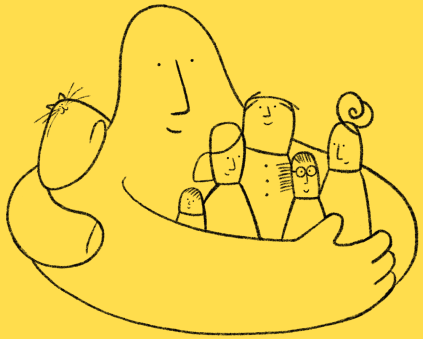
Engineering progression framework



Farewill

Introducing our engineering progression framework

✏️ **In 2020 we decided to create a progression framework for Engineers at Farewill**
We're using this to have shared expectations across the company around what we expect of engineers at different levels. It'll help people in planning their career growth, will act as a communication aid between engineers and their managers, and will also help folks to give appropriate feedback during reviews. We'll also use it more widely, for things like assessing candidates' seniority as part of our hiring process, and making salary offers at fair levels.

📊 **This document aims to help us describe what's expected at different levels**
Each level comes with a description, and an illustration of the type of behaviours, impact, and skills we think are reflective of someone at that level, but importantly, *it's not an exhaustive checklist*.

We've intentionally focused on a core set of examples that we think can fairly apply to any engineer at Farewill, but they're not intended as an finite list of everything a great developer could do or be. You'll almost certainly be doing important things that aren't in the framework. There are many 'shapes' of engineers, and we'll aim to celebrate people's different strengths whilst also aiming for fairness and clarity through our core expectations.

If you're a Farewillian, we've got more detailed information on the Notion page (internal only)

Parts of the framework

What you do

 Impact

Your ability to maximise the impact you have – making sure you're tackling the most important problems at each level, in the right way, to a sufficiently high standard.

How you do it

 Knowledge

The breadth and depth of your technical knowledge and skills, as well as domain and business-specific context you've gathered.

 Communication

The quality of your written and verbal communication to technical and non-technical audiences. This includes awareness of how your tone, chosen medium, and context can impact outcomes.

Leadership

Your influence in engineering and the wider organisation. This includes organising collaborative change, making the world better for those around you and giving others a platform to be great.

Junior Engineer

You deliver smaller, well-scoped pieces of work

This is the beginning of your engineering career, so other people will support you as you learn.

Day-to-day you focus on small, contained tasks to gain confidence as an independent engineer.

You're curious and ask for support when you need it, which helps build up your engineering toolkit and foundational skills.

⏱️

Everyone's different, but as a guide we'd expect someone to take around **1 to 1.5 years** to progress through this level.

Junior Engineer

Impact

- Work independently on smaller, well-scoped tasks and bugs to help your team deliver high-quality work on time. (e.g. add an email to notify users when they change their password)
- Contribute to quarterly planning (e.g, ask clarifying questions about objectives or key results)
- Build relationships with stakeholders (e.g. demo features, gather feedback, notify about the release of a feature)
- Proactively scope out and create tickets for small bugs or improvements.

Knowledge

- Set up and maintain your own dev environment (eg. install versions of Node, get your editor set up, and get applications running following readmes).
- Write code that matches the existing code style and conventions, so the rest of the team can work with it.
- Write automated tests for your code, matching the style and coverage in the codebase.
- Use version control effectively.
- Debug issues with others, using monitoring, metrics and logs. (e.g. pair with someone to find out more about a slow endpoint using Honeycomb, or look into logs to find more context around an error)
- Understand the purpose of the tools your team uses (e.g. the database used by an application, how to keep track of bugs)
- Join user research sessions to better understand how your work relates to user needs and product vision.

Junior Engineer

💬 Communication

- Ask for help when stuck, clearly communicating the problem and what you've done so far.
- Ask clarifying questions to remove ambiguity.
- Get involved in the PR review process (e.g. ask questions to learn, point out syntactical errors).
- Write PR descriptions that provide basic context and fit our standard format.
- Communicate work progress and blockers clearly to your teammates.
- Contribute to documentation.
- Seek out feedback (e.g. on a PR, ask your manager for growth areas, after running small meetings).

