EAP Coach & Program Manager Guide (Staff Only)

Purpose of this document

This guide defines the **professional posture, authority boundaries, and responsibilities** of coaches and program managers within the EAP project.

It is intended to:

- ensure consistent staff behaviour across cohorts;
- protect team autonomy and intern role ownership (including Product Owner and DevOps roles);
- safeguard assessment fairness and professional realism;
- support staff in observing Scrum and DevOps maturity across the full project period.

This document is **staff-only** and must not be shared with trainees.

Coaching philosophy

Coach ≠ problem solver

Coaches do **not**:

- design solutions for the team;
- prioritise backlog items;
- resolve technical problems for the team;
- take over operational responsibility during incidents.

Coaches **do**:

- ask clarifying questions;
- challenge assumptions;
- make risks visible;
- support reflection on decisions and outcomes;
- reinforce agreed processes and quality standards.

Principle: The goal is to develop **professional autonomy**, not dependency.

Role of the program manager

The program manager:

- safeguards consistency across cohorts and staff members;
- ensures assessment fairness and comparability;
- aligns project execution with program governance;

- supports coaches in difficult situations (conflicts, escalation, safety concerns).

The program manager does not interfere with day-to-day team decisions unless governance boundaries are crossed.

Authority boundaries that protect team ownership

This project gives the team real responsibility. Staff must protect that responsibility by keeping role boundaries clear.

Product Owner boundary (team-owned)

The Product Owner role is fulfilled by a **team member**.

Staff must not:

- reorder the Product Backlog;
- define acceptance criteria;
- decide scope on behalf of the Product Owner.

Staff may:

- ask questions about reasoning;
- highlight risks, constraints, or stakeholder impact;
- act as stakeholders requesting clarification;
- support the Product Owner in stakeholder communication without taking decisions away.

This boundary must be respected to ensure fair assessment and real Product Ownership.

Layered DevOps responsibility and staff authority

DevOps responsibility in this project is intentionally **layered**, reflecting enterprise environments.

Intern DevOps engineers – operational responsibility

One or more team members act as **DevOps engineers**. Their responsibilities include:

- maintaining CI/CD pipelines;
- managing deployments;
- monitoring systems and logs;
- responding to operational incidents;
- coordinating rollback or remediation with the team and Product Owner.

Intern DevOps engineers **operate the system** day-to-day.

Staff – platform and governance authority

Staff members act in platform-level roles, such as:

- cloud or infrastructure owner;
- CI/CD platform administrator;
- security and access authority.

Staff **govern the platform**, but do not operate it as part of the team.

Authority boundary (non-negotiable)

Staff must **not**:

- take over operational responsibility;
- fix incidents for the team;
- act as shadow DevOps engineers;
- silently correct problems before interns observe them;
- compete with intern DevOps engineers.

Staff **may**:

- apply platform or infrastructure changes (including without prior notice) when acting in a legitimate platform/operator role;
- enforce quality and security constraints;
- act as external forces affecting the system.

This boundary applies **at all times**, not only during incidents.

Interpretable change rule

Any staff-initiated infrastructure or platform change must be:

- diagnosable through logs, metrics, or pipeline output;
- reversible by staff;
- investigable by the team without insider knowledge;
- fully explainable during debrief.

If a change cannot be reasonably investigated by the team, it must not be introduced.

Observing Scrum maturity

Use the observations below to guide feedback and to support consistent assessment.

Sprint Planning

Observe whether the team:

- defines a clear Sprint Goal;
- discusses risks and dependencies;
- considers testing and CI/CD impact;
- negotiates scope based on capacity.

Red flags:

- planning as task assignment only;
- no discussion of uncertainty;
- ignoring quality implications;
- “we will see later” replacing decision-making.

Daily Scrum

Observe:

- transparency about progress and blockers;
- ownership of impediments;
- peer-to-peer communication (not only reporting to Scrum Master);
- follow-up on yesterday’s commitments.

Sprint Review / Demo

Observe:

- working software, not slides;
- honest discussion of what works and what doesn’t;
- evidence of quality (tests, pipeline, logs) where relevant;
- shared ownership (not one person presenting everything).

Retrospective

Observe:

- psychological safety;
- specific, actionable improvement points;
- follow-up on actions from previous retrospectives;
- willingness to discuss process, not only tasks.

Observing DevOps maturity (team level)

Key focus areas:

- consistent use of Git Flow;
- quality of Pull Requests and reviews;
- respect for CI/CD quality gates;
- reaction to failing pipelines (calm diagnosis vs blame);
- monitoring/logging usage (not only debugging by “guessing”).

Important principle:

- **Failure is acceptable.**
- **Bypassing quality gates is not acceptable**, unless explicitly approved under a documented exception (rare).

