Engineering Progression Framework

The Big Picture

Our approach is built around a small number of core ideas that shape engineering progression at [Company Name]:

- We focus on the impact each engineer has on moving the business forward, rather than on long lists of specific skills and behaviours.
- Scope increases with seniority. As you grow, so does the complexity and size of the problems you are expected to solve. As the business grows, the engineering challenges should grow with it.
- This framework brings the most important behaviours and expectations together in a simple way. It is not meant to cover everything an engineer might do, but to highlight the meaningful differences between levels.
- Behaviours are intentionally broad so they reflect the many ways engineers can contribute. Our aim is to describe the level of technical contribution and mindset we expect at each stage, not create a checklist of tasks.
- We want to make space for the different ways people add value. This framework exists to support 1:1s, performance reviews and development conversations, helping managers and engineers talk clearly about scope, impact and alignment with our engineering principles.
- The framework focuses on the practice of engineering itself, not the specific tools or languages you use.

Levels are cumulative. Each level builds on the one before it. Engineers are expected to show the behaviours and technical habits of earlier levels as they progress, for example proactive testing at Engineer still applies at Senior Engineer.

Impact

Impact is the primary driver for progression at [Company Name]. Your impact reflects your sphere of influence and the contribution you make to our mission and goals. Different roles and disciplines create impact in different ways.

Focusing on impact allows us to recognise engineers who move things forward (and who “get things done”) without requiring them to perform specific routines. Impact grows through a mix of building technical skill, gaining real experience, learning from that experience and pairing it with the right behaviours.

Excellence here looks like:

- Choosing work that meaningfully moves your team, discipline and the business forward, in line with the scope of your role.
- Identifying opportunities to improve engineering outcomes and making them clear so the team can prioritise effectively.
- Consistently getting things done, contributing to the team and earning a reputation as a reliable, high-quality engineer.

Technical Skills

This section is about how you apply your technical ability and develop your craft. It describes the behaviours that show effective technical contribution at different levels of complexity and ambiguity.

We look at technical contribution through four lenses: quality, testability, performance and your ability to design and review systems.

Excellence here looks like:

- Your code and technical work are considered high quality by your peers and senior engineers.
- You can design systems that solve business problems efficiently and reduce ambiguity at both the technical and product level.
- You design with the right level of complexity, keeping things simple where possible.
- Your work is resilient, well tested and capable of scaling as the business grows.

Behaviours

Behaviours sit alongside technical skills and impact. They cover the core habits, mindsets and ways of working we expect engineers at [Company Name] to show.

Through your behaviour you set the tone for those around you. Great engineers role model great behaviours, and self-aware engineers know when they are setting a strong example.

Behaviours reflect consistent habits and intentional choices, rather than doing the right thing by chance or on “autopilot”.

How to Use This Framework

The purpose of this framework is to support clear, honest conversations about performance and progression between engineers and their managers. It outlines the expectations at each level, the type of impact we are looking for and the way we work as an engineering team.

This document is not a checklist or a standalone tool. It is a guide, not a set of rules.

Progression from one level to another requires a conversation with your manager and/or a vacancy to become available. Sometimes you may move up without having experienced every single skill or behaviour, provided you have shown strength and confidence in meeting the expectations of the next level. In other cases, a significant gap may prevent you from progressing.

For any promotion, engineers and managers should be able to clearly explain:

- How the engineer has demonstrated the level of impact required for the next step.
- How they have shown the technical skills AND behaviours needed to achieve that impact.

Associate Engineer

Scope Summary: A new engineer at the start of their career, focussed on expanding their knowledge and skill set to further increase their impact within a team.

“Associate Engineers” are at the first stage of their software engineering career. They’re contributing to the team’s goals through executing on well-defined tasks, with support from more experienced engineers. As someone early in their career, they’re focussed on increasing their Software Engineering skill set and knowledge at a high pace. So that they can start to deliver tasks with less support and progress towards “Engineer”.

Impact

Delivers well-defined tasks, with support from more experienced engineers.

Improves their Software Engineering skill set at a high pace. Focussed on accelerating their knowledge and mastery of their relevant programming language, data structures, common algorithms, development environment, version control and internal systems.

Technical skills	**Behaviours**
Writes code and documentation in-line with our engineering principles. With support from other team members. Starts to demonstrate understanding of common Software Engineering concepts. Works to develop their usage of tooling and systems, including their development environment, source control, and internal systems tooling.	Proactively seeks out feedback and takes action to accelerate their learning. Proactively communicates progress and blockers when delivering tasks. Escalating quickly if stuck. Invests in their own learning to better serve their team. Demonstrates an ability to self-serve in seeking out new knowledge, with guidance from more experienced engineers. Applies learnings so that similar tasks require less support over time.

Engineer

Scope Summary: An engineer with high potential, developing the fundamentals they need to be successful in their team.

"Engineer" engineers are in the earlier stages of their software engineering career. They're working on well-defined tasks and are supported by the team when stuck. They're expected to ask lots of questions. As someone progresses toward "**Senior Engineer"** they should start to gain confidence to pick up larger or less well-defined tasks with less required support.

Impact

Delivers well-defined tasks, in collaboration with more experienced engineers	
Technical skills	**Behaviours**
Writes well tested code and documentation in-line with our engineering principles. Able to safely release code to production in collaboration with more experienced engineers. Demonstrates understanding of [Company Name] system architecture. Debugs and resolves issues related to their code changes in production.	Proactively learns from the work of others through activities such as reading & commenting requests and proposals the team produces. Proactively communicates to their team what they are working on, why, how it's going and what help they need. Asks questions to understand how to write effective code and prioritises their own learning to better serve their team. Proactively improves the test coverage and documentation of existing code.

1 x Lead Software Engineer
1 x Software Engineer
1 x Associate Support Engineer
1 x Systems Engineer