💛 Leadership

- Run small team meetings effectively (e.g. product team retro, team social event).
- Fix mistakes or inaccuracies with documentation to help others who could need it in the future.
- Participate in a small way to guild-level activities (e.g. suggest topics to discuss, give an Eng 10 talk)

Mid Engineer 1

You deliver larger, less well-scoped pieces of work

At this level you're comfortable with what you know and what you don't know. You deliver great work independently for your team. And you seek out opportunities to fill in your knowledge gaps, broadening and/or deepening your technical skills and experience.

You're also mindful that success as an engineer isn't entirely reliant on your technical skills. You work on other core skills, and start to be more involved in the wider Engineering Guild.

⏱️

Everyone's different, but as a guide we'd expect someone to take around **2 years** to progress through this level.

Mid Engineer 1

Impact

- Work independently on larger features and bugs to help your team deliver high-quality work on time (e.g. adding an address lookup tool to the site).
- Help sprint planning run well by suggesting work and goals for the team.
- Help the team work faster and more effectively by unblocking other engineers (e.g. notice, and pair with someone who's stuck).
- Identify ways to work more efficiently (e.g. improve a manual chore with automation or by adding documentation)
- Look for the best way to solve problems (e.g. improve on solutions suggested by others when writing cards and talk through options with stakeholders)

Mid Engineer 1

🧠 Knowledge

- Write code that shows you're comfortable working independently with the primary languages or technologies you use day-to-day.
- Test your work in a variety of ways.
- Debug production bugs, errors and alerts and work with others to fix them. (e.g. review a Sentry alert, reproduce the issue, discuss solutions with another engineer then implement a fix together)
- Implement monitoring and logging in the code you write. (e.g. add request method logs to help with debugging, or measure timings on a performance-sensitive script)
- Show an awareness and consideration of good engineering practices and key non-functional principles that relate to the domains you work in, for example:
  - Model data appropriately, applying sensible structure, types, validation, indexes etc. (things like independently writing migrations to add new tables, or defining new document schemas).
  - Use accessibility principles (like making sure forms on our site can be understood and completed using screen reader and keyboard).
  - Improve application performance (like reducing redundant re-renders, avoiding N+1 queries, using async code or task queues, and writing effective queries).
  - Keep our data well-structured, robust, and safe from accidental errors (e.g. effective form validation).
  - Defend against security risks (e.g. code and practices don't introduce OWASP Top 10 risks).
  - Protect data integrity, privacy, security and retention. (e.g. using dummy data in staging, limiting access for new data sources, using atomic processes to ensure a good data state).
- Consider company and team goals when scoping, making technical decisions, or choosing solutions.

Mid Engineer 1

💬 Communication

- Write PR descriptions that give context on any bugs you faced, the tradeoffs you considered and decisions you made, in line with our standards.
- Review code in line with our standards.
- Give constructive feedback that's actionable and kind (like this example).
- Explain technical concepts to your team members and stakeholders so everyone can understand.
- Contribute to team discussions (whether written or spoken), and encourage others (especially more junior) to contribute too.
- Feed back what you've learnt to your team.
- Show empathy towards your colleagues and communicate in a no-blame way.
- Contribute to the creation and context of bug postmortems.
- Constructively challenge others in a kind way.

💛 Leadership

- Take part in pair programming interviews. Give balanced feedback on scorecards and make sure the candidate has a great experience. Mark interview code tasks fairly, giving balanced feedback.
- Participate in Engineering Guild activities (like giving a Tech Talk or putting forward a tech proposal)
- Support and encourage others to give talks and share knowledge.
- Create and maintain documentation on things you know the most about, making it easy for future engineers to interact with those systems/code.
- Ask why, flagging instances when you think we could make better product or technical decisions.

Mid Engineer 2

You lead on the design and implementation of a typical project

The team can rely on you to guide a typical engineering project from start to finish. You may not know everything, but that's ok because you draw on past experiences and have the skills to solve problems effectively.

Your work is high quality, you deliver what's needed, and draw on your knowledge of good practice and non-functional principles. You also share what you've learnt with others, whether that's more junior members of your team, or at Guild Level.

⏱️

Everyone's different, but as a guide we'd expect someone to take around **2 to 3 years** to progress through this level.

