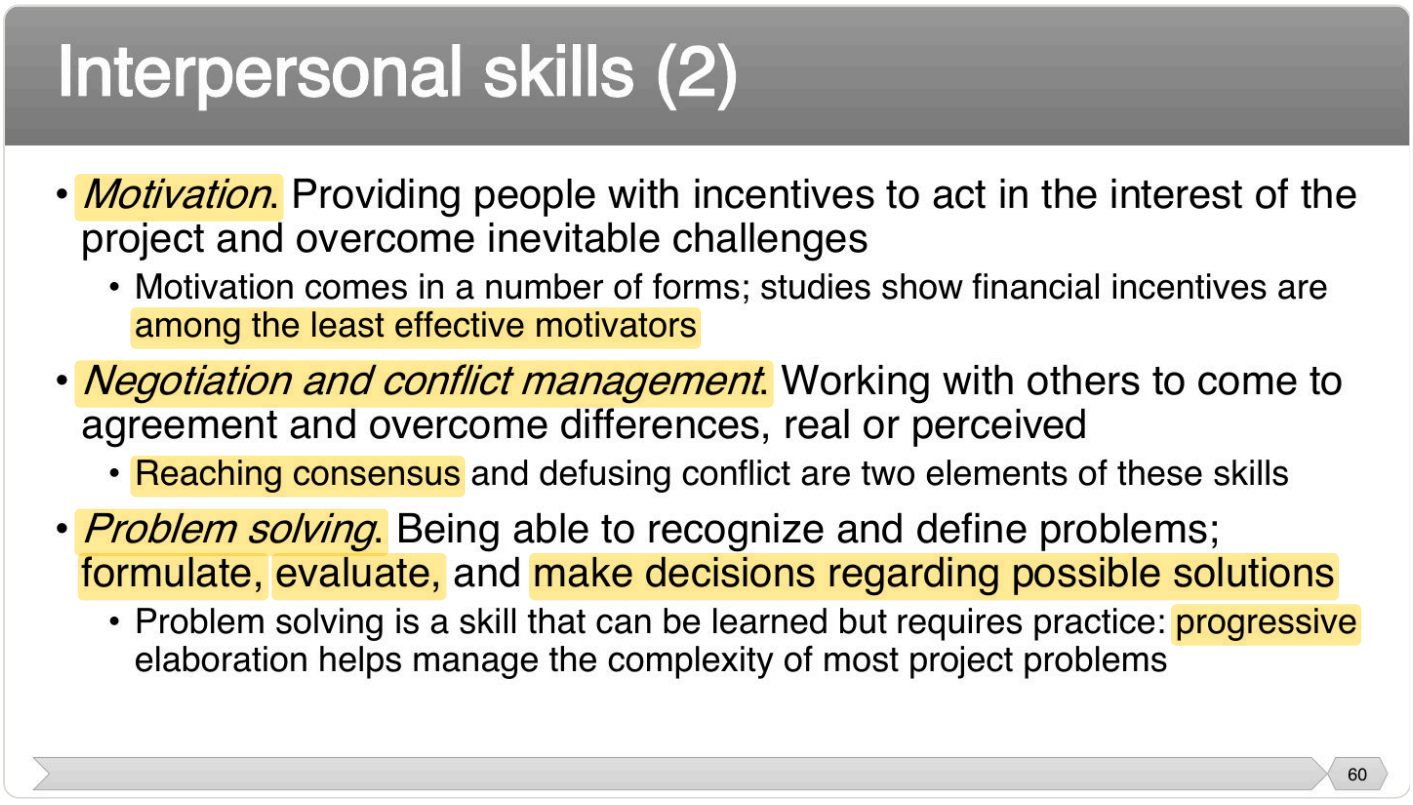
3. Exam trap: someone with no title can lead and a manager can fail to. **Leadership** is about whether people choose to follow you toward a goal, which has **little to do with authority**.

EXAM ANGLE

Past exam · Final 251 · Q3.1 Q1

In what ways do proficient communication abilities contribute to the management of project risks and the attainment of project success? Illustrate with an instance of how a breakdown in communication could affect a project, and propose methods for prevention or resolution.
**Answer:** Good communication surfaces risks and issues early, while they are cheap to handle; keeps requirements and expectations shared across the team and stakeholders; and supports fast, informed decisions. Example: the backend team changes an API without telling the frontend team, integration fails at the end of the sprint, causing rework and a missed release. Prevention: a communication plan with defined channels (stand-ups, change notifications, API contracts), timely updates to everyone affected, and an escalation path; resolution: a joint meeting to agree the interface and re-plan.

SLIDE 60 / 77 **Interpersonal skills (2)**




NOTES

1. Money works up to a point, but it fades quickly and can't make a job interesting. The surprising point is that **financial incentives** are among the weakest motivators, so try recognition, growth and meaningful work first.

2. Differences may only be perceived, so listening can dissolve them. A PM rarely has the power to simply impose an outcome, so **negotiation** is an everyday tool, not a rare event.

3 Most project problems are poorly defined at first. So the first skill is working out what's really wrong, and then **progressive elaboration** means solving complex problems in steps rather than all at once.

TIPS & TRICKS

- Slides 59 and 60 together, **IMPLEmeNt**: Influencing, Motivation, Problem solving, Leadership, Effective communication, Negotiation.

EXAM ANGLE

Prediction Q1

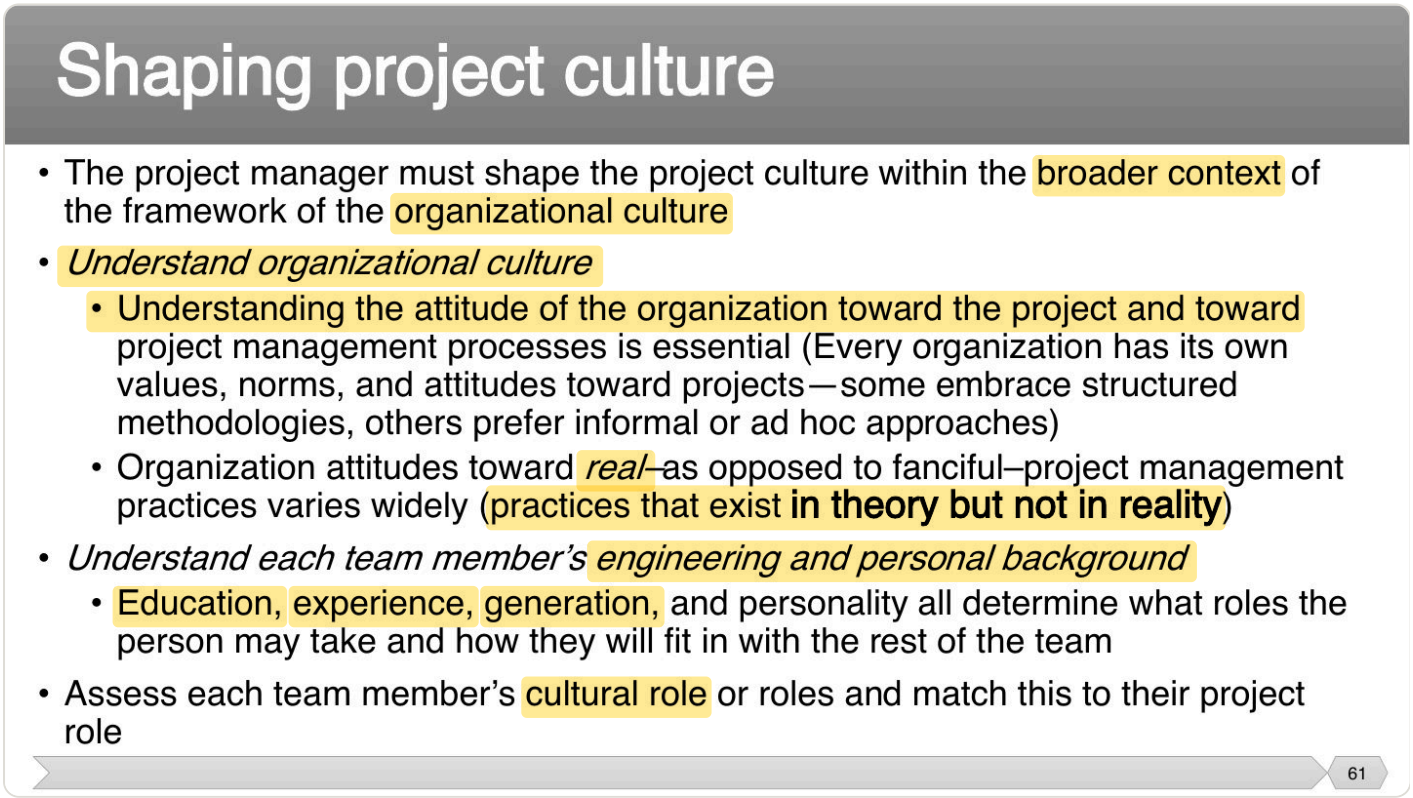
Studies show financial incentives are among the most effective motivators for project teams.

