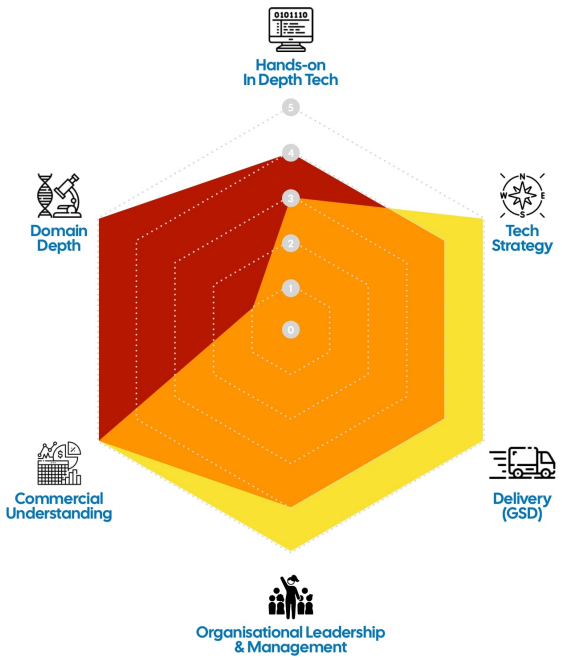
The Career Vectors Framework

See where you stand, and what to do next.

Meri Williams



Created by *Meri Williams* and presented by *Build Your Edge*.

Watch the full episode

This handout complements Meri's *Build Your Edge* episode, "The Six Skills Every Technical Leader Must Master". Start by listening to the conversation, as it brings these ideas to life and provides the context to get the most value from this guide.

Contents

Introduction

The Career Vectors Framework gives you a practical way to understand what a role really demands, and whether you're the right fit for it today.

Instead of focusing on titles or ladders, it maps the shape of a role against the shape of your capabilities, revealing where you're strong, where you can stretch, and where you'll struggle. This is not a career framework, although it can inform one.

It helps you answer three critical questions:

1. What does this role actually require?
2. Where do my strengths sit right now?
3. What's the gap, and do I want to close it?

Career Vectors work whether you're a senior engineer, an EM, a director, a CTO, or a CEO trying to hire one.

What Career Vectors are

Career Vectors plot six dimensions of technical leadership on a scale from 0 to 5.

You create two shapes:

- The role's expectations (defined by the CEO / hiring manager)
- Your honest self-assessment

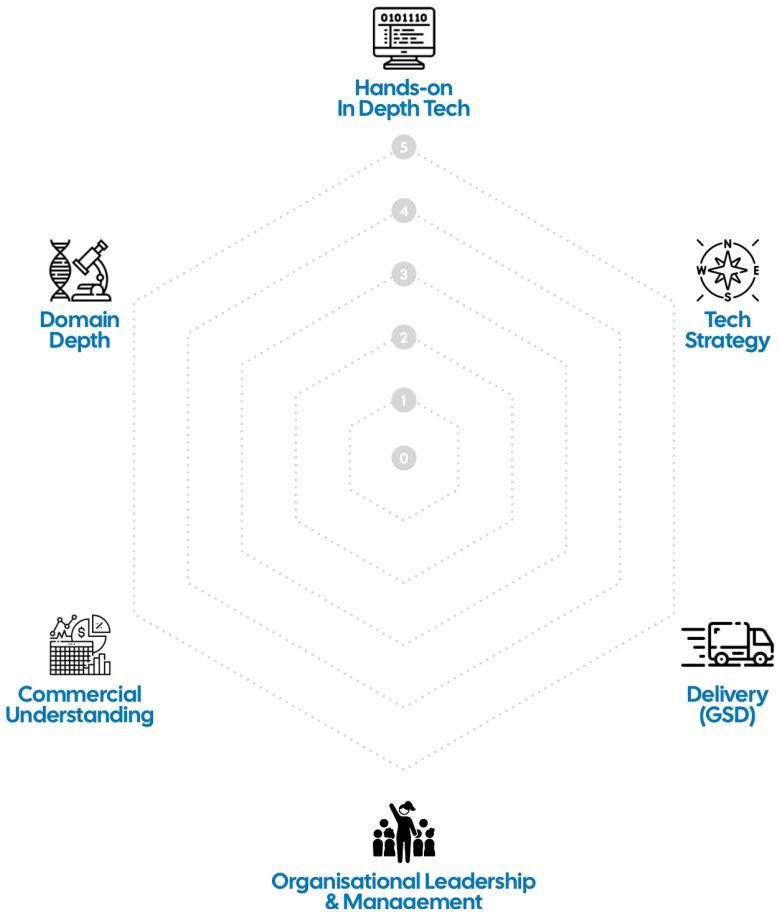
Overlay them, and you immediately see:

- Where you're a close match
- Where you'd grow
- Where you'd be miscast
- Whether the job will energise you, stretch you, frustrate you, or burn you out

It removes guesswork. It forces clarity. And it sparks the most valuable conversation you will ever have with your manager or CEO: *“What do you actually expect from me, and are we aligned?”,* something many teams never have until it’s too late.

The Six Dimensions

Each Career Vector uses the same six dimensions. Together, they cover the full spectrum of senior technical leadership.



Career Vectors: *The six dimensions technical leaders should master.*

1. Hands-on / In-Depth Tech

Your real technical depth today, not the depth you once had. Signals whether you can contribute technically or whether you must rely entirely on the experts you hire. How deep is your technical knowledge today: not in theory, not "*back when you coded every day,*" but right now?

Ask yourself:

- Could I step into a team and be productive as a senior/staff engineer?
- Can I meaningfully contribute to architectural discussions?
- Can I evaluate technical risks with confidence?

This dimension changes dramatically depending on:

- Company stage
- Team maturity
- Whether tech is the product
- Whether you're expected to be an expert or hire experts

2. Tech Strategy

Your ability to define the direction of technology in service of the business.

Includes:

- Architectural strategy
- Buy vs build
- Long-term technical investments
- Engineering principles
- Choosing what not to do

Ask yourself:

- Do I create clear guidance for teams?
- Can I explain technical decisions in business terms?
- Do I shape strategy, or simply react to it?

3. Delivery (GSD)

Your ability to help teams ship predictably and consistently. Strong leaders here create momentum. Weak leaders here create drift.

Includes:

- Ways of working
- Delivery systems
- Program/project management

- Removing bottlenecks
- Keeping customer value flowing even during large transformations

Ask yourself:

- Am I strong at execution?
- Do teams I lead get things done?
- Do I personally strengthen or weaken delivery?

4. Organisational Leadership & Management

Your ability to build, lead, and scale an organisation, from leading one team to leading multiple layers. Where many engineers discover that leadership is not a promotion; it's a new profession.

Includes:

- Line management
- Career development
- Coaching
- Performance management
- Delegation
- Leading managers of managers
- Creating clarity
- Setting culture

Ask yourself:

- Can I lead teams of 10?
- Can I lead teams of 100?
- Can I shape a whole engineering organisation?

This is where many senior ICs discover that leadership is not a "promotion"; it's a career change.

5. Commercial Understanding

Your grasp of how the business operates, where money is made or lost, and how tech decisions influence those things. Great leaders speak finance as confidently as they talk about engineering.

Includes:

- Reading a P&L

- Understanding gross margin & cost levers
- Appreciating pricing, sales, and acquisition models
- Evaluating ROI of tech choices
- Understanding the cost of architectural decisions (e.g. cloud, observability, AI)

Ask yourself:

- Do I understand the economics of my product?
- Do I speak the language of finance, sales, and product?
- Do I make commercially intelligent decisions?

6. Domain Depth

Your understanding of the industry you're operating in. This dimension varies the most and is the one most likely to cause failed hires if ignored.