Mid Engineer 2

Impact

- You lead technical implementations of projects to help your team deliver high-quality work on time (e.g. Standalone Lasting Power of Attorneys, Search Engine Optimisation platform work, improving Wills Ops tooling)
- Support your team to work more effectively by helping to break down and coordinate work. Spot areas of conflicting work in the team, and know how to get support to fix these.
- Make sure work isn't blocked by giving people the technical skills and context they need to make progress (like sharing project details with another engineer if they haven't worked on it before).
- Improve engineering practices outside of your team (for example, bring ideas to improve testing to a Guild meeting).
- Remove distractions for the team (like taking on urgent work to let others focus on a project).

Mid Engineer 2

Knowledge

- Support your team's goals by working on features of any size across multiple codebases, as needed.
- Use techniques (like story mapping or MoSCoW prioritisation) to help your team estimate tasks and prioritise work.
- Highlight product decisions early if they might have complex technical or legal consequences.
- Build in controls to make sure we have a good sense of the quality of code you're responsible for. This could include things like:
  - Implementing or improving observability in applications you work on (e.g. adding a health-check, or setting up dashboards and alerting around a new product/feature)
- Increase your depth and breadth of engineering good practice and non-functional principles, to solve problems at a more complex or abstract level. That could include things like:
  - Putting good data practices in place (like applying encryption at rest, configuring appropriate backup/retention policies, filtering/masking logs, implementing access controls, and adding audit logs if appropriate
  - Finding and fixing problems with security and performance debt. (like adding new Express middleware that makes it easy to enforce access controls, or improving performance of slow endpoints)
- Keep up to date with best practices and recent technology changes in your area.

Mid Engineer 2

💬 Communication

- Communicate the pros and cons of different problem solutions to your team.
- Give valuable input to proposals from the team.
- Create diagrams and learning materials for your team's projects. [like this example]
- Share what you've learnt with the Engineering Guild.
- Set up postmortems and identify actions to avoid the same problems in future.
- Be empathetic towards your teammates and our users.
- Try to understand the views of others before defending your own.

💛 Leadership

- Help with onboarding (e.g. be an engineering buddy for a new team member in their first month).
- Help others in your team progress, for example through:
  - Involving them when they're needed for a piece of work.
  - Mentoring more junior engineers (e.g. finding suitable work for them, pairing on tickets, giving context and guidance).
- Identify areas of improvement for your team, suggesting and implementing changes.
- Support your team in learning about and implementing best practices and technologies, including sharing your knowledge of non-functional principles.
- Be involved regularly with the Engineering Guild, for example by:
  - bringing up important issues
  - encouraging others to voice opinions
  - collaborating on proposals
  - instilling our engineering principles in other engineers (like sharing an example of a great collaborative code review).
- Ask 'why' in more challenging situations, and don't take assumptions at face value unless you understand exactly where they're coming from (especially with regards to regulation, compliance, etc).
- Lead well-structured meetings that run to time and have tangible outputs.

Engineer and Engineering Manager tracks split here

Senior Engineer

Your leadership and impact directly contribute to the success of your team, and those around you

At this stage of your career, leadership comes in many forms. You may be a hands-on domain expert leading by example in your work, working with an architectural focus, or being a tech lead for a team. Whatever you do, you do it in a way where your impact is felt substantially by the people around you. It's no longer just about your own output, it's about having impact on a wider scale.

This is a step up in responsibility and expectations, it's not easy to reach this point, and you should celebrate!

At this level your journey becomes incredibly individual, but we'd expect someone to take around **3 years** to progress through this level.

While your day-to-day focus is mostly at team level, you can work across teams too (e.g. when more complex collaboration and coordination is needed, or as a trusted person to jump in for something urgent).

You're a respected member of the Engineering Guild and work to make things even better, particularly representing the business area or technologies you work closely with.

At this point you're well equipped to make decisions and recommendations, and get buy-in. You work to change situations instead of expecting others to.