T True

**F False**

The slide says they are among the least effective.

SLIDE 61 / 77 **Shaping project culture**




NOTES

1 You can't build a culture that fights the wider company for long. So learn how your **organizational culture** treats projects first, and shape your team's habits within those limits.

2 Some companies have impressive process documents that nobody follows. Knowing whether **real** practice matches the "fanciful" version tells you which rules you actually need to work with.

3 Someone’s education and past experience shape how they work and what they’ll accept. A person’s **background** can tell you what kind of role they’d do well in, so you need to know your people.

4 People tend to play a natural part in a team, whatever their job title. Knowing each person’s **cultural role** lets you give them work that suits it, which the next slide explains.

EXAM ANGLE

Prediction Q1

Why must a PM first understand the organization’s culture?

A **Project culture is shaped within that broader context**

> You can’t build a culture that fights the wider company for long.

B Culture is the same everywhere

> It differs by organization.

C It replaces the project plan

> It doesn’t.

D It decides the code language

> That isn’t the point.

Prediction Q2

What does “fanciful” project management refer to?

A Team celebrations

> That isn’t the meaning.

B Creative product design

> That isn’t the meaning.

C **Practices that exist in theory, but not in reality**

> Compare real and fanciful practices.

D Practices that are always followed

> That’s real practice.

 **Shaping project culture**

# Shaping project culture

- Match cultural and project roles to people
- Representative cultural roles include:
  - Leader (Takes initiative, inspires others)
  - Listener/talker (Facilitates communication, bridges gaps)
  - Complainer/naysayer (who habitually expresses negative or skeptical views: Raises concerns, challenges assumptions)
  - Expert (Offers deep knowledge and credibility)
  - Charger/plodder (Pushes forward with energy)
- Significant project roles include:
  - *Project manager*. Guides the team planning and tracks progress against the plan
  - *Team leader*. Builds and maintains an effective team
  - *Development leader*. Produces a superior product
- Remain non-judgmental: every cultural role has potential value for the team

62

NOTES

1. These describe how people behave in a team, not their job title. A tester may be the team's **expert** or its **charger**, and a healthy team needs a mix...
2. ...and the **complainer/naysayer**, who habitually voices negative or skeptical views but raises concerns and challenges assumptions, often the first to spot a risk.
3. Don't confuse these with the cultural roles: these are assigned jobs. The **team leader** looks after people, and the **development leader** looks after the product, and on a small team one person may do both.
4. It's easy to label a critic as a problem, but they may be the only one seeing a risk. Being **non-judgmental** means asking what each role can contribute, rather than trying to change who they are.

TIPS & TRICKS

- Cultural roles, in slide order: **L**azy **L**ions **C**atch **E**very **C**ow. Leader, Listener or talker, Complainer, Expert, Charger or plodder.

EXAM ANGLE

Prediction Q1

Someone often points out what might go wrong. How should the PM treat this person?

A Make them the sponsor

> That's not what's suggested.

B Remove them right away

> The slide says to remain non-judgmental.

C Ignore all their concerns

> They may be first to spot a risk.

**D As having potential value, since they raise concerns and challenge assumptions**

> Every cultural role has potential value.

Prediction Q2

Which of these is a project role rather than a cultural role?

A Listener or talker

> That's a cultural role.

B Charger or plodder

> That's a cultural role.

C Complainer

> That's a cultural role.

**D Development leader**

> It's an assigned job.

 **Project environments (context: project culture)**

# Project environments (context: project culture)

- Cultural and social factors
  - Projects are planned and implemented in a social and cultural context
  - Some factors that will influence a project might be: economic, educational, ethical, ethnic, and religious
  - These factors can affect the authority of the project manager
- International and political
  - As globalization expands, familiarity with local laws and customs is essential for effective project management
  - Shifting international, national, and local political factors can have dramatic influences on a project
  - Some mundane factors that will influence a project might be: time zone differences, holidays, travel requirements, teleconferencing capabilities



NOTES

1. A young PM may struggle where age earns respect, regardless of the org chart. So **authority** doesn't only come from a title, and your authority in one culture may not carry over to another.
2. What's normal in one country may be illegal in another, and a change of government can change the rules. So on international projects, **local laws and customs** are real planning inputs.
3. If half the team sleeps while the other half works, every question can take a full day to answer. These **mundane** details can quietly cause much of the delay on a global project.

EXAM ANGLE

Prediction Q1

A young PM is struggling where age earns authority. Which factor does this show?

A Time zones

That's a mundane factor.

B Contingency reserves

That's about money.

C Only the org chart decides authority

Context matters as well.

**D Cultural and social factors can affect the PM's authority**

The slide says so.

Prediction Q2

Half a team is asleep when the other half works. Which kind of factor is this?

A A deliverable

That's an output.

B A hygiene factor

That's from motivation.

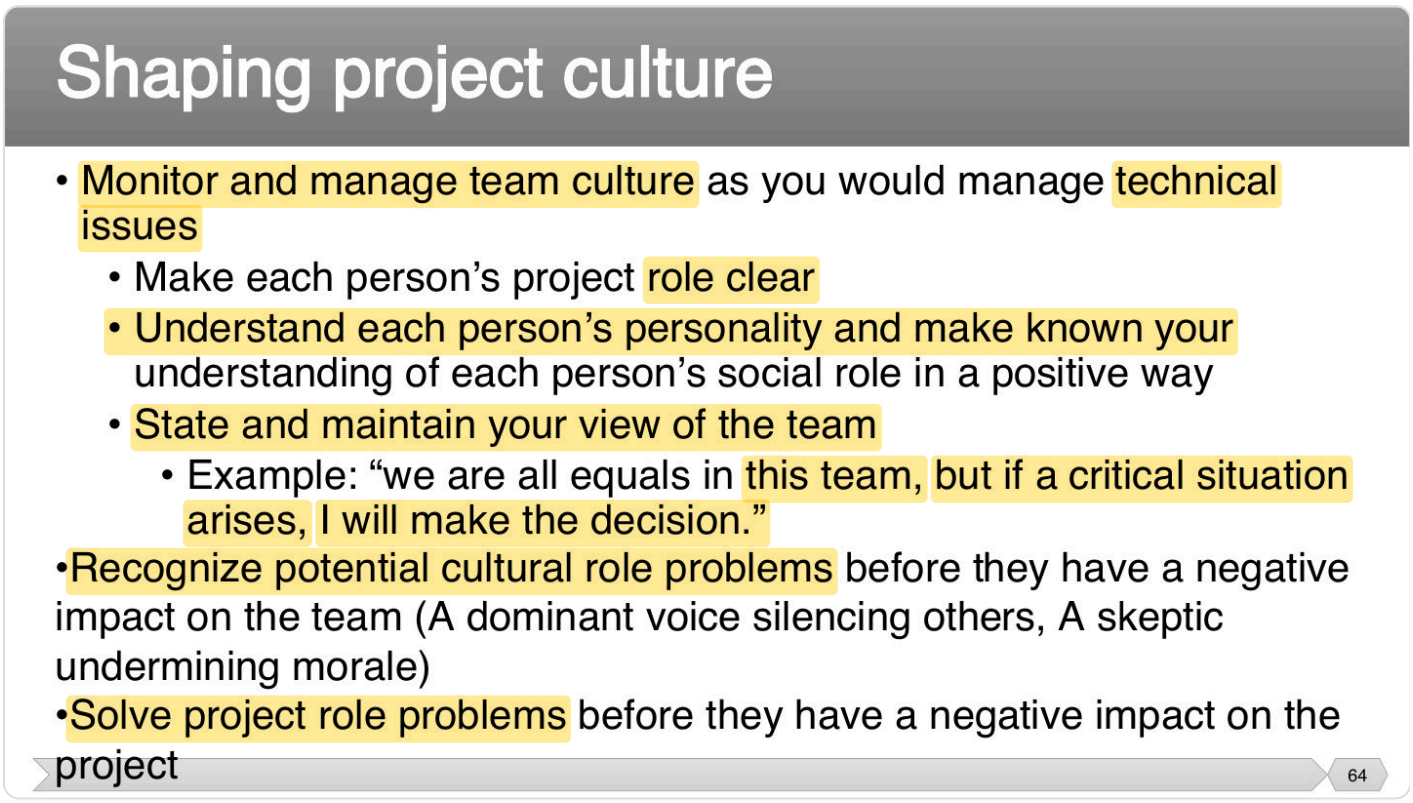
**C A mundane factor: time zones**

The slide lists these.

