

How do I evaluate ~~Managers~~ Enablers?

Himanshu Jain
Last Updated: Apr 15, 2026

Manager ≠ Boss

Great Manager = Team member whose primary job is to help other team members (ICs) perform at their absolute best. Simply put, they are enablers.

The 7 Pillars of a Strong Enabler

[1. Leadership and Team Management](#)

[2. Communication](#)

[3. Technical Expertise](#)

[4. Execution and Delivery](#)

[5. Strategic Thinking](#)

[6. Process-Oriented](#)

[7. Results-Oriented](#)

[Red/Yellow Flags](#)

THE 7 PILLARS OF A STRONG ENABLER

Manager ≠ Boss • Manager = Enabler



Great Manager = Team member whose primary job is to help other team members (ICs) perform at their absolute best.



1. Leadership and Team Management

- Coaching and Development
 - Encourage and motivate engineers towards excellence - never through **fear or penalties**.
 - Act as a **coach and multiplier to the team**, not merely a supervisor.
 - Provide **frequent, actionable feedback**, not just during formal review cycles.
- Task Delegation
 - Delegate intentionally to **stretch capability**, not just distribute tasks.
 - Avoid over-delegation to high-performing individuals, which may result in burnout and disengagement.

- Avoid under-delegation to low-performing individuals, which may impede their development and negatively affect team performance.
- Technical Growth
 - Promote and organize technical sessions within the team.
 - Record these sessions and distribute the agenda in advance to ensure clarity.
- Team Attrition
 - Track attrition closely
 - Loss of high performers is a **critical red flag** and demands introspection.
- Enhancing the Company Culture
 - Zero tolerance for bias
 - Always extend the benefit of the doubt to team members.
 - Promote inclusivity and diversity within the team.
- Team Morale
 - Work Environment
 - Respect team members' working hours
 - Encourage flexibility in working hours, where possible.
 - Maintain **uncompromising standards** for the quality and quantity of output **without creating stress**
 - Address insecurity early—unspoken fear spreads fast.
 - Prevent the creation of a stressful working environment.
 - Remain receptive to input from stakeholders.
 - Foster openness and actively listen to suggestions and input from the team.
- Conflict Resolution & Ownership
 - Ensure every task has:
 - Clear scope
 - Clear ownership
 - Clear success criteria
 - Implement a well-defined & simple process and system for task management.
- Feedback & 1:1s
 - Constructive feedback
 - Conduct regular, **meaningful** 1:1s.
 - Focus on growth, blockers, motivation, and clarity—not just status.

2. Communication

Great leaders communicate, and great communicators lead – Simon Sinek

- Clarity is Non-Negotiable
 - Team
 - Individual goals
 - Team goals
 - Expectations
 - Feedback
 - Senior Management
 - Challenges
 - Risks

- Delays
 - Constraints
 - Stakeholders
 - Progress
 - Dependencies
 - Blockers
 - Risks
- Team Engagement
 - Help the team understand the rationale behind each KPI
 - Clearly explain
 - Company vision
 - Org vision
 - Team / squad vision
- Cross-functional Communication
 - Bridge gaps and align the team with broader business goals
 - Keep stakeholders updated weekly or fortnightly
- Team Engagement
 - Regular team outings
 - Cross-team and cross-tech discussions
- Thoughtful Team size & Team cost optimization

3. Technical Expertise

Don't be a People Manager, be a Techno Manager

- Strong understanding of:
 - Business metrics tied to the squad
 - Upstream and downstream dependencies
 - Capabilities and limitations (business and technical) of owned services
- Technical Ownership
 - Maintain a clear inventory of:
 - Technical documentation written by the team
 - Documentation authored by you as the EM
 - Conduct demo sessions for every major deliverable.
 - Ensure proper feature and KPI handovers.
- Engineering Quality
 - Track post-deployment issues.
 - Enforce strong engineering practices.
 - Use data—not opinions—for technical decisions.
- Decision-Making
 - Balance short-term delivery with long-term technical debt.
 - Prioritize based on impact, not urgency alone.
 - Informed, data-driven decisions
- Problem-Solving
 - Proactively remove technical roadblocks.
 - Continuously question: *Is there a better way?*

4. Execution and Delivery

The secret of success is execution, not merely knowledge – Eike Batista

- Consistent, predictable delivery
- Project Management
 - Maintain
 - Asana board for KPI
 - Central bandwidth and capacity planning sheets
 - Keep progress updated and visible.
- Prioritization
 - Balance:
 - Technical health
 - Business impact
- Resource Management
 - Plan realistically
 - Protect focus
 - Avoid chronic over-commitment

5. Strategic Thinking

Execution without vision is just passing time – Henry Ford

- Vision
 - Understand
 - Observe trends
 - Motivate others
 - Steer towards vision
- Innovation
 - Encourage experimentation
 - Explore emerging technologies (e.g., AI)
 - Identify missed opportunities early
 - Identify better approaches
- Risk Management
 - Business Goal
 - Technical Goal
 - Team Management
- Balance:
 - Business goals
 - Technical sustainability
 - Team health

6. Process-Oriented

A strong engineering team owns business metrics for the services it builds—and actively monitors them. Calling it “the business’s responsibility” is just an excuse.

- Well-defined and exhaustive Alerts
- Well-defined and exhaustive docs
 - Regular updation of docs
- Engineering Excellence contribution
 - SonarQube for code coverage, bugs, vulnerabilities, etc
- Monitoring
 - Service Dashboards
 - Quarterly improvements
 - Service Alerts
 - No. of OpsGenie alerts vs actionable alerts (reduce noise)
 - Monitoring of business metrics and not only technical metrics
- Sevs raised with Jira filter
- KTLO %, admin %, Feature %, Bug fixes %
 - Entire team
 - Manager
- Tempo compliance and accurate effort logging by team members
- AI enablement in the team and yourself
 - Concrete steps taken and quantifiable results obtained
- Sprints
 - Sprint velocity (variance +- 10%)
 - Sprint burndown
 - ticket completion in sprint (variance +- 10%)
 - Proper grooming of tickets before the sprint start
 - Proper estimates and rightsizing of tickets
 - original estimate vs actual efforts (variance +- 10%)
- Cost optimization
 - Targets
 - Strategy
- SLO improvements
 - Targets
 - Strategy
- ROI - cost vs GP/revenue
- Team performance
 - Commits
 - PR
 - Code Review
 - Lines of code written
- New onboarding plans and processes

Process is a balancing act.

Too much of it becomes micromanagement and kills engineers’ morale.

Too little of it removes guardrails and signals leadership laziness.

7. Results-Oriented

- **Metrics and KPIs**

- Quarterly KPIs with clear revenue or impact linkage.
- Measure **what changed**, not just what shipped (**Outcomes Over Outputs**)
- Ensure consistent KPI completion

Red/Yellow Flags

- **Red Flags**

- Micromanagement
- Excessive status tracking and housekeeping
- Persistent loss of high performers
- Negative skip-level feedback patterns

- **Yellow Flags**

- Over-promising and unrealistic bandwidth allocation
- Ignoring soft skills and team sentiment

PS: Perfection isn't the goal—progress is. Revisit this monthly to stay on track. This guidance is a living document: always evolving, always a work in progress.