Senior Engineer

Impact

- **Help your *team* deliver great work** through things like:
  - using your domain expertise to write code for more ambiguous and challenging technical problems the team need to solve (e.g. the SCA project).
  - working to architect solutions and get the team's support for them.
  - setting the team's technical vision.
- Improve your team's efficiency and effectiveness by getting buy-in for work like tackling specific technical debt or automating manual tasks.
- Help your team create more robust technical solutions by balancing current and future needs.
- Help your team make the best decisions by running discovery or experiments on providers and software.
- Have the toolkit to work on more complex and unknown problems under pressure or at speed (like fixing a business-critical incident).
- Propose and get buy-in for projects that would help Farewill achieve its goals.
- Help us do new things by leading on longer-term technical improvements or projects across quarters (e.g. migrating away from a library, or improving test coverage).

Senior Engineer

🧠 Knowledge

- Are seen by others as an authority in your technical domain, or on your team's systems.
- Understand and shape the technical direction and vision for your team's systems.
- Lead on technology improvements both in your team and the problem space that they work in (ideally leading to positive effects wider than the team).
- Make sure your team is using suitable engineering practices, helping them to adopt new and emerging best practices and tooling where appropriate.
- Spot technical knowledge gaps in the team and find ways to fill them.
- Help *groups* of others (in your team or more widely) to improve their knowledge and skills of technology best practices and non-functional principles. This could be done through improving standards and tooling, like:
  - Implementing best practices for observability in applications and processes using Honeycomb (after picking an appropriate technology).
  - Defining our cross-device standards, associated testing strategy, and getting buy-in for a physical device lab or virtual services.

Senior Engineer

💬 Communication

- Manage up effectively (e.g. give your manager feedback, maintain a brag doc, own your development and goals).
- Explain technical concepts to non-technical stakeholders in the wider team.
- Coordinate and communicate across teams to standardise code and practices.
- Get buy-in from stakeholders and teammates, and encourage transparent communication when you're leading on larger pieces of work.
- Get buy-in for technical solutions.
- Nurture empathy, psychological safety, and a no-blame culture on your team and within the guild.
- Give opinions in a way that invites participation from others and explains your context or thinking.
- Can fix technical disagreements and make decisions (like asking questions that dig into why, and de-escalating opinion X vs opinion Y arguments)

Senior Engineer

💛 Leadership

- Help engineers on your team to work more safely and effectively by leading process improvements (like adding pull request templates or improving how we respond to alerts).
- Find ways to improve the technical quality of the team's output.
- Lead changes to engineering practices with well-reasoned arguments, but be open to counter-suggestions.
- Improve our candidate experience for hiring new team members (like improving a pair programming interview or updating a hiring task).
- Work with Engineering Managers to help other engineers perform and grow, finding learning opportunities for others to hit their goals or development targets.
- Delegate low-risk technical decisions that we can change if we need to, and own high-risk technical decisions that are harder to change.
- Find mentoring opportunities with junior and mid-level engineers.
- Break down delivery and knowledge silos in your team.
- Get people excited to work here by promoting our products and culture, and representing us in writing, speaking or open source contributions.
- Help maintain and evolve the Farewill culture in the wider company.
- Know when to bring together different functions (including external parties) to solve tricky problems, and help make these meetings or initiatives effective.
- Lead by example, gaining respect from your team for both your technical skills and values.
- Find ways to solve problems, rather than expecting others to solve them for you.

Staff Engineer

You have day-to-day responsibility working across a set of different teams, problem spaces, and systems. People feel your impact in those teams, as well as at Guild and company level.

You have oversight across multiple work streams and areas of context, you're known company-wide as a technical authority in your area, and you're operating at a level that's highly strategic. You see opportunities, and know how to make them happen.

Stories about being a staff engineer in other companies

⏱️

At this level your journey is still incredibly individual, but you're likely to spend at least **3 to 5 years** here before making the next leap

ℹ️ As we grow we may decide to introduce a new level after this (Senior Staff Engineers)

Staff Engineer

Impact

- Help us to tackle the company's most ambiguous, strategic, and challenging technical problems, driving architectural decisions and systems design that affect multiple teams.
- Bring about engineering improvements that benefit multiple teams, both technically and through how we work. These should cover areas like quality, performance, observability, stability, and scalability.
- Give the business confidence that we can meet our high-level product strategy and timings, working closely with members of the Product Guild.
- Champion the importance of engineering-led improvements and investing in less-visible work, helping us to prioritise accordingly.

