

Balderton.

Building Great Teams - Part 4

Career Development Frameworks

Leah Sutton (Chief Portfolio Talent Officer)

Dan Teodosiu (Executive in Residence)

About ourselves

Leah Sutton

- 25 years of HR operating experience, rapidly scaling global, distributed teams and companies in different sectors:
 - Clean Tech (Bloom Energy) - 150-1500ees, led HR for Asia Pacific
 - Food / Consumer Goods (Plum Organics/Campbell Soup Company - led through an acquisition and integration
 - Software/OSS (Elastic) - pre-to post-IPO, 250 to 3200 employees in 40+ countries



About ourselves

Dan Teodosiu

- 30+ years of industry experience with:
 - Scale-ups (CTO of Criteo and Onfido)
 - Large companies (Google, Microsoft)
 - Startups (as a founder, VP Eng, technical advisor)
- Hired > 1500 people (Engineers, Product, Support, HR, Finance – from individual contributors to C-level)
 - Ran tech recruiting at Criteo and Onfido
- Scaled high-performance engineering & research teams to > 600 people
- As an EiR at Balderton, I'm here to **help you!**



Outline

CDF = Career Development Framework

What we'll cover in this presentation:

- What is a CDF
- Anatomy of a CDF
- Implementing a CDF
- How to use a CDF
- When to introduce a CDF

What is a CDF?

Formalization of **career tracks** and **seniority levels**:

- **Tracks**: Software Engineer, Engineering Manager, Product Manager, Recruiter, HRBP, ...
- **Levels** e.g. for engineer IC: Software Engineer (SWE), SWE II, Senior SWE, Staff Engineer...

Every company past a certain size needs a CDF:

- CDF needs to be **tailored to your company and culture**
- CDF should reflect your **company values** and will **amplify them** over time

CDF is **used for**:

- Determining compensation
- Performance evaluation
- Promotions
- Hiring
- Career development
- Career changes

B.

Anatomy of a CDF

Widget Inc. CDF:

	Career Tracks			
Level	Software Engineer	Engineering Manager	Product Manager	...
1	Software Engineer <expectations for SWE 1>	N/A	N/A	
2	Software Engineer II <expectations for SWE 2>	N/A	Product Manager <expectations for PM 2>	
3	Senior Software Engineer <expectations for SWE 3>	N/A	Senior Product Manager <expectations for PM 3>	
4	Staff Engineer <expectations for SWE 4>	Engineering Manager <expectations for EM 4>	Senior Program Manager II <expectations for PM 4>	
...				

B.

Anatomy of a CDF

Widget Inc. CDF:

	Career Tracks			
Level	Software Engineer	Engineering Manager	Product Manager	
1	Software Engineer <expectations for SWE 1>	N/A	N/A	...
2	Software Engineer II <expectations for SWE 2>	N/A	Product Manager <expectations for PM 2>	
3	Senior Software Engineer <expectations for SWE 3>	N/A	Senior Product Manager <expectations for PM 3>	
4	Staff Engineer <expectations for SWE 4>	Engineering Manager <expectations for EM 4>	Senior Program Manager II <expectations for PM 4>	
		...		

Anatomy of a CDF

Defining standard titles and expectations

Align standard titles to **industry standards**:

- Helps with **recruiting** (e.g. looking for “Staff Engineer”)
- Conveys meaningful **seniority information** e.g. when talking to customers

Expectations for each level:

- General ones: **align to company culture & values** – similar across different career tracks
- Career track and level-specific ones
- Split into areas, e.g. for Software Engineer (SWE) **individual contributor**:
 - **Technical Skills**
 - **Product/Customer Awareness**
 - **Teamwork**
 - **Ownership and Accountability**
 - **Recruiting**

Anatomy of a CDF

Expectations example for Software Engineer (SWE 1)