Varies dramatically:

- E-commerce? Lighter. You're already a customer.
- Drug discovery? Deeply technical, heavily regulated.
- Banking? Complex, legacy, risk-driven.
- Logistics? Operational, optimisation-heavy.

Ask yourself:

- Do I understand how this industry really works?
- Can I make decisions with confidence?
- Am I motivated to learn this domain, or would it drain me?

This is the dimension with the highest variation, and often the one that causes failed hires.

How to Use Career Vectors

1. Define the shape of the role (CEO / hiring manager)

Before hiring or promoting anyone, define the role's shape. Be explicit about what the company needs now, not what every CTO ever should be. Define:

- What the role actually needs
- What level (0–5) is required for each dimension
- What's nice-to-have vs essential
- Which gaps can be covered by the rest of the leadership team

This avoids the classic CTO job spec: "*Expert in everything, all the time.*"

2. Self-assess your shape (technical leader)

Score yourself honestly. Think about today, not your historical best. Motivation matters; don't claim a gap you don't want to close. The value comes from clarity, not bravado.

Ask:

- What am I truly strong at?
- What drains me?
- What am I motivated to learn?
- Where do I need support?

Plot your shape.

3. Overlay the shapes and see the truth

When you overlay "role needs" vs "what I bring," three zones emerge:

- **Overlap:** you're a strong match

- **Stretch:** you'll grow and be challenged
- **Mismatch:** the role needs something you don't want to build (career misalignment)

This is where the magic happens. It leads to a clear conversation: *"Here's where I'm strong. Here's where I'll grow. And here's where I'll need help."*

Applying the Framework

1. Mapping a role you already have

Use Vectors to:

- Re-align expectations
- Create a clear development plan
- Understand where to double down
- Identify where you need support
- Avoid burnout by spotting mismatches early

This prevents the slow slide into "*I'm struggling, and I don't know why.*"

2. Evaluating a new role

Before saying yes to a job:

- Map the role
- Map yourself
- Compare
- Decide deliberately

This stops people from walking into roles they're not right for, or roles that aren't right for them.

3. Using it as a CEO

For CEOs, this framework:

- Clarifies what type of leader you're actually looking for
- Prevents cookie-cutter CTO hires
- Reveals whether a candidate matches the stage of your business
- Helps you cast the right person for the role – not the mythical "superhero CTO"

4. Comparing similar roles

Career Vectors also help you compare related roles, such as:

- Engineering Manager vs Software Delivery Manager
- Engineer vs Architect
- CTO vs VP Engineering
- Technical Program Manager vs Product Manager

These comparisons show:

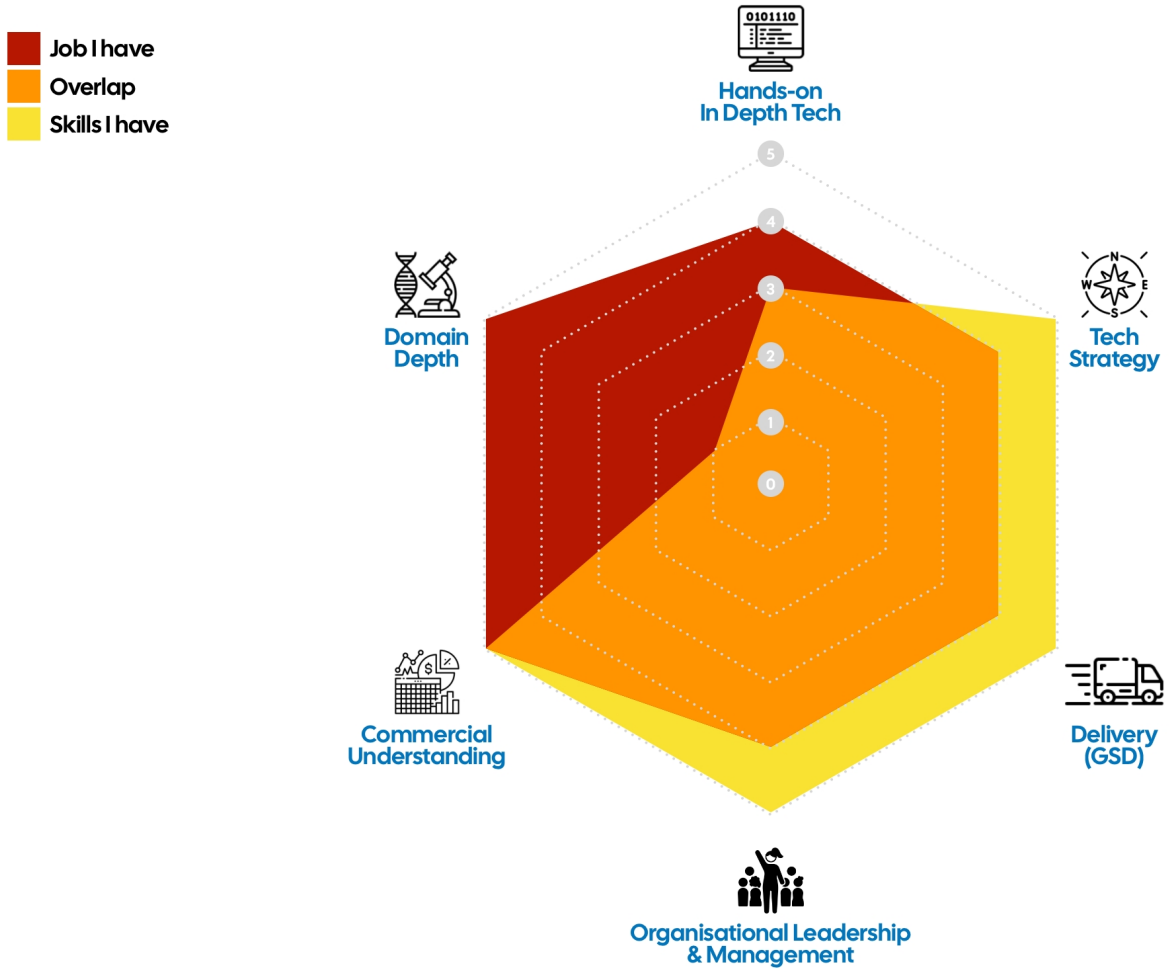
- Where roles overlap
- Where responsibilities diverge
- Why some people thrive in one but not the other
- Where career paths naturally flow

Example charts

Here are some charts that represent some scenarios. They might not be accurate, and you might draw them differently, but they are here to illustrate the concept.

