

# How do I evaluate ICs?

Himanshu Jain

Last Updated: Apr 15, 2026

IC ≠ Developer; IC = Owner

An Individual Contributor is not measured by lines of code or tickets closed alone. ICs are accountable owners of outcomes, quality, reliability, and continuous improvement of the systems and business problems they work on.

This document exists to **standardize expectations across the org**, bring **clarity and fairness** to evaluations, and shift conversations from *activity* to *impact*.

## The 7 Pillars of a Strong IC

[Evaluation Philosophy](#)

[The 7 Pillars of a Strong IC](#)

- [1. Ownership & Accountability](#)
- [2. Execution & Delivery](#)
- [3. Technical Excellence](#)
- [4. Reliability & Operational Ownership](#)
- [5. Collaboration & Team Impact](#)
- [6. Communication & Responsiveness](#)
- [7. Growth, Initiative & Influence](#)

[What We Explicitly Track \(Inputs, Not Scores\)](#)

[Red & Yellow Flags](#)

[Red Flags](#)

[Yellow Flags](#)

[References](#)

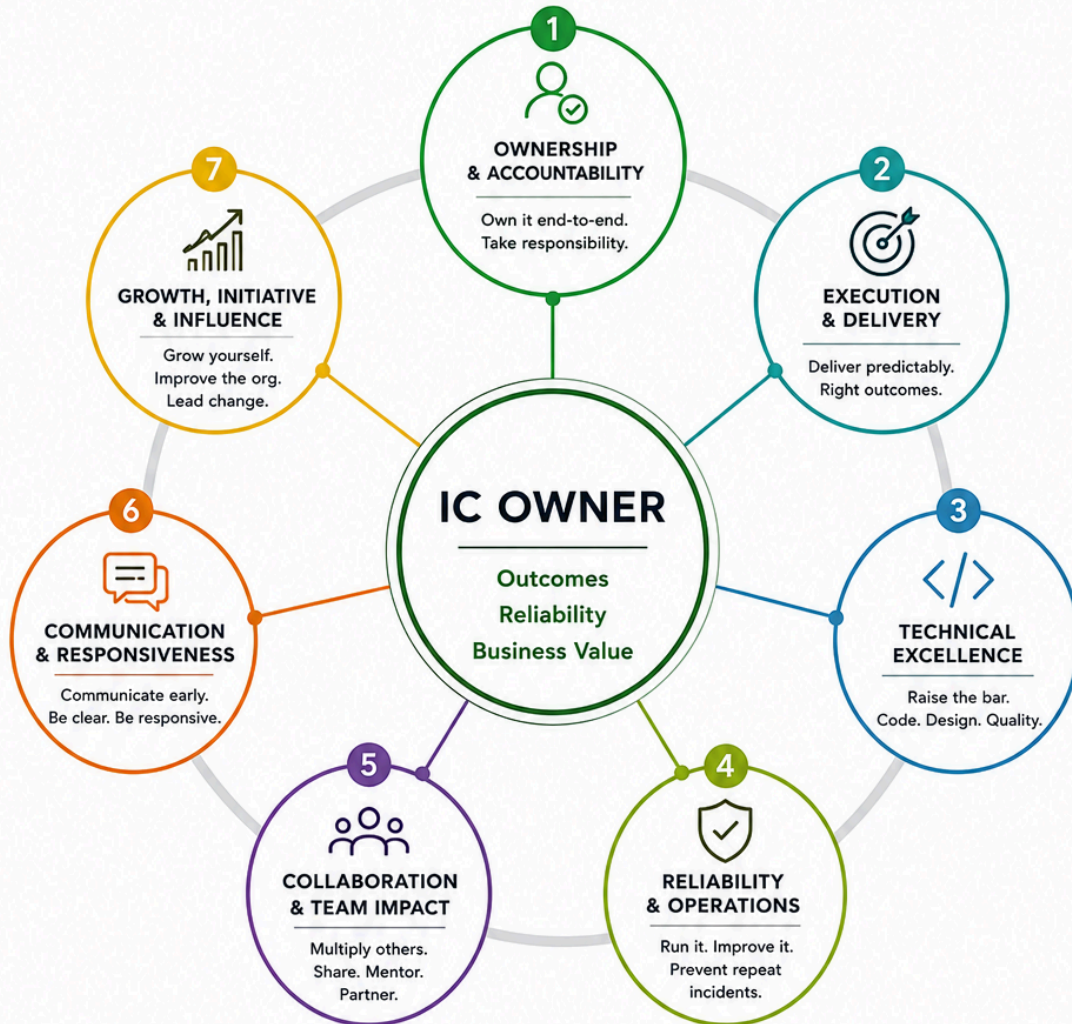
## Evaluation Philosophy

We evaluate ICs on **impact over output**, **ownership over execution**, and **signal over noise**.