Software Engineer (SWE 1)			
	Company Values:		
		Collaboration	<ul style="list-style-type: none">• Build great working relationship within their team• Effective at communicating own status to their team• ...
		Focus on the customer	<ul style="list-style-type: none">• Understand customer requirements and take them into account• Puts themselves in the customer's shoes by using the product• Responds to customer enquiries/issues promptly & effectively• ...
		...	
	Engineering Skills:		
		Technical skills	<ul style="list-style-type: none">• Understands & applies fundamentals of software engineering• Produces correct and readable code• Asks for peer reviews of any code produced• ...
		Ownership & accountability	<ul style="list-style-type: none">• Demonstrates accountability for the quality of their code• Feels responsible for delivering a successful component or product• May require some instructions for daily work• ...
		...	

Implementing a CDF

Copy & adapt

Don't reinvent the wheel, **copy shamelessly**:

- Many useful resources can be found on the web
- E.g. great resource on **GitHub**: <https://github.com/posquit0/awesome-engineering-ladders>
- More work has been put into these than meets the eye (e.g. **promo velocity** is an important implicit aspect of every CDF)

Adapt to your **company culture and values**:

- These should be **accurately reflected** in the CDF
- The CDF defines **how people are evaluated** and **how they grow** in your company
- The CDF shows what **behaviors** are **encouraged** in your company
- The CDF becomes a **lever** for **scaling your company culture**

Implementing a CDF

How many levels? Which tracks?

Number of levels:

- **Avoid too many levels** → more work to define and maintain the CDF
- **Aim for the lower 4–6 levels initially**, covering fresh college grad to Director
- Levels determine **career progression velocity**
- Refine and add levels **later** as needed

Which tracks:

- Start with **a few**, e.g. covering main roles in Engineering and Product
- E.g. Software Engineer, Development Lead, Engineering Manager, Product Manager
- **Specialize later** as you develop the roles, e.g. DevOps, Researcher, Technical Program Manager, etc.
- Run a **pilot** with the initial tracks, then expand to other functions

Avoid the “glass ceiling”:

- All tracks should go up to the same level (whenever possible)

Implementing a CDF

Who develops the CDF?

Strong adoption on your team is key:

- **Bottom-up** effort & **partnership** between the respective function (e.g. Engineering) and HR
- Put together a small working group per track or group of tracks (e.g. in Engineering or Product)
- Have your entire team **review and provide feedback**

It's an **in-house effort**:

- Don't use consultants – generally saves you time and €

Transparency and **openness** is key:

- Publish CDF and employee leveling internally
- (Later) make promotions visible internally
- Encourage employees to use their standard titles on LinkedIn

Implementing a CDF

Initial leveling

Level people on defined (and “adjacent”) tracks:

- **Assign** people to defined tracks
- **Fudge it** if you don’t yet have a track defined (e.g. if you only have one DevOps in your team, use the SWE track for them)
- You can only do this for the functions where you’ve defined (relatively close) tracks

Deciding on initial levels:

- Use a group of managers and peers to **assign a level** to everyone in their track
- **Broad acceptance** of this initial leveling is key
- If you’ve used “random titles” before, you may need to have some difficult conversations
- Be prepared to have to do **salary adjustments** at this point

B.

Using the CDF

What to use the CDF for?

- Compensation
- Performance evaluation
- Promotions
- Hiring
- Career development
- Career changes

Using the CDF

Compensation (1): using compensation bands

Make sure you have a **company compensation philosophy**:

- What are your target types of companies you're competing with for talent?
- How do you want to position your comp relative to them? E.g. top 80%.