1. The job I have compared with the skills I have.

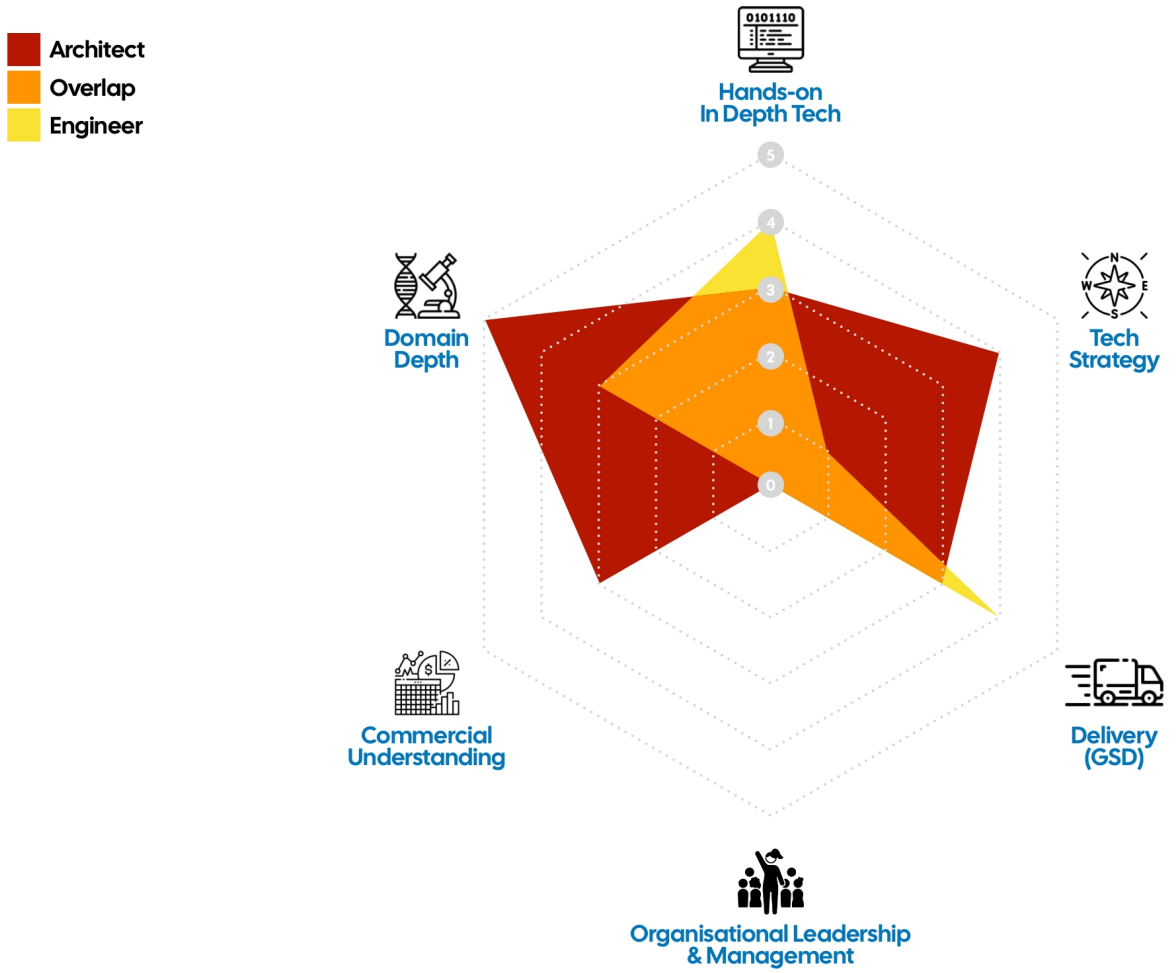
This example shows how you might map the role you are currently in and compare it with the skills you have. This shows where you are a close match and where you have room to develop and grow.



Example 1: Your role vs your skills: Where you match, where you stretch, and where you'll struggle. A clear starting point for an honest development plan.