D A political factor

It's not political.

 **Shaping project culture**




NOTES

1. Culture problems can kill a project just as a technical flaw can, and yet nobody tracks them. Treat team culture as you would **technical issues**: look for it, discuss it and fix it.

2. People who are unsure of their role either overlap or leave gaps. **Clear roles** stop that, and telling someone you value their style helps them accept the role.

3. People work better when they know the rules for hard moments. The slide's example shows it: equal in normal times, and one decider in a crisis. Stating your **view of the team** up front avoids a fight later.

4. It's far easier to deal with one person dominating early than after others have stopped talking. So **recognize** the warning signs, and act before they harm the team.

EXAM ANGLE

Prediction

Q1

A PM says: “We’re all equals, but in a crisis I decide.” What’s this an example of?

**A** Groupthink

> That’s protecting group harmony.

**B** Micromanagement

> It’s simply setting expectations.

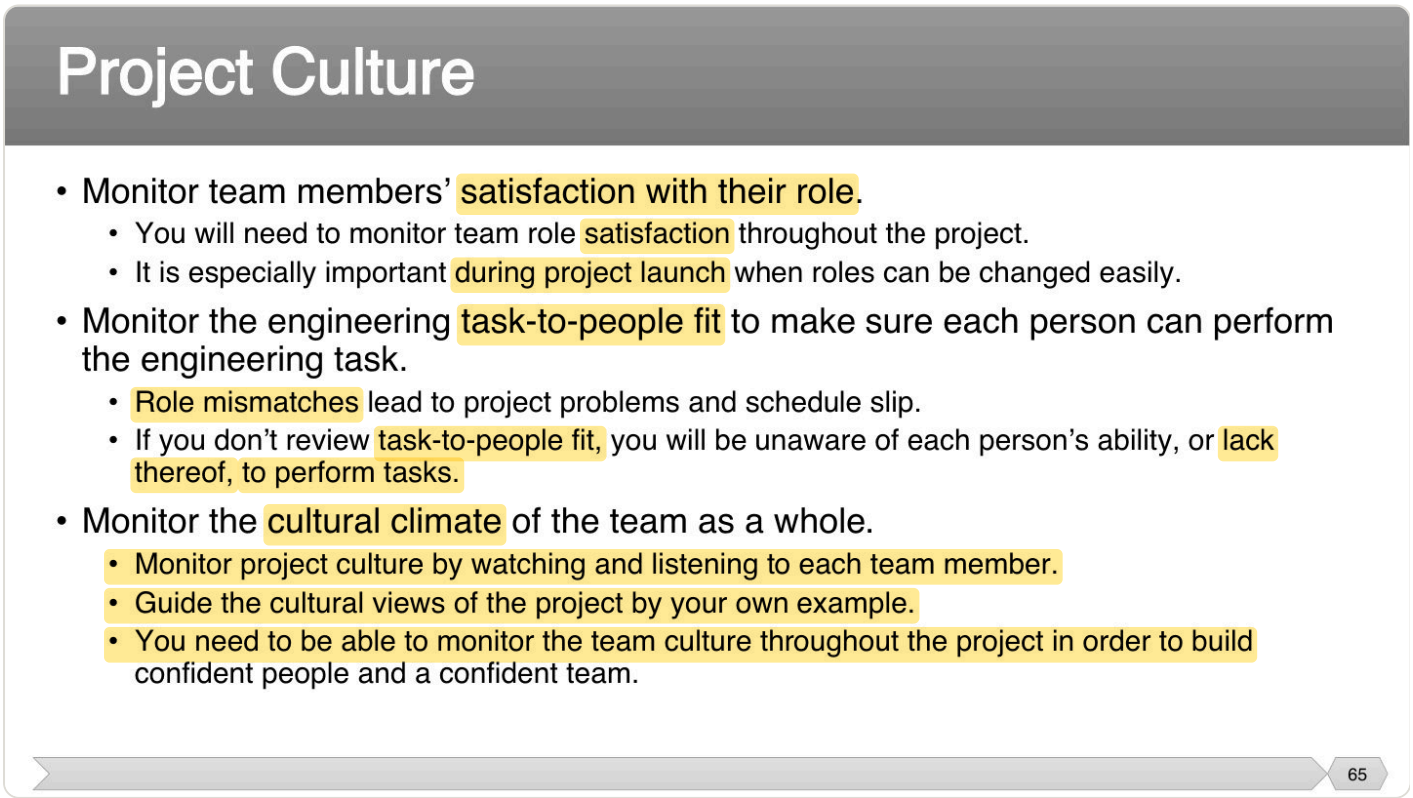
**C** **Stating and maintaining your view of the team**

> It’s the slide’s example.

**D** Withdrawal

> That’s avoiding conflict.

SLIDE 65 / 77 **Project Culture**




NOTES

1. A role that seems fine on day one may turn out badly, and people rarely volunteer that they’re unhappy. Keep asking about **satisfaction**, especially **during launch**, when a change costs almost nothing.

2. Don’t assume a job title means someone can do the task. A **role mismatch** shows up late as missed dates, and checking early is much cheaper than finding out after the schedule has slipped.

3. You won’t find team mood in a report. Watch how people talk in meetings. Your behavior sets the tone too, so the **cultural climate** responds to your own example.

---

EXAM ANGLE

Prediction Q1

When is it easiest to change someone’s role?

A Just before release

> That’s when it’s costly.

B After the project closes

> That’s too late.

C It can never change

> It can, especially early.

**D During project launch**

> The slide stresses monitoring at this point.

Prediction Q2

What can a role mismatch lead to?

A Better estimates

> Nothing says so.

B Fewer defects

> That’s the opposite.

**C Project problems and schedule slip**

> The slide says this.

D Automatic savings

> That’s not what’s stated.

 **Managing People**

# Managing People

- The main problems of our work are not so much technological as sociological in nature
- (optional reading) One of the most enlightened sources for information about managing people is:
  *Peopleware: Productive Projects and Teams*, Third Edition,
- Most technical managers come from technical backgrounds and manage that way as if they are solving an engineering problem
- Most managers do not consider individual idiosyncrasies (personal strange habit), yet these can make or break a project

66

NOTES

1. A project rarely fails because the technology was impossible. It's more often poor communication or a team that doesn't work together, and that's what **sociological** means here.
2. **Peopleware** (DeMarco and Lister) is the recommended optional reading on managing people.
3. Promoted engineers tend to treat people as a puzzle with a correct answer. But people aren't logical systems, so one person's **idiosyncrasies**, like needing quiet, can decide whether they do great work.

EXAM ANGLE

Prediction Q1

According to the slide, what are the main problems in our work?

**A More sociological than technological**

People issues, not technology.

B Only legal

That's not what it says.