Knowledge

- Are comfortable solving any technical problem in your area, whether or not you've experienced it before.
- Maintain a technical vision of how our products and services work together in the domain you work across.
- Set technical strategy in your area.
- Identify and mitigate against architectural decisions and tech debt that could slow us down or cause problems, before they do.
- Have an in-depth understanding of the legal/product market we work in and how that informs tech decisions and strategy.
- Propose company-wide technical strategies to move towards business goals and get buy-in to do them.

Staff Engineer

💬 Communication

- Can communicate with lots of different people in different ways, including:
  - sharing future plans and direction to non-technical stakeholders.
  - adjusting when speaking to more junior teammates, to be mindful of the imbalance of seniority and experience
  - being relied on to explain technical concepts effectively at high-pressure times (like to partners or investors externally).
- Spot engineering factors affecting team health in your area and help improve them.
- Consider solutions based on strong data and reasoning, and identify potential flaws, risks, and unknowns.
- Foster environments that lead to collective problem solving, instead of defending individual solutions.
- Can be challenged by anyone in the Guild.
- Can calmly influence, organise, and lead others during company emergencies or critical incidents.
- Get buy-in for the importance of less-visible technical work, especially if people have challenged or underappreciated this work before.
- Raise and fix any systemic problems of inequality or unfairness you find in engineering.
- Discuss and make decisions based on informed experience, but stay open to other solutions.

Staff Engineer

💛 Leadership

- Coordinate timely, successful company technical projects, giving team members opportunities to grow (you delegate effectively, and know when to involve others).
- Attract other very senior hires, and can sell Farewill as part of the hiring process.
- Sponsor and mentor other more senior engineers, ideally helping them to grow their influence and network outside of Farewill.
- Help Farewill become more respected in the wider technical community (like finding opportunities for the team to make their work public, speaking to large audiences, sharing knowledge).
- Consistently improve our approach to systems design across the company.
- Lead change when you see a need, without anyone asking you to.
- Maintain high quality and standards company-wide, by championing our standards and principles and helping others to learn about them
- Be a confident leader of engineers, even across technologies that you don't have lots of first-hand experience with.

Principal Engineer

You have responsibility across the entire company, and every team feels your impact

ℹ️ This isn't a role that we need yet, as the scope we're operating at can be covered at Staff Level. But as we grow, we'd expect people to move into this level.

This is the most senior level for someone on the engineering path at Farewill – reflected by the scale of your responsibilities and expectations. By this point in your career you'll have a deep specialism, whether that's in a particular technology (or across a set of technologies), or with context that's unique to our business.

Your remit is across all current engineering work, but you'll also set us up for success in the future.

👩🏽‍💻 Impact

- Have responsibility for company-wide technology decisions, systems health, engineering practices, and risk, across our entire codebase.
- Set the company up well for the future, considering possibilities that could be a number of years ahead.
- Support business goals by identifying where technology can help us.

Principal Engineer

🧠 Knowledge

- Are an overall authority on the company's architecture.
- Develop the business's multi-year technology strategy, tying it into the overall business strategy.
- Can pinpoint engineering problems at the root, and successfully bring about changes that touch every level of engineers.
- Define or update company-level engineering principles and standards.

💬 Communication

- Have enough oversight of engineering as a whole to spot duplication of effort, or conflicting approaches, and can bring people together to resolve these.
- Comfortable communicating at board level when needed.
- Can communicate difficult decisions where needed, and bring these to a successful conclusion.

💛 Leadership

- Can influence and bring about change across the Engineering Guild, without needing reporting lines.
- Develop other people as engineering leaders for the future.
- Help attract and hire our most senior engineering/engineering manager roles.
- Foster a sense of inclusion and community in engineering as a whole.
- Can get people excited and bought into the technical vision that you've set.

Executive role

There's a theoretical path to being on the Executive team, but in reality this is unlikely. It would depend on company needs (whether there's a role in the first place), and we'd also run a competitive process with internal and external candidates. So it could be part of your path, but you shouldn't see it as a natural part of progression.