Inputs come from multiple systems (Asana, Jira, JSM/On-call, SonarQube, engineering excellence, debugging capabilities, peer feedback, and business feedback), but **no single metric stands alone**. Context matters.

# THE 7 PILLARS OF A STRONG IC

IC ≠ Developer • IC = Owner



## WHAT WE LOOK AT (SIGNALS, NOT SCORES)



## FLAGS GUIDE

- RED FLAGS**
  - Repeated ownership gaps
  - Consistent quality regressions
  - Poor incident behavior
  - Negative peer feedback
  - Blame-shifting during failures
- YELLOW FLAGS**
  - High output but low collaboration
  - Over-indexing on delivery at the cost of quality
  - Avoidance of ops or reviews

## OUR PHILOSOPHY



**PERFECTION ISN'T THE GOAL—PROGRESS IS.**

This is a living framework. Revisit monthly. Keep raising the bar.

## 1. Ownership & Accountability

A strong IC:

- Owns services, features, and outcomes end-to-end
- Thinks beyond assigned tickets
- Takes responsibility during failures, not just successes

Signals we look at:

- Sense of ownership toward services
- Responsiveness to issues in gchat channels (e.g. incidents, business queries)
- Proactive follow-ups with cross-functional teams
- Quality of incident ownership and post-incident actions

## 2. Execution & Delivery

Delivery must be **predictable, reliable, and right-sized**.

Signals we look at:

- Jira story points completed with consistency (not heroics)
- Feature delivery weighted by complexity
- Sprint commitments vs actuals ( $\pm 10\%$  variance)
- KTLO, bug-fix, and feature balance
- Proper grooming and estimation quality

Anti-patterns:

- Chasing points over outcomes
- Chronic spillovers
- Over-commitment followed by burnout

## 3. Technical Excellence

Every IC leaves the codebase healthier than they found it.

Strong ICs raise the technical bar of the team.

Signals we look at:

- Code quality (SonarQube: bugs, vulnerabilities, coverage)
- Thoughtful design docs with depth and trade-offs
- Code reviews done (quality over quantity)
- Reduction in post-deployment issues
- Use of data to make technical decisions

#### 4. Reliability & Operational Ownership

Running production is a core engineering skill.

Signals we look at:

- On-call incidents resolved and acknowledged
- Quality of incident reports and learnings
- Improvements to alerts (noise reduction, actionable alerts)
- Monitoring dashboards created and maintained
- Reduction in repeat incidents

Anti-patterns:

- Alert fatigue without action
- Treating ops as "someone else's job"

#### 5. Collaboration & Team Impact

Red flag: Strong individual output with consistently weak peer feedback.

High-performing ICs multiply others.

Signals we look at:

- Peer feedback (Workday, cross-team inputs)
- Cross-team collaboration and follow-through
- Knowledge transfers (KTs) and mentorship
- Buddying new joiners
- Constructive participation in retros

## 6. Communication & Responsiveness

Silence is risk. Early communication is professionalism.

Clarity builds trust.

Signals we look at:

- Clear async updates (Jira, docs, Slack)
- Responsiveness to business and ops queries
- Ability to explain trade-offs and risks
- Timely escalation of blockers

## 7. Growth, Initiative & Influence

Strong ICs grow themselves—and the org.

Signals we look at:

- Initiatives taken beyond assigned work
- Tech talks delivered (with recordings)
- Documentation improvements
- Interview participation
- Exploration and adoption of better tools, patterns, or technologies (including AI)

Anti-pattern:

- Waiting to be told what to improve

## What We Explicitly Track (Inputs, Not Scores)

- Workday self-assessment and peer feedback
- Manager and PM feedback
- Jira delivery metrics (features, KTLO, bugs)
- SonarQube quality metrics
- Oncall incident data
- Documentation contributions
- Monitoring and alerting improvements
- Interviewing and hiring participation

These are **signals**, not a checklist.

# Red & Yellow Flags

## ● Red Flags

- Repeated ownership gaps
- Consistent quality regressions
- Poor incident behavior
- Negative peer feedback patterns
- Blame-shifting during failures

## ● Yellow Flags

- High output but low collaboration
- Over-indexing on delivery at the cost of quality
- Avoidance of ops or reviews

Flags trigger **coaching conversations**, not surprises.

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.