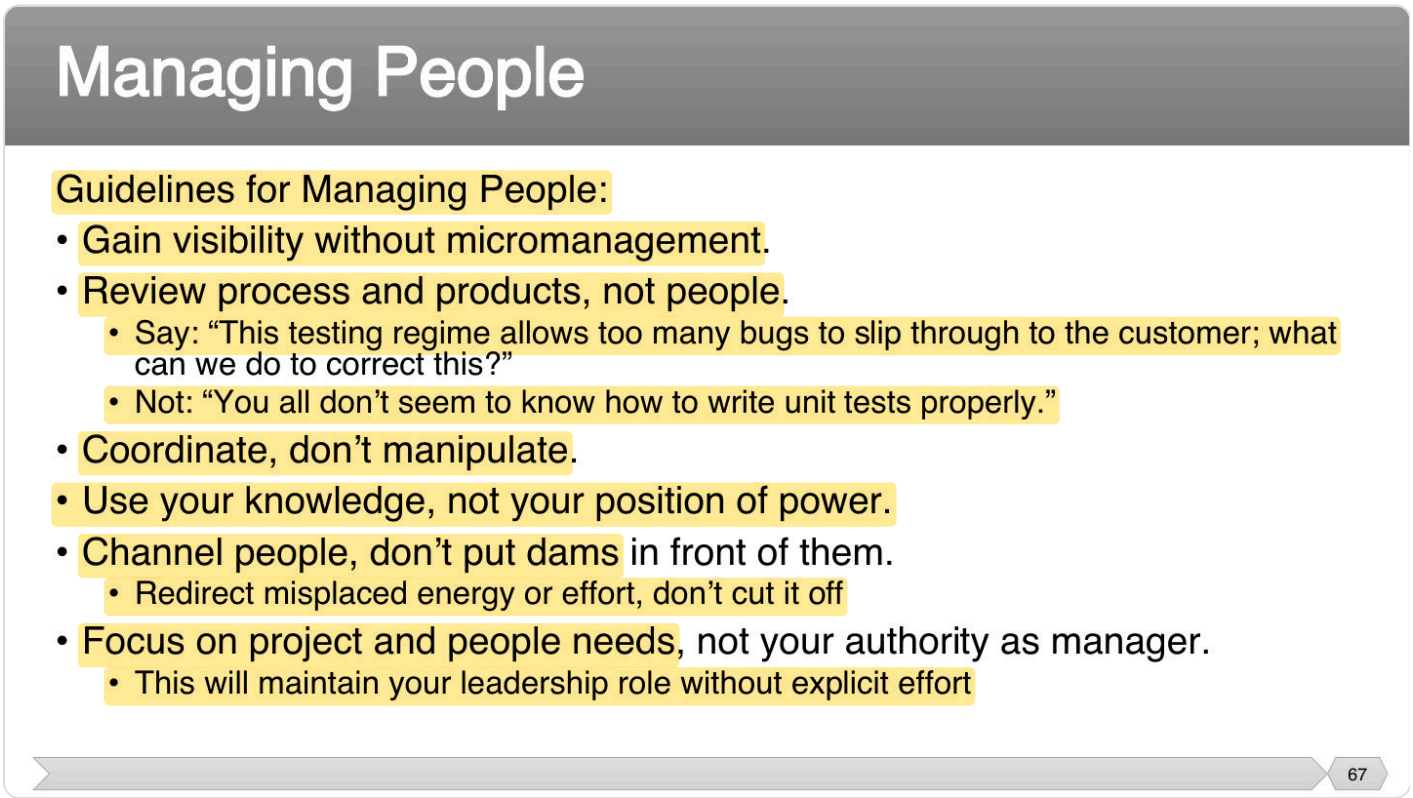
C Only technological

That's what it argues against.

D Only financial

That's not what it says.

SLIDE 67 / 77 **Managing People**




NOTES

1 **Gain visibility without micromanagement**: know the status through boards and reviews, not by watching over shoulders.

2. Criticizing someone's ability makes them defensive, and they stop listening. Discussing the **process** puts everyone on the same side against the problem, as the slide's two sentences show.

3. Tricking or pressuring people may work once, but it destroys trust. **Coordinate** by explaining why, and if someone's effort is going to the wrong place, redirect it rather than discouraging them altogether.

4. People notice when you're protecting your position rather than helping. When your attention is on what the project and team need, they respect you without being told to. That's why you **focus on needs**.

EXAM ANGLE

Prediction Q1

Which statement follows the guideline "review process and products, not people"?

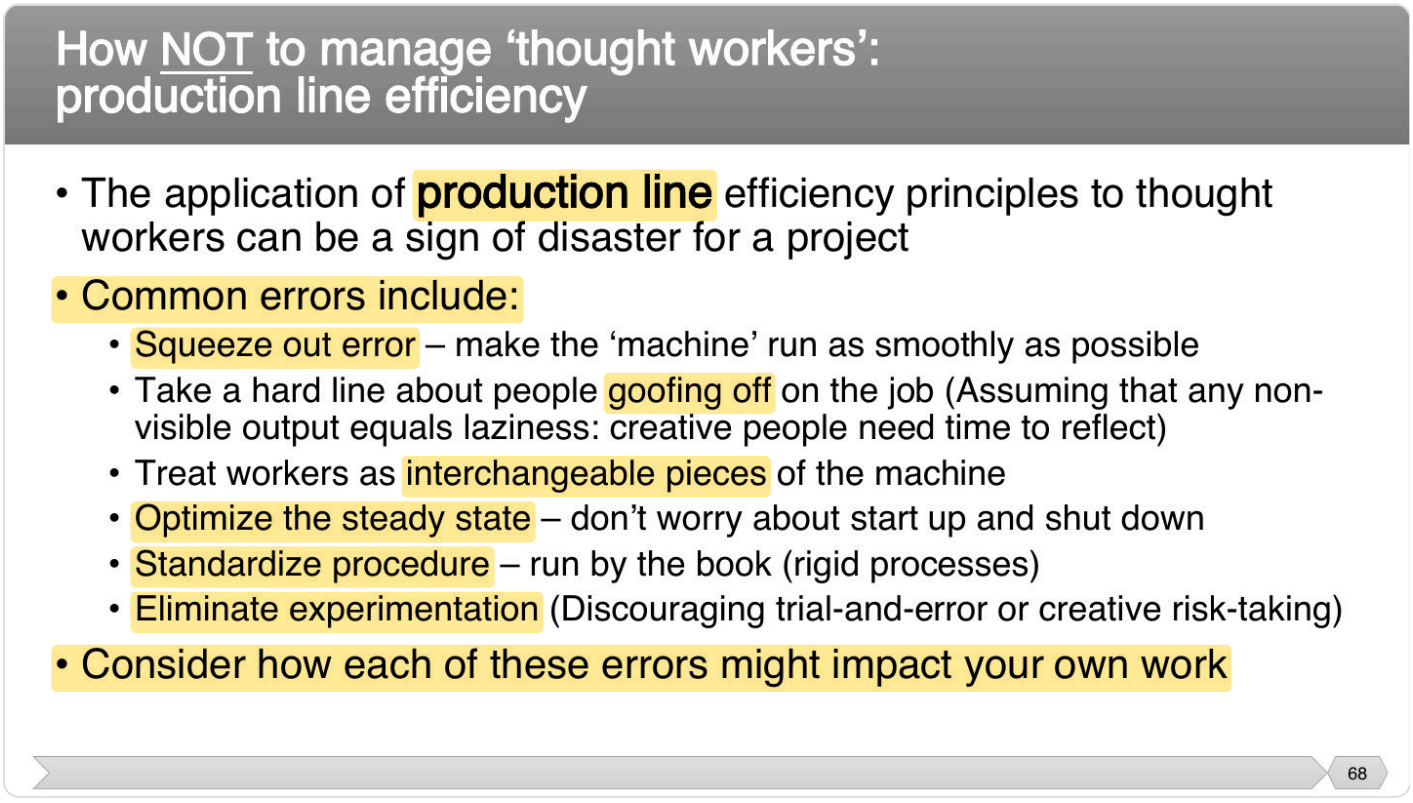
A "You don't know how to write unit tests."

B "Whoever wrote this is careless."

C "I'll decide, since I'm the manager."

D **"This testing regime lets too many bugs through; how can we fix it?"**

SLIDE 68 / 77 **production line efficiency**



NOTES

1. Factory methods assume that identical work gets repeated. Software is different each time, so applying **production-line** thinking to it causes the errors listed below.

2. On a line, an error is a defect, while in design it's often how you learn. Someone staring out a window may be solving your hardest problem, so treating that as **goofing off** drives out your best thinking.

3. A new person takes months to reach full speed, so people aren't **interchangeable** like machine parts. Plans that assume a smooth steady state ignore the expensive start and end of work.

4. Strict rules suit repeated tasks, but software keeps meeting new problems. If trying something new isn't allowed, you won't find better ways, so **eliminating experimentation** is harmful.

EXAM ANGLE

Prediction Q1

A manager treats a developer thinking at their desk as “goofing off”. What’s the error?

A Standardizing procedure

> That’s another error.

B Optimizing the steady state

> That’s another error.

**C Assuming that non-visible output means laziness**

> Creative people need time to reflect.