Coaching Interpersonal Conflicts

When Team Members Feel Unheard or Undervalued

Common Scenario: Team member's input was not followed in a decision, feels undervalued or not taken seriously.

Don't:

- ❌ Solve it for them ("I'll talk to the team" / "Let me fix this")
- ❌ Take sides in the conflict
- ❌ Minimize feelings ("Just get over it" / "It's not a big deal")
- ❌ Let it fester without addressing it
- ❌ Allow passive-aggressive behavior to continue

Do:

1. ✅ **Validate feelings:** "I understand you're frustrated. That's valid."
2. ✅ **Coach direct communication:** "What do you need to feel valued? Tell them directly."
3. ✅ **Enforce face-to-face communication:** "Have this conversation in person, not via chat."
4. ✅ **Give structure:** "Use: 'When X happened, I felt Y, I need Z'"
5. ✅ **Follow up:** Check if conversation happened and what the outcome was.

Why Not Via Chat for Emotional Topics?

For interpersonal or emotional topics, **insist on face-to-face** (or video call):

- Tone and body language matter critically
- Immediate feedback loop prevents misunderstanding
- Builds real human connection
- Prevents written misinterpretation and escalation
- **Professional skill they need:** Difficult conversations happen face-to-face in corporate environments

Exception: If there are safety concerns (harassment, intimidation, bullying), involve staff immediately and do not ask parties to resolve it themselves.

Coaching Script for Direct Communication

When team member comes to you with interpersonal issue:

> "I hear that you're feeling [emotion]. That's valid and I appreciate you sharing that with me.
>
> Here's what I want you to do: Talk to [person] **face-to-face**, not via Teams or Slack.
> Sit down together, and tell them:
>
> - When [specific situation] happened
> - I felt [specific emotion]
> - What I need is [specific request]
>
> Then ask them: 'Can you help me with that?'
>
> This is a skill you'll use throughout your career. Difficult conversations happen face-to-face.
> Try it, and let me know how it went."

What This Teaches

Professional Skills:

- **Self-advocacy:** Articulating own needs clearly
- **Direct communication:** Addressing issues, not avoiding them
- **Emotional intelligence:** Naming feelings and needs
- **Conflict resolution:** Not escalating, not avoiding, but addressing
- **Relationship ownership:** Taking responsibility for own relationships

This is more valuable than solving the problem for them.

Follow-up (Next Week)

Brief, informal check-in:

> "Hey [name], did you have that conversation with [person]?"

If yes:

> "How did it go? What did you learn?"
> [Listen without judgment]
> "Good that you did that. That's professional behavior."

If no:

> "What's holding you back?"
> [Listen for fears/avoidance patterns]
> "Do you want me to sit in as observer, or do you want to try it yourself?"
> [Usually they choose to do it themselves when offered support]

If it went poorly:

> "Okay, tell me what happened."
> [Listen]
> "Do you want to try again with different framing?"
> [Coach better communication approach]

Red Flags to Monitor

If team member consistently:

- Has difficulty accepting decisions they don't agree with
- Feels unheard even after being heard
- Struggles to work within constraints
- Avoids direct communication repeatedly

Then:

1. Observe if this is a **personal pattern** (vs. situational)
2. Document in confidential incident observations
3. Consider if EAP fit is appropriate
4. Have more direct 1-on-1 conversation:

   > "I'm noticing a pattern where you struggle with X. In professional contexts, this skill is important. How can we work on this together?"

Give opportunity to grow first. Most people can learn these skills with coaching.

When to Escalate to Staff Leadership

Escalate immediately if:

- Safety concerns (harassment, bullying, discrimination)
- Aggressive or threatening behavior
- Team member refuses to engage constructively after multiple coaching attempts
- Conflict is affecting entire team's ability to function
- Legal or ethical boundaries crossed

Do not try to coach through serious interpersonal violations. These require staff leadership intervention.

Intervention guidelines

Intervene when:

- safety, ethics, or respect are at risk;
- the team repeatedly ignores agreed processes or governance constraints;
- assessment integrity is threatened (e.g., staff are pushed to take decisions);
- technical or operational risk becomes unacceptable without visibility.

Interventions should:

- be minimal;
- focus on behaviour, boundaries, and decision-making process (not solutions);
- be documented briefly for context (what happened, what boundary was reinforced, what outcome followed).

Documentation expectations for staff

To support consistency and fairness across cohorts:

- record key observations per sprint (short bullet notes);
- keep notes factual (behaviour and evidence, not personality judgement);
- link observations to agreed expectations (Scrum, DevOps, professionalism);
- ensure that any incident-related interventions are logged privately and debriefed appropriately.

Final note

This guide exists to ensure **consistent, professional coaching** across the program.

> Staff authority exists to **govern and challenge**, not to solve.

END OF STAFF GUIDE