2. An architect compared with an engineer.

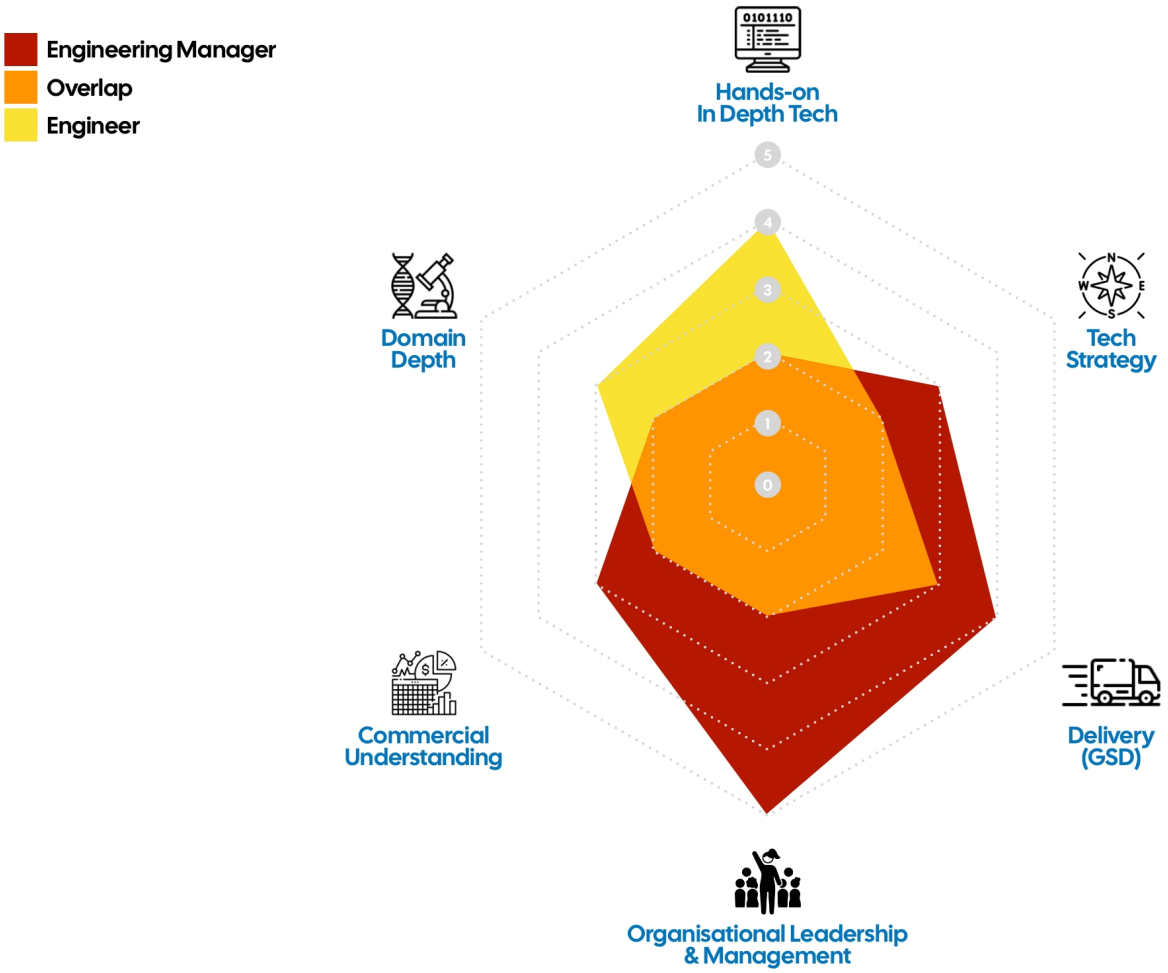
At first, you might assume an Architect is merely a more tenured Senior Engineer. When you map them, it's clear they are not. An Architect specialises in understanding the domain and the business, and in constructing technical strategies that deliver on them. In contrast, an Engineer is heads-down in the code, building the software to meet the plan designed by the Architect.



Example 2: An Architect compared with an Engineer. Architects aren't '*senior engineers*.' They specialise in strategy, domain depth, and shaping systems. Engineers focus on building: different strengths, different demands.