D Allowing experimentation

> That isn’t an error.

Prediction Q2

Why is treating workers as interchangeable an error?

**A People aren’t identical machine parts**

> Thought work depends on the person.

B It’s good for ramp-up

> It ignores ramp-up.

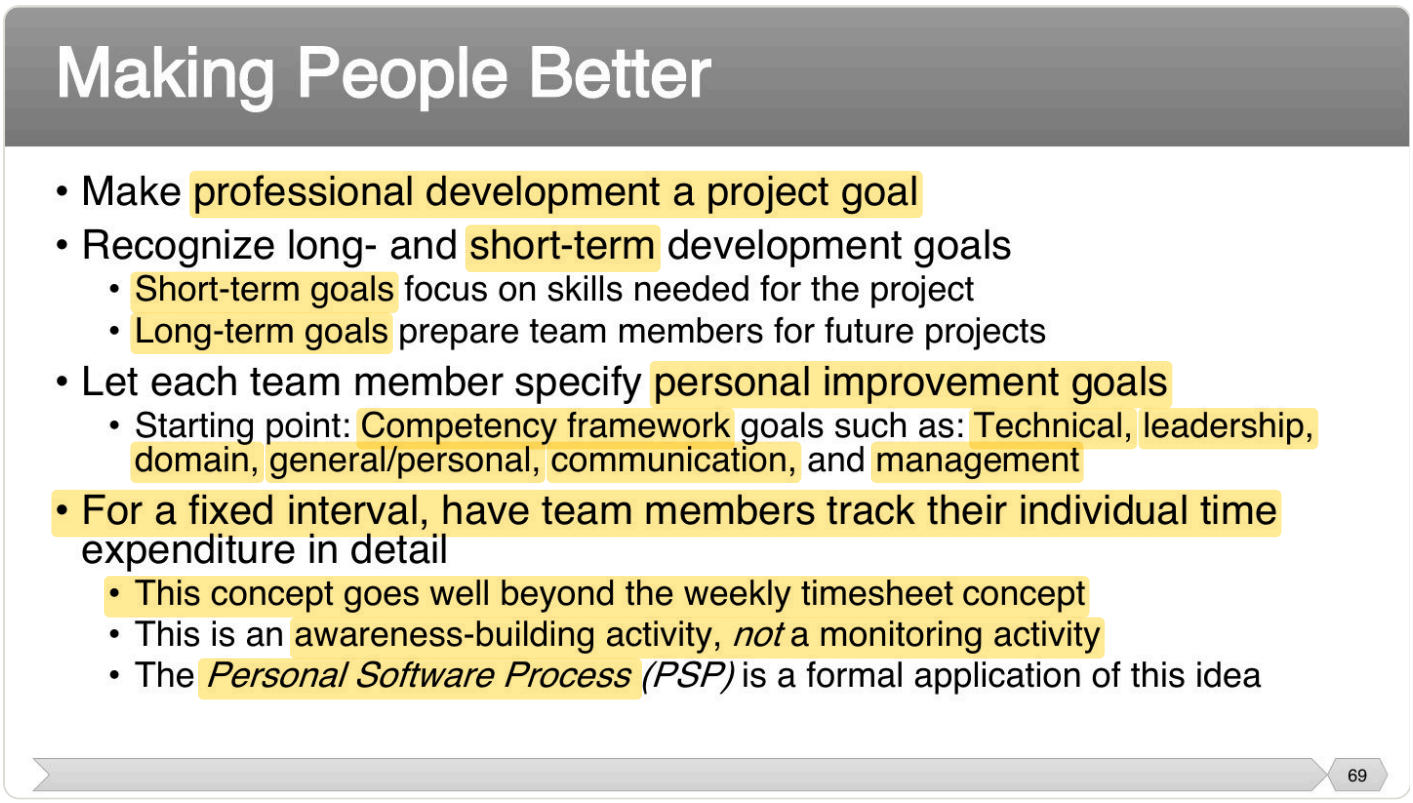
C It improves creativity

> It doesn’t.

D Machines are cheaper

> That isn’t the reason.

 **Making People Better**




NOTES

1. It's usually the first thing dropped under pressure, unless it's a stated goal. Treating **professional development** like any other goal means time and money get set aside for it.
2. You need both. **Short-term** learning helps this project now, while **long-term** learning keeps people growing, and it's a large part of why good staff stay.
3. People work harder at goals they chose. A **competency framework** just gives them a useful list to pick from, so that "improve" turns into something specific.
4. Most people don't know where their time really goes, and the results often surprise them. It only helps if it's safe, so stress that this is for their own **awareness**, not for monitoring.

## EXAM ANGLE

Prediction

Q1

A team tracks its time in detail for a few weeks. What is the purpose?

A Catching people out

> That’s monitoring, which it isn’t.

B Replacing all estimates

> It helps estimating, without replacing it.

C Setting salaries

> That isn’t stated.

**D Building awareness, not monitoring**

> The slide stresses this.

Prediction

Q2

A short-term development goal focuses on:

A Future projects only

> That’s long-term.

B The company’s mission

> That’s not stated.

C Salary reviews

> That’s not stated.

**D Skills needed for this project**

> Long-term goals prepare for future projects.

# Making People Better

# Making People Better

g

- *Have team members track their individual time*
  - Goes beyond the weekly timesheet concept
- One formal approach to this is the *Personal Software Process* (PSP) which shows developers how to:
  - Manage the quality of their projects
  - Make commitments they can meet
  - Improve estimating and planning
  - Reduce defects in their products

[See note below for more info.]

- One element of PSP is to track daily time spent on *all* project-related tasks:
  - Reading
  - Writing
  - Meeting
  - Training
  - Requirements
  - Design
  - Coding
  - Testing
  - Deployment
  - E-mail
  - Phone



NOTES

1. A weekly timesheet is too rough to teach you anything. Tracking **daily** gives a developer real evidence, such as how long testing actually takes compared with what they guessed.

2. With your own numbers, your next estimate gets better, and you stop promising what you can't deliver. That's how the **PSP** helps developers **make commitments they can meet**.

EXAM ANGLE

Prediction

Q1

What is one thing the PSP shows developers how to do?

**A** Avoid all meetings

> That's not it.

**B** Manage other teams' budgets

> That's not it.

**C** Skip testing

> That's not it.

**D** **Make commitments they can meet**

> Their own data improves their estimates.

 **Leadership**

# Leadership

- *Leadership* is the capacity to mobilize a group to solve the most challenging problems (wrong to think that leadership is about authority)
- Two types of leadership:
  - *Technical leadership* (a.k.a. *authority*) motivates people when the problem can be solved by *known* means: the leader directs, instructs and applies best practices.
  - *Adaptive leadership* motivates people to solve problems that *cannot* be solved by known means (needed when the problem is **complex or novel**)
    - Requires learning on the part of the leader as well as the group (facilitate learning and research)
    - Often involves leading with good questions (provoke thinking, guide inquiry)
    - Requires a deep clarity of purpose (to keep the team anchored toward its goals) : Example: during brainstorming meetings, the leader is the one that needs to stop the conversation from diverging to unimportant arguments, problems and bring the discussion back to the essential questions.

71

NOTES

1. Having a title doesn't get a group to solve hard problems. **Mobilize** is the key word: people have to want to take it on, so it's wrong to equate leadership with authority.
2. If the problem has a known fix, the team just needs someone who knows it. **Technical leadership** is direct and efficient, and asking "what do you all think?" would just waste time.
3. If nobody knows the answer, a leader giving orders is only guessing. **Adaptive leadership** helps the group learn its way there, and good questions do more than instructions.
4. When people are exploring, it's easy to drift. A **clarity of purpose** gives everyone a reason to keep going without a clear path, and lets the leader pull a discussion back.

## EXAM ANGLE

Prediction Q1

A novel problem has no known solution. Which leadership is needed?

**A** None

> Leadership is still needed.

**B** Technical

> That's for known means.

**C** Autocratic only

> Giving orders doesn't help when nobody knows the answer.

**D** **Adaptive**

> It's for problems known means can't solve.

Prediction Q2

Leadership is mainly a matter of having authority.

**T** True

> It says it's wrong to think leadership is about authority.

**F** **False**

> The slide calls that wrong: it's the capacity to mobilize a group.

