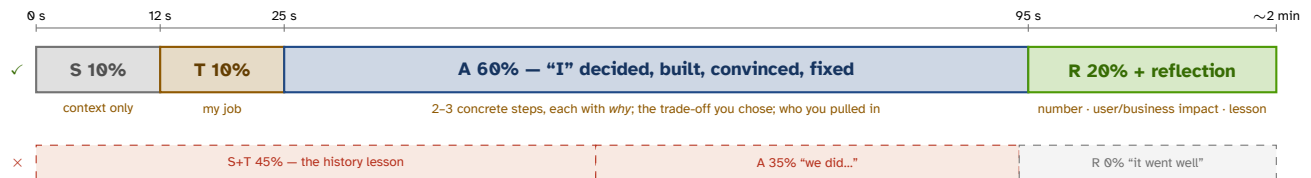


In one line: Every behavioural answer is a ~2-minute story: **Situation** (context, 1–2 sentences) · **Task** (*your* responsibility) · **Action** (what **I** did — the bulk) · **Result** (a number or concrete outcome) · then the **reflection** (what I learned / would do differently). They are not scoring the project — they are scoring *you*: ownership, influence, judgement, growth.



The rules that score

- **“I”, not “we”.** “The team” sets context; every *action* sentence starts with **I**: I measured, I proposed, I wrote, I convinced. Owning your part is not arrogance — it is the thing being assessed.
- **Quantify the Result:** crash-free 97.1% → 99.6 %, cold start 2.4 s → 1.1 s, build time –40 %, 3 teams adopted it. No number → a concrete outcome (“removed the weekly manual release step”), never “it went well”.
- **Close with reflection:** “Next time I’d involve QA a sprint earlier” — the seniority signal: self-awareness + behaviour change.
- **Land in ~2 minutes,** then stop and let them dig. Follow-ups are where the score is made.
- **Bank 5–6 stories** that each cover several questions; map the question to the nearest story instead of improvising.
- **Real, recent, yours:** last 2–3 years, stakes you can describe, details you can defend three “why?”s deep.

Senior signals vs anti-patterns

Signal	Anti-pattern
Ownership: noticed it, defined the approach, saw it through release + follow-up	“I was assigned X and I did X”
Influence w/o authority: data, prototype, shared goal	escalated / went over their head
Understood the other side, disagree-and-commit	blaming (“they were lazy”, ex-employer trashing)
Mitigate first, root-cause, blameless postmortem, fix the <i>system</i>	hero story (“20 hours alone”)
Real failure, owned, lesson, changed behaviour	fake failure (“perfectionism”) / someone else’s fault
Multiplier: reviews that teach, docs, mentoring	only your own commits
Structured, specific, measured	vague , no result, rambling 5 min

Interview traps

- ! Too much Situation: if you are still explaining the codebase at 45 s, you have lost the Action.
- ! A conflict story where you were simply right and they came round — no understanding, no compromise, no signal.
- ! Inventing numbers: they will ask how you measured it. Use a range or an honest proxy.
- ! “I have no questions” at the end — always ask 2 (team decisions, on-call, 6-month success, biggest risk).

Remember **S+T** short, **A** long and **“I”**, **R** with a number, end with the lesson — then stop.

Question bank — and what they really probe

Conflict with a colleague **probes:** empathy, maturity. Sought their view, found the shared goal, decided on evidence.

A failure / mistake **probes:** self-awareness. Real stakes (bad release, missed deadline), own it, what changed *afterwards*.

Disagreed with product / design **probes:** judgement + collaboration. Argue with user data or cost (“this animation costs 2 sprints”), offer an alternative, then **disagree and commit**.

Hardest bug **probes:** method. Repro → bisect/Instruments/logs → hypothesis → proof → fix + a test/guard so it cannot return.

Leading without authority **probes:** influence. RFC, prototype, metrics that moved another team (modularisation, CI).

Tight deadline / trade-off **probes:** prioritisation. What you cut (scope, not quality), the risk you accepted, who you told, the tech debt ticket you filed and paid.

Mentoring **probes:** multiplying. A named junior, what you did (pairing, review style), their visible growth.

Why this company / role? **probes:** motivation + homework. Their product, their stack/scale, your fit — never salary or “I need a change”.

STAR story template — prepare 5–6

Story name / tag:

Covers: ☐ conflict ☐ failure ☐ disagreement ☐ bug ☐ lead ☐ deadline ☐ mentor

S (1 sentence):

T (my responsibility):

A1 I

A2 I

A3 I

R (number):

Differently next time:

Follow-up I must survive (“how did you measure?”):

Likely prompts

1. Disagreed with a designer? — data, alternative, commit.
2. Biggest mistake? — real, owned, the guard added after.
3. Led without authority? — prototype, metrics, RFC.
4. Shipped under pressure? — scope cut, risk named.