Define compensation bands:

- Par **career track, level** and **geo**
- For each band: **low end, high end, mid-range** (not necessarily in the middle)
- Start with **identical comp bands** for all engineering & Product tracks, per geo

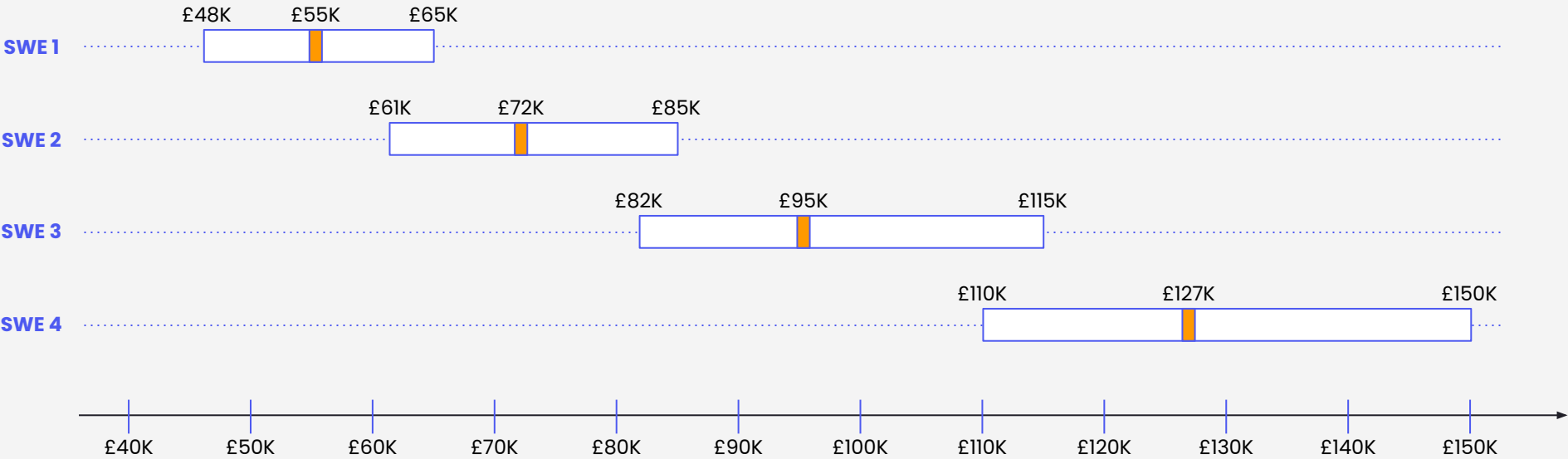
Use **market intelligence** to define the compensation bands:

- Could use a specialized company such as [Ravio](#), [Pave](#) (both accessible to our portfolio companies) or [Figures](#)
- Redo benchmarks periodically depending on how the market is moving (once or twice a year)

B.

Using the CDF

Compensation (2): example bands for Software Engineer track, levels 1-4, in the UK



Note: these are for illustration only, **not** current/accurate market figures.

Using the CDF

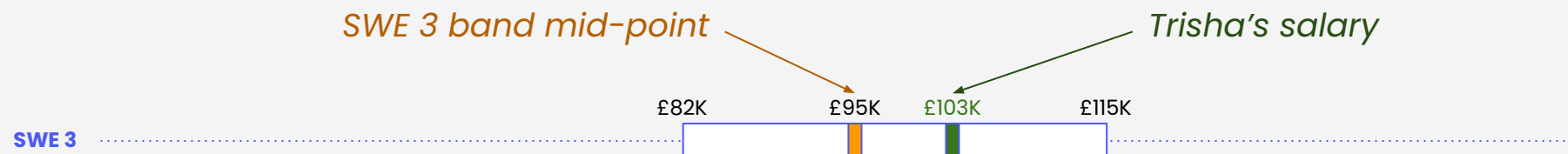
Compensation (3): Compa Ratio (CR)

Use the Comp Ratio (CR) to **compare compensation** across tracks, levels and geos:

- **CR** = **Salary** / **Mid-point**
- Allows for easy comparison across levels and geos

For instance:

- Trisha is an SWE 3 who earns £103K p.a.
- The mid-point for SWE 3 has been set at £95K
- **Trisha's CR** = $103 / 95 = 1.08$



Using the CDF

Compensation (4): review after initial leveling

For each person leveled:

- Compute actual **CR_{actual}**
- Choose target **CR_{target}** based on:
 - Performance
 - Length in level (\approx time in the company)
- May need to **adjust salaries** at this point:
 - $CR_{actual} > CR_{target} \rightarrow$ you've probably overpaying, use smaller annual adjustments to catch up
 - $CR_{actual} < CR_{target} \rightarrow$ you may consider adjusting their salary
 - **CR_{actual} < 0.85** \rightarrow they are paid **below market!** Potential flight risk?!