 **Technical vs. adaptive leadership**

# Technical *vs. adaptive leadership*

| | Technical leadership | Adaptive Leadership |
|---|---|---|
| Approach to problem-solving | Technically-oriented: solves problems by known means | Adaptive: approaches problems not solved by known means |
| Motivation driver | Vision: personal view of outcome. The leader's personal vision drives the team —"Here's what I see for us." | Sense of purpose: big-picture view of goals: The leader anchors the team in shared meaning —"Why does this matter?" |
| Tool for motivation | Personality: Charisma, technical expertise, confidence, and decisiveness inspire trust and action. | Progress: Motivation comes from learning, iteration, and visible movement toward clarity. |



NOTES

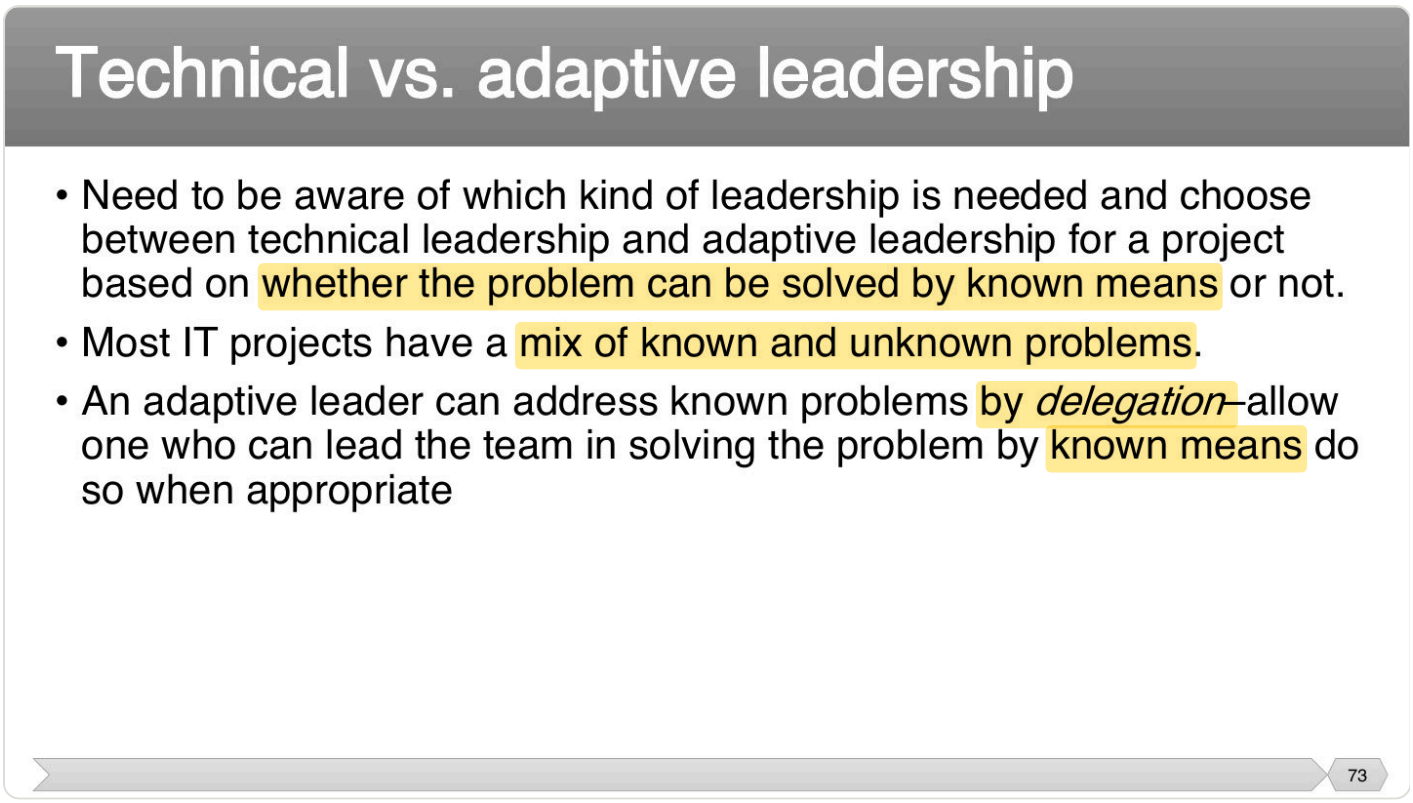
1. Ask first: does anyone know how to solve this? The answer decides which **approach** fits, and using the wrong one either wastes time or gives false confidence.
2. Technical says "here's what I see for us", and it works because they know the way. Adaptive doesn't have that map, so it uses a **sense of purpose**: why it's worth working through.
3. **Tool for motivation**: technical relies on **personality**: charisma, expertise, confidence, decisiveness. Adaptive relies on **progress**: learning, iteration and visible movement toward clarity. In short: technical = "follow me, I know"; adaptive = "let's figure it out together."

EXAM ANGLE

Past exam · Midterm 251 · V1 · Short Q5 (also V2 Short Q5) Q1

How can a project manager balance technical leadership with adaptive leadership in a software team?
**Answer:** First diagnose the problem: can it be solved by known means? When it can (a coding standard, a deployment the team has done before), use technical leadership: give direction from expertise and vision, or delegate it to someone who can lead it. When it can't (new technology, unclear requirements), use adaptive leadership: frame the purpose, lead with questions, let the team experiment and learn, and motivate through visible progress. Most IT projects mix both, so the PM switches between the two as the situation changes, delegating the known parts and focusing on the unknown ones.

 **Technical vs. adaptive leadership**




## NOTES

1. Neither style is always better, so diagnose first. Fixing a known bug doesn't need a discussion, and a brand-new product can't be directed by a checklist. **Known means** is the deciding question.
2. One project contains both: routine deployment work and a new algorithm nobody's tried. So a PM has to switch between them, which is why a **mix** matters.
3. A leader doesn't need to handle everything personally. **Delegation** lets the person who knows the answer lead that part, which frees the leader to work on the problems nobody has solved.

## EXAM ANGLE

**Prediction** Q1

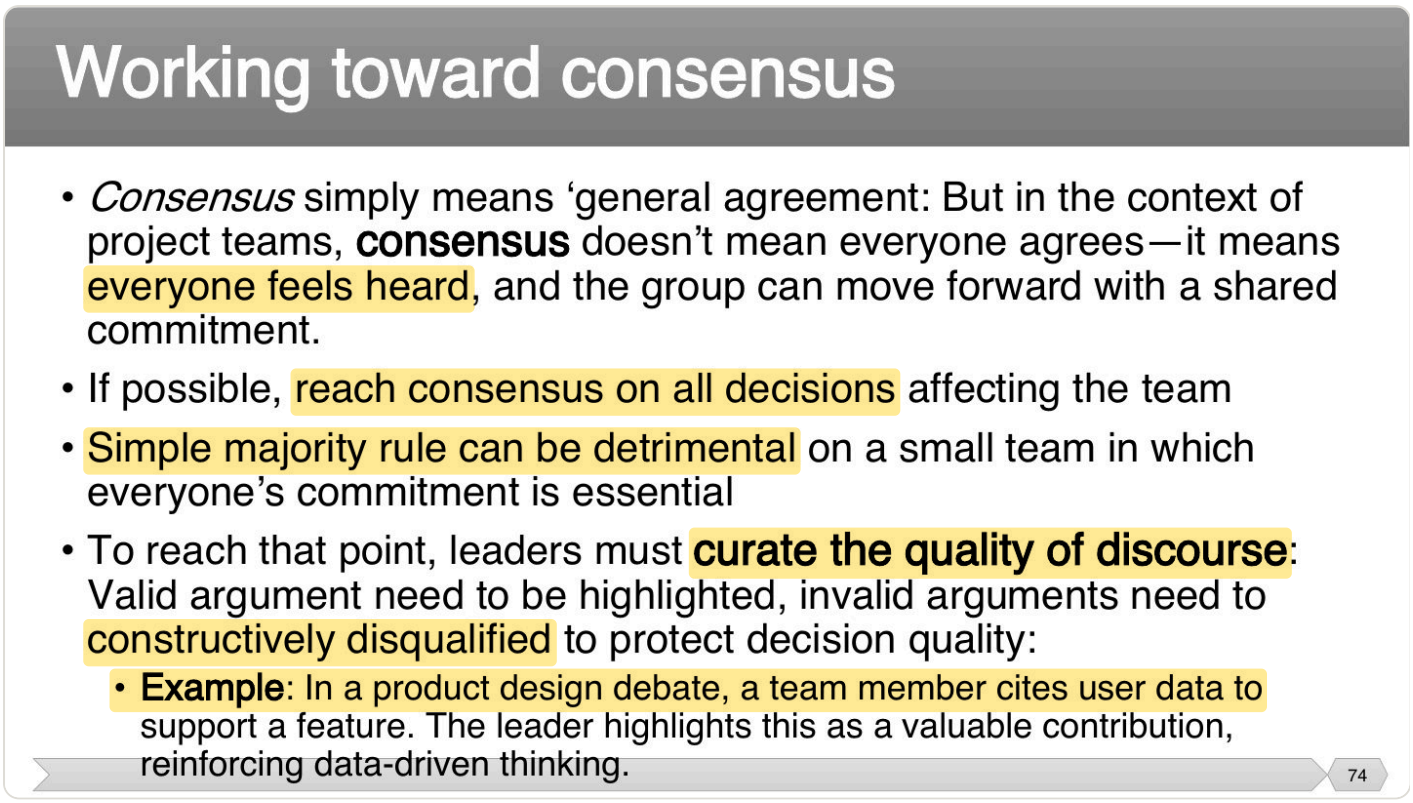
An adaptive leader should personally solve all known technical problems.