3. An Engineering Manager compared with an Engineer.

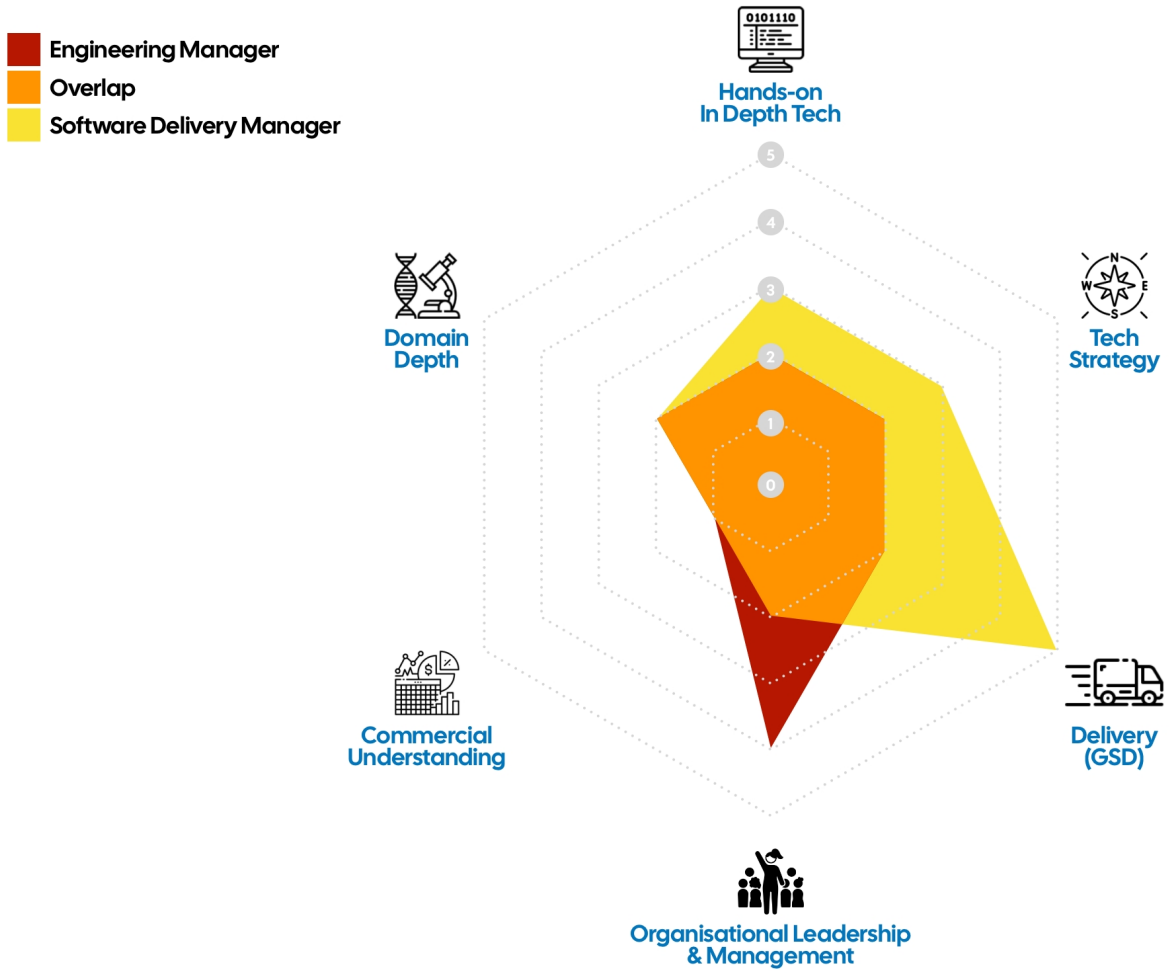
There is a common idea that moving from an Engineer to an Engineering Manager is a promotion. When you map the expectations of both roles, it becomes apparent that there is a significant shift. Engineers can certainly progress into management, but they must understand the different demands: what they will stop doing (e.g., less coding) and what they will start doing (e.g., people management and delivery).



Example 3: An Engineer compared with an Engineering Manager. Engineering management isn't a promotion; it's a role with entirely different skills: less code, more people, more leadership.