Using the CDF

Performance evaluation

Expectations per track & level:

- Provide a framework to uniformly assess how people perform
- Spell out what behaviors are in line with company culture/are encouraged
- Close to, but **not an exact science**

Performance score:

- Use a performance score (e.g. Below, Average, Strong, Outstanding)

Performance should determine comp:

- **Annual increase** in CR **based on score**
- Example: Below 0%, Average 2%, Strong 4%, Outstanding 5%
- Additional **equity grants** should also be based on performance score

Using the CDF

Promotions

When to grant a promotion?

- Person performs at **70–80%** of expectations for next level
- Again, CDF is not an exact science but provides **useful guidelines**

Promotion velocity you should target for great performers:

- 1.5–2 years for level 1→2
- 2–3 years for higher levels

Adjust comp on a promo:

- Typically target **CR = 0.9** in new level
- Be flexible to provide a reasonable bump (at least **+10%** in comp)

Using the CDF

Hiring

Level new hires correctly:

- Based on interview performance, seniority, comparison to existing employees
- Hiring Manager should determine the target level of a candidate

Use the level to determine their **package**:

- Target **CR = 0.85-0.9** if possible (but their current salary/experience may exceed that)
- Also compare against CR's of other people on your team

Avoid **leveling mistakes**:

- **Under-leveling** is easy to correct (at end of probationary period or via a rapid promo)
- **Over-leveling** is very hard to fix (unproductive discussions, cannot lower their salary)

Using the CDF

Career development

CDF can be a very useful **resource** to answer questions such as:

- “What criteria will I be rated on during performance evaluation?”
- “What skills do I need to develop to qualify for a promotion?”
- “What behaviors are in line with our company culture?”
- “What do I need to prove I can do to qualify?”

Discuss during 1-1's with your reports:

- Which 70–80% of expectations for the next level should they focus on?
- What's the plan for them to ramp up?

Long-term **career visibility**:

- “What do I need to know to eventually make it to Staff Engineer?”

Using the CDF

Career changes

Career changes should be **iso-level** as much as possible, e.g.:

- Software Engineer (individual contributor) SWE 3 → Development Lead DL 3
- Development Lead DL 4 → Engineering Manager EM 4
- Software Engineer SWE 2 → Product Manager PM 2

Do not present track changes as **promotions**:

- They are just changes in role
- **Keep the door open** for the person to revert to their old track if the new one doesn't work out
- Keeping the same comp makes the changes more **fluid**

Some moves may not be possible, e.g.:

- SWE 1 → PM 1 since the PM career track starts at level 2 (requires industry experience)

When to introduce a CDF

Not to soon & not too late

Your team needs to reach a **critical mass** before a CDF makes sense:

- Minimum **8-10 people** for a given career track
- Use “adjacent” tracks when possible (e.g. SWE for DevOps)
- Don’t delay too much though – avoid having large comp discrepancies that you’ll need to fix later

Start with **Tech** (Engineering + Product):

- This will most likely be the largest population in your company
- Easier to build the CDF for these tracks with the help of the Tech people
- This can be copied & adapted later for other tracks

Summary

What we've covered in this presentation

- Why you need a CDF
- How to introduce a CDF
- What to use the CDF for

Balderton.

Q&A

Balderton.

THANK YOU.

LET'S TALK.

For any questions, please reach out to

Leah: lsutton@balderton.com or Dan: dt.eir@balderton.com

balderton.com

[@balderton](#)