T True

F **False**

> They can address known problems by delegation.

 **Working toward consensus**




NOTES

1. Exam trap: consensus doesn't require full agreement. People can accept a decision they'd have made differently if they feel they had a fair hearing. That's why it's about being **heard**.
2. On a five-person team, a 3–2 vote leaves two people who may quietly stop trying, and you need all five. That's why **majority rule** can do harm, even though it seems fair.
3. Everyone being heard doesn't mean every argument is good. The leader should strengthen the good points and gently remove the bad ones, because they must **curate** what the decision rests on.

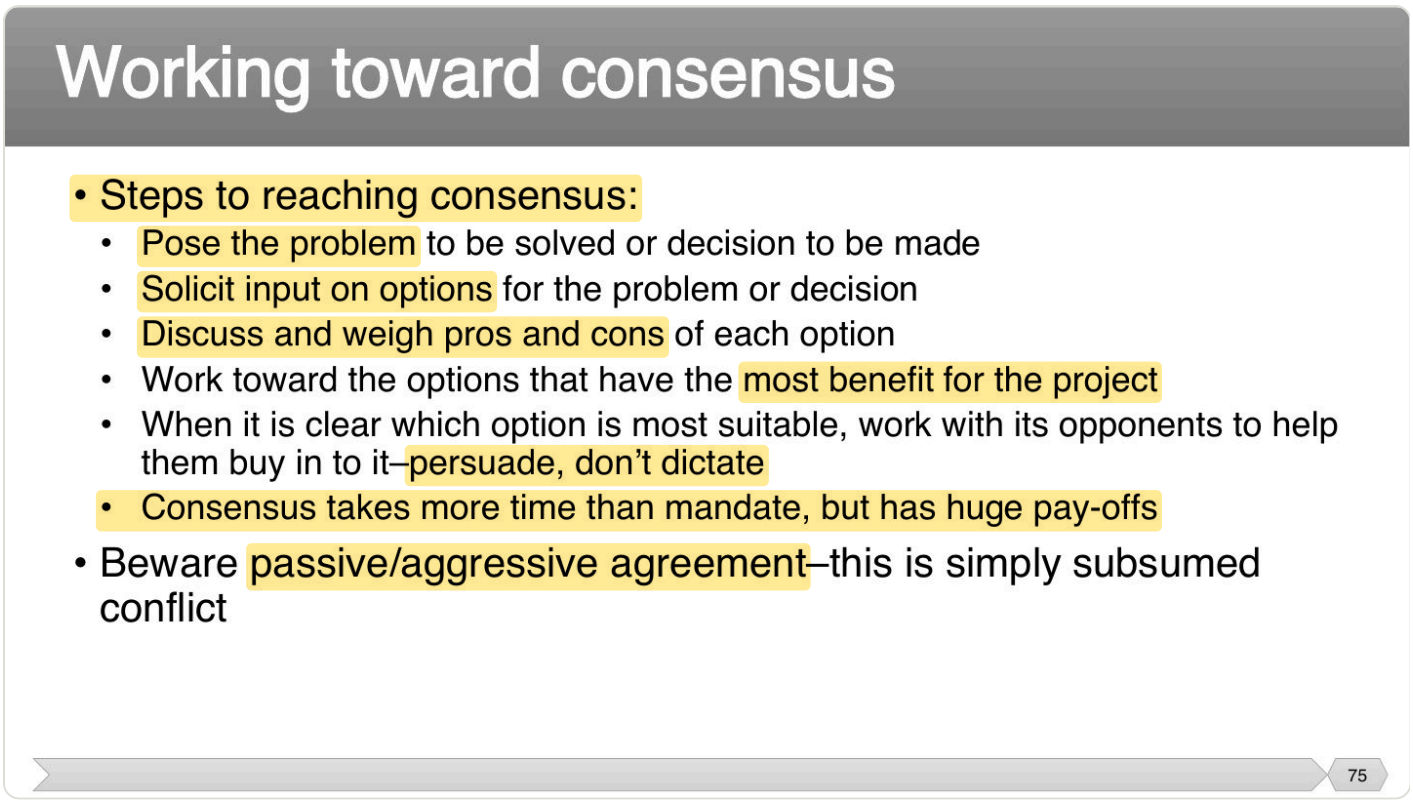
EXAM ANGLE

Past exam · Final 251 · Q5.1 Q1

Share an experience from a software project you were involved in where collaboration among team members significantly contributed to its success. Describe the specific strategies or techniques you employed to foster collaboration.

**Answer:** Any honest experience earns marks if it names concrete techniques. For example: in a course project building a booking app, we split the work by feature with clear owners, met twice a week for short stand-ups, tracked tasks on a shared board, used pull requests so every change was reviewed by a teammate, and paired up on the hardest integration work. Decisions were discussed until everyone had been heard, so we committed to them even when we disagreed.

 **Working toward consensus**




NOTES

1. Don't start with solutions: agree on the problem first, and collect options before anybody defends one. The order is what keeps the **discussion** from turning into a contest.
2. People who were simply overruled will resist later. Spending time to **persuade** now costs less than dealing with resistance during delivery, which is why the extra time pays off.
3. Beware **passive-aggressive agreement**: people who say yes but don't act on it. It is just hidden (subsumed) conflict.

TIPS & TRICKS

- The order matters here: **P**lease **S**it **D**own, **W**e **C**onfirm. Pose the problem, Solicit input, Discuss pros and cons, Work toward the best options, Confirm the choice.

