
expoQA®26
MADRID 26th, 27th & 28th May
expoqa.eu

AI Skills for Quality Guardrails

Making code review consistent, evidence-based, and automated

90 minutes • Interactive session

← → or click to navigate

1 / 16

The Problem

Same code, different reviewer = different outcome

"LGTM" on Monday, "needs rewrite" on Friday

"Improve error handling" — how? to what standard?

Quality depends on who reviews, when, and how much coffee they've had.

← → or click to navigate

1 / 16

Activity: Review Roulette

1. Silently review the code snippet (2 min)
2. Write down what you'd flag in a PR review
3. Pair up — compare notes (3 min)

What did your partner catch that you missed?

Review this code:

```
// welcomeService.ts — Sends welcome message to new users
function buildWelcomeMessage(userName: string, loginCount: number): string {
    let greeting: string;
    if (loginCount === 1) {
        greeting = `Welcome, ${userName}! Great to have you.`;
    } else if (loginCount > 1) {
        greeting = `Welcome back, ${userName}! Visit #${loginCount}`;
    } else {
        greeting = "Hello!";
    }
    if (userName.length > 50) {
        userName = userName.substring(0, 50);
    }
    console.log("Greeting sent to: " + userName);
    return greeting;
}
```

2 minutes — what would you flag?

What I Know / Want to Know

✅

I already know

about AI + code quality

🤔

I'm skeptical about

what concerns you

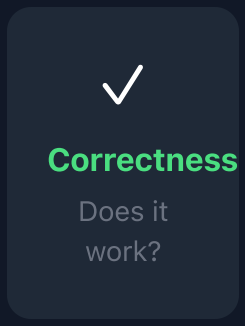
🎯

I want to learn

what you hope to take away

Add 1–2 stickies per column

Quality is Dimensions, Not a Feeling



Correctness

Does it work?

Security

Is it safe?

Readability

Can others understand?

Test Coverage

Is it tested?

Maintainability

Can it change?

Each scored independently. No halo effect.

← → or click to navigate

1 / 16

Mini-Exercise: Score This (3 min)

```
// bookingService.ts
function calculateTotal(
  nights: number,
  pricePerNight: number,
  discountCode: string
): number {
  let total = nights * pricePerNight;
  if (discountCode === "STAFF")
    total = total * 0.5;
  if (discountCode === "SUMMER")
    total = total - 30;
  if (total < 0) total = 0;
  console.log(
    `Booking: ${total} code ${discountCode}`);
  return Math.round(total * 100) / 100;
}
```

In pairs — gut feeling, 1–5 each:

Correctness	Does it produce right results?	____
Security	Is it safe from abuse?	____
Readability	Can others understand?	____
Testability	Easy to test?	____
Maintainability	Can it change?	____

Compare with your partner. Where do you disagree?

Scores Map to Actions

5	Excellent — fully meets all criteria	→ Merge
4	Good — minor gaps only	→ Merge
3	Acceptable — notable gaps	→ Merge with conditions
2	Poor — significant issues	→ Do NOT merge
1	Unacceptable — fundamentally fails	→ Block

A Skill is Just Markdown

```
---
name: quality-judge
description: Evaluate code against rubrics...
allowed-tools:
  - Read
  - mcp__gitlab__get_merge_request
---
# Quality Judge
## Workflow
1. Identify target (MR, branch, file)
2. Select rubric
3. Score each dimension 1-5 with evidence
4. Output: JSON + decision (MERGE / DON'T MERGE)
```

No code to compile. No package to install. A recipe the agent follows.

Evidence, Not Opinion

Bad

"Security feels weak"

Vague. What do I fix?

Good

"Line 13: user input concatenated into SQL query — injection risk"

Specific. Actionable. Verifiable.

Every score must cite a specific line, pattern, or behavior.

The Three-Layer Model

🤖 **Layer 1: Agent (quality-judge)**
Catches structural issues — mechanical, consistent, tireless

👤 **Layer 2: Human Review**
Catches what agents miss — wrong approach, bad tradeoffs, context

💥 **Layer 3: Adversarial Testing**
Catches what both missed — edge cases, compound failures

Not "set and forget" — layers of defense.

Exercise: Design a Rubric

Groups of 3–4 • 15 minutes

1. Define 4–5 dimensions that matter for YOUR code (5 min)
2. For your top dimension, define: what's a 5? a 3? a 1? (5 min)
3. Define ONE blocking rule: "If ___ scores ≤ 2, always BLOCK because ___" (5 min)

Challenge questions:
- Is it observable? Can you point at evidence?
- If someone scores a 2, do they know what to fix?

Exercise: Run the Quality Judge

Everyone at laptops • 20 minutes

Round 1 — Run it (5 min)

```
/quality-judge — evaluate this MR against code-quality rubric
```

Round 2 — Challenge it (5 min)

Do you agree with the scores? What would you override?

Round 3 — Custom rubric (10 min)

Feed YOUR rubric from Exercise 1 to the agent. Does it work?

Key Takeaways

1. Quality = **multiple independent dimensions**, not one feeling
2. Scores must **map to actions** (merge / don't merge)
3. Every score needs **specific evidence** — no opinion without proof
4. Agent = **consistent first pass**, human = final judgment
5. YOU define the rubric — **your standards, automated**

One Thing I'll Use Monday

"Starting next week, I will _____
because I learned _____."

- "...run /quality-judge on my MRs before requesting review"
- "...write a team rubric for our API endpoints"
- "...always check if feedback cites specific evidence"

"Quality guardrails catch the stuff you'd catch on your best day, even on your worst day."

The agent handles the mechanical.
You handle the meaningful.

Follow-up: 6-week Slack series starting next week

← → or click to navigate

1 / 16

expoQA®26

MADRID 26th, 27th & 28th May

Thank you for attending

expoqa.eu