4. An Engineering Manager compared with a Software Development Manager.

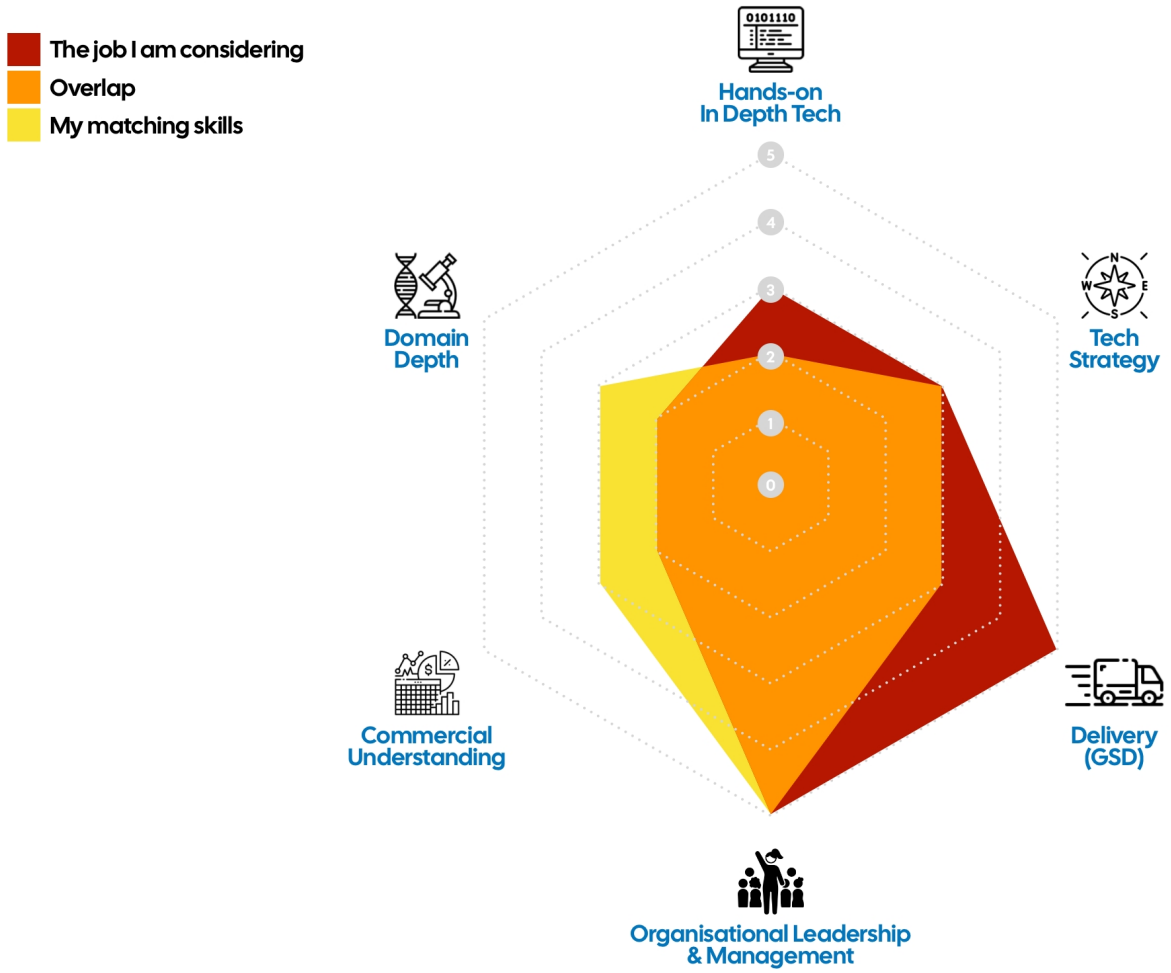
Compared with a Software Development Manager, an Engineering Manager has stronger people-management skills and less focus on delivery, and vice versa.



Example 4: An Engineering Manager compared with a Software Development Manager. Both drive delivery, but in different ways. EMs develop people and teams; SDMs emphasise execution, coordination, and delivery systems.

5. Mapping your skills with a prospective role.

You can use this method to map your skills against a role you are considering. Whilst it might be tempting to accept a new job, it's worth pausing to see how well you align with the expectations. Sometimes, it's better to say no than to fail.



Example 5: Mapping your skills with a prospective role. Before accepting a job, map the role's shape against your own. A mismatch now prevents a painful miscast later.

Why Career Vectors Matter

Career Vectors matter because they make leadership real. They replace vague job descriptions with clarity. They replace assumptions with shared understanding. They turn “maybe this job will fit” into “now I know.”

Most importantly, they help people find roles where they can thrive and avoid the ones where they’ll drown.

And it opens the most important door of all: *A transparent, grown-up conversation about what you need to succeed.*