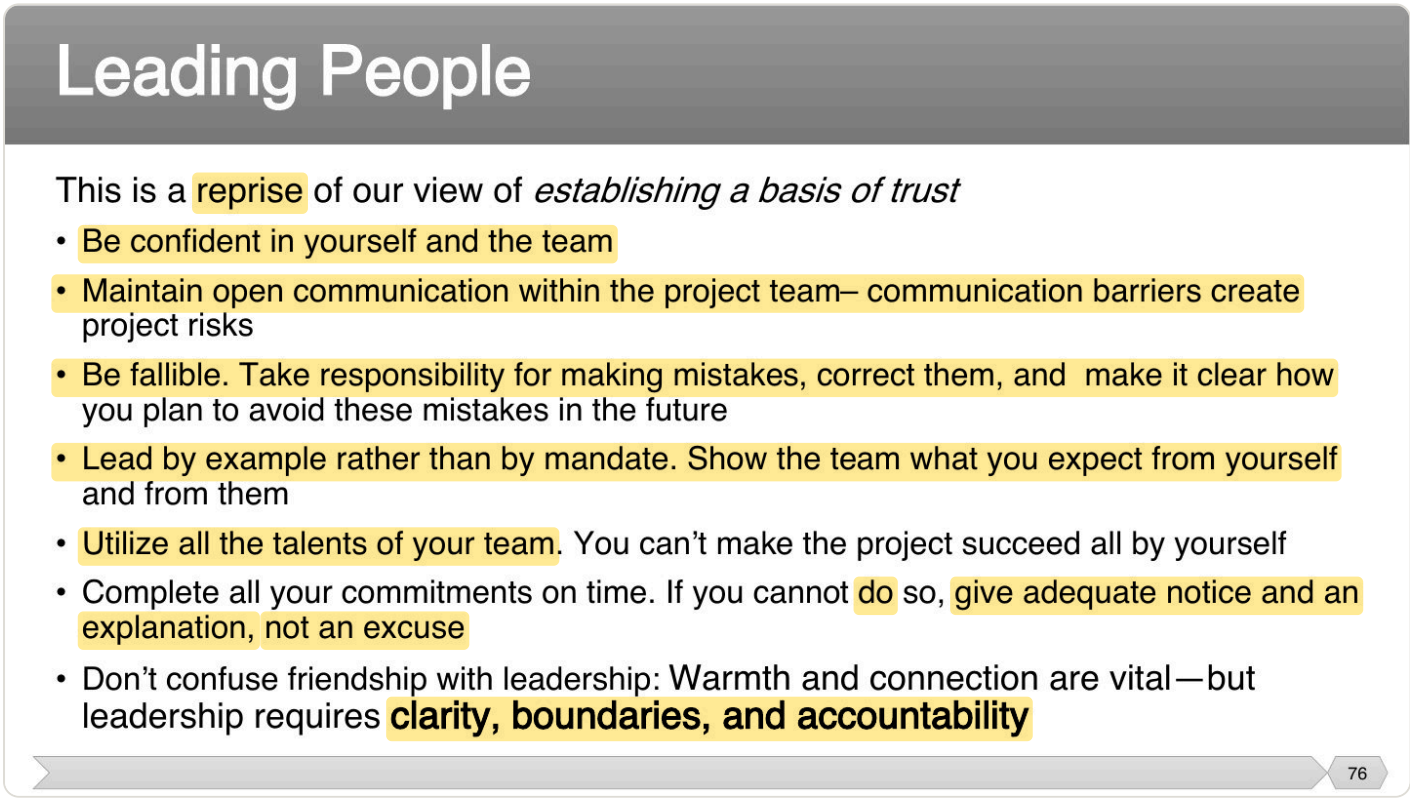
EXAM ANGLE

Past exam · Final 251 · Q5.2 Q1

You are leading a cross-functional team tasked with selecting a continuous integration/continuous deployment (CI/CD) platform for an upcoming project. The team includes software developers, DevOps engineers, and QA testers, each with distinct expectations regarding automation workflows and pipeline management. Describe your approach to collaboratively choosing the most suitable CI/CD platform.

**Answer:** Run the consensus steps. Pose the decision in a kickoff workshop where developers, DevOps engineers and QA testers each list their needs (build speed, deployment control, test automation). Agree the criteria and their weights together: integration with the repository and tools, automation and test support, pipeline flexibility, scalability, security, cost, ease of use and vendor support. Shortlist options, weigh each against the criteria (a weighted scoring table, or a short proof of concept), converge on the one with the most benefit for the project, then work with those who preferred others so they buy in. Don't just vote or dictate, and watch for passive-aggressive agreement.

SLIDE 76 / 77 **Leading People**




NOTES

1. The same habits that build trust are how you lead day to day. If a question mixes these slides, the answer is the same, as this is a **reprise**.
2. A nervous leader makes a team nervous. **Be confident**, without pretending you can do it alone, and you'll get more from people who feel their talents are being used.
3. A good relationship helps, but a friend may avoid saying hard things. You must sometimes give tough feedback, and that needs **boundaries**, even with people you like.

EXAM ANGLE

Prediction Q1

Why does a PM need to use all the team's talents?

**A** **You can't make the project succeed by yourself**

> That's the slide's reason.

**B** It removes the PM's accountability

> It doesn't.

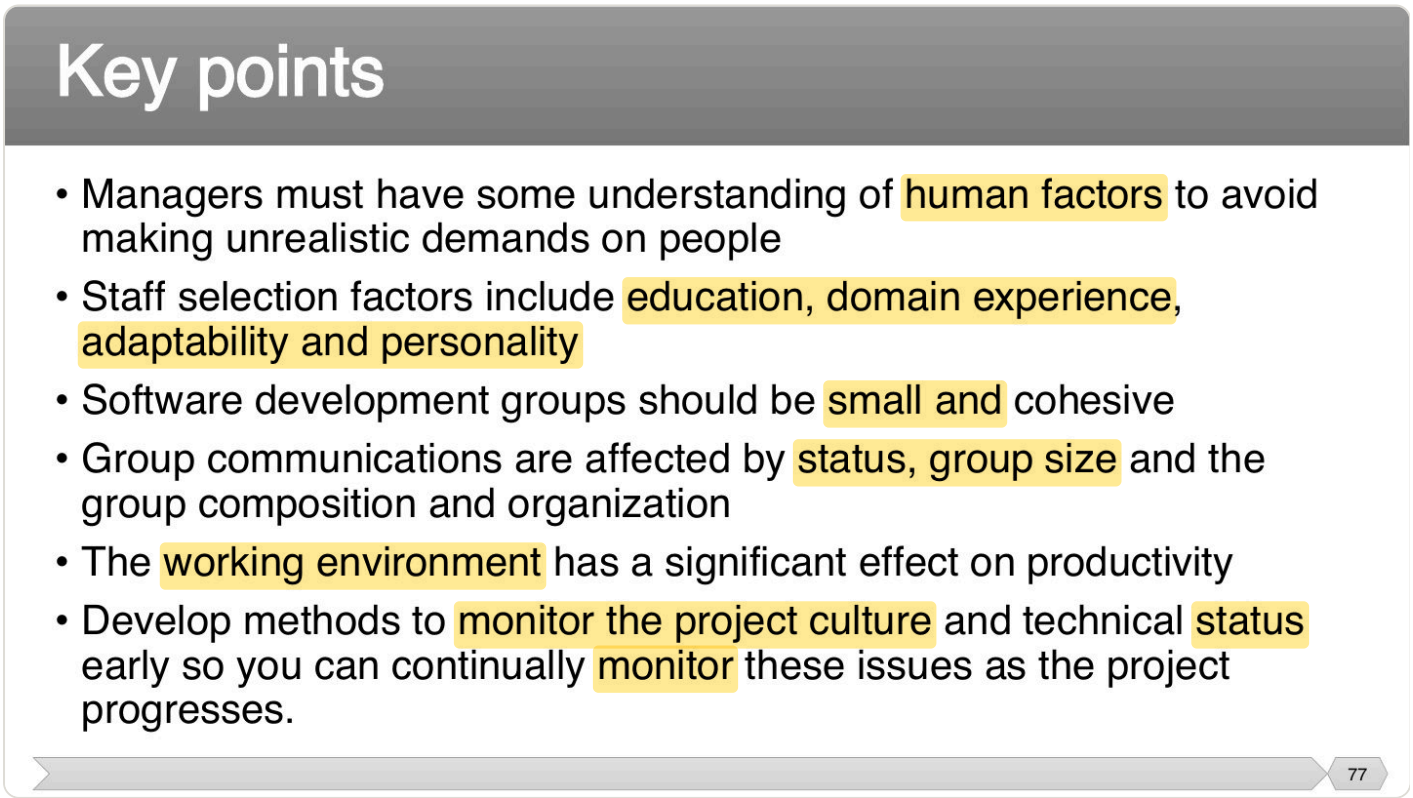
**C** It makes friendship unnecessary

> That isn't the point.

**D** It's required by the contract

> That isn't the reason.

SLIDE 77 / 77 **Key points**




NOTES

1. Asking for what people can't realistically do leads to burnout and resignations. Knowing the **human factors** tells you what's reasonable to demand.

2. A good exam summary of slide 53. **Domain experience** and adaptability often count for more than knowing a specific language.

3. The reason is communication paths: small groups share information easily. **Status** matters because juniors may stay quiet, even when they're right.

4. Don't wait for a culture problem to show up as a missed deadline. If you **monitor** early, you can act while it's still cheap to fix. Your environment also affects output every day.

---

## EXAM ANGLE

Prediction Q1

Why do managers need to understand human factors?

**A** To increase team size

> Groups should stay small.

**B** To reduce the team's pay

> That isn't stated.

**C** To avoid monitoring culture

> They should monitor it.

**D** **To avoid making unrealistic demands on people**

> That's the key point.

---

Original slides from the SE423 lecture deck, with study notes. Check exact wording against the original deck.
Notes and questions are study aids. Made by Shoug Alomran.