

Balderton.

Building Great Teams - Part 3

Hiring Great Technical Teams

Dan Teodosiu (EiR)

About myself

I'm passionate about building great teams to build great products

- 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

Your team is your most valuable asset

"The secret of my success is that we have gone to exceptional lengths to hire the best people in the world." [Steve Jobs]

What we'll cover in this presentation:

- Hiring is a key part of your culture
- Bootstrapping and scaling recruiting
- Recruiting pipeline
- Interview process
- Go/no-go and offer stage
- Improving your recruiting skills and accuracy

B.

Hiring is a key part of your culture

B.

Hiring is a key part of your culture

Recruiting is everyone's job in your company

"Refocusing your resources on hiring better will have a higher return than almost any training program you can develop." [Laszlo Bock]

- Your talent acquisition team is a **partner** in recruiting, but...
It's also your job (and everyone's in the company)
- **Invest as much time as needed** – this will pay off in the long run
- Sourcing, referring, contacting, interviewing, closing candidates
- Make sure **your teams do this as well**
- Recruiting should become a key **part of your company culture**
 - Everyone should be a "bar raiser"



B.

Hiring is a key part of your culture

Only go for the best

“A players hire A players; B players hire C players; and C players hire D players. It doesn't take long to get to Z players. This trickle-down effect causes bozo explosions in companies.” [Steve Jobs]

- Team dynamics:
 - Great people **attract** other great people
 - 2 mediocre engineers get (much) **less** work done than 1 outstanding one
 - One mediocre engineer can **slow down** an entire team
- Tension between quality and quantity/speed of hires:
 - **Always go for quality**, even if it takes much longer (you'll **save time** in the long run)
 - Hiring the best is **difficult** irrespective of the economic environment

B.

Bootstrapping and scaling recruiting

Bootstrapping and scaling recruiting

The first hires you make set the tone

"I hire people brighter than me and then I get out of their way." [Lee Iacocca]

- Use your network to **source** and **interview** candidates
- Founders should spend **significant time** on recruiting
- **New hires** who join the company should contribute to recruiting
- Hire (mostly) **senior people** initially:
 - Junior people are cheaper and easier to find
 - But are you sure you can scale your team that way? To 50? 100?

Bootstrapping and scaling recruiting

Get professional help

"If you think it's expensive to hire a professional, wait until you hire an amateur." [Red Adair]

- Below C-level, **in-house or freelancer** works better than agency model (based on my experience)
- Attributes of a **great recruiter**:
 - Passion & perseverance/hunter mentality
 - Ability to articulate problem space and product in a compelling way
 - Adapting to each candidate's personality
 - Great intuition for the quality of candidates
 - Operates in a data-driven mode
- How many recruiters do you need?
 - A good recruiter can help you get **2-4 offers/month**
- The first **recruiting lead** you hire should be **hands-on**

B.

Recruiting pipeline

B.

Recruiting pipeline

Feed your pipeline

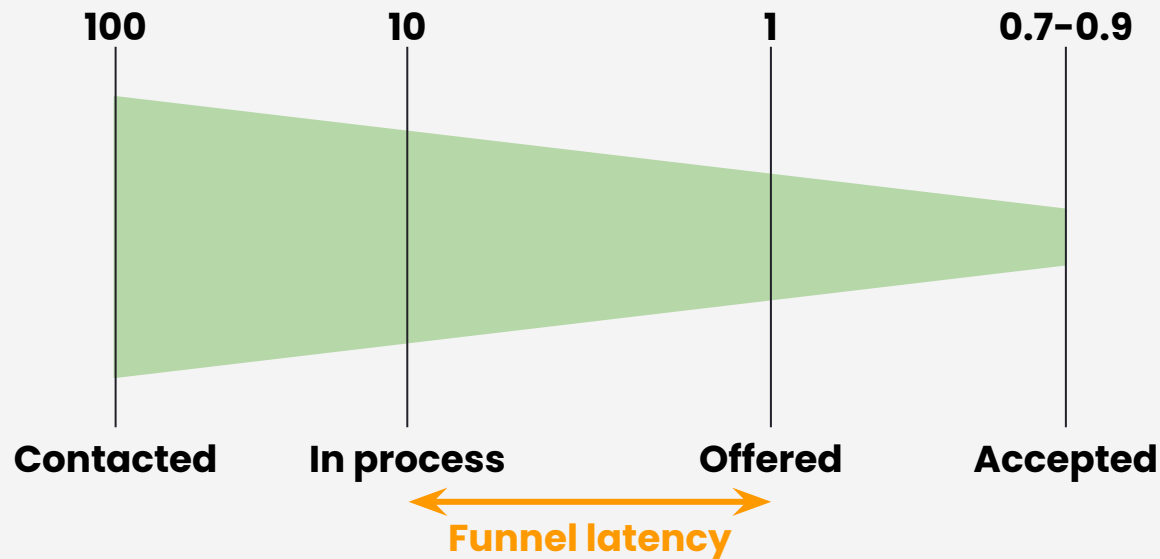
“Acquiring the right talent is the most important key to growth.” [Marc Benioff]

- Don't expect people to just walk through your door:
 - **Sourcing** + **referrals** are your friends
- Build the pipeline in a **systematic way**:
 - Decide on what **geos** you want to focus on first
 - What (types of) **companies** yield the best profiles?
- Focus on **diversity** from the very beginning
- Look for the right kind of **experience/background – general skills**, not narrow ones:
 - In most cases, senior candidates can ramp up on your tech stack in 1-2 months
 - However, this doesn't apply for highly specialized areas such as ML, Security, DevOps...

B.

Recruiting pipeline

Watch your pipeline



100:1, really?

- Applies to cold-contacted candidates
- I've seen 80:1 up to 400:1 in the industry
- In process:offered ratio varies less

- **Track** your pipeline stats, mainly **funnel ratios** and **funnel latency**
 - Optimize when you see significant deviation from these ratios
- Use an **ATS** (Applicant Tracking System) such as Lever early on
- **Referrals** can improve the ratios significantly: target **20-30% of hires** (→ pay a **referral bonus**)

B.

Recruiting pipeline

Tune your pipeline

“Hiring is the most important people function [...], and most of us aren’t as good at it as we think.” [Laszlo Bock]

Contacted	<ul style="list-style-type: none">• Sourcing volume is the adjustment variable here• Involve your teams in sourcing (including focused diversity sourcing)
In process	<ul style="list-style-type: none">• Message sent to candidates, who contacts them (e.g. CTO)• Hone your pitch & make sure your recruiter(s) master it
Offered	<ul style="list-style-type: none">• Quality of sourcing & focus on the right profiles• Interview process & probing for the right skills
Accepted	<ul style="list-style-type: none">• Pitch, offer package, perseverance

B.

Interview process

Interview process

It's all about probing depth

- When interviewing, you're **measuring depth, not checking boxes**
- **Formalize** the interview process for each profile/job spec:
 - E.g. phone screen + 4 steps (avoid more than 4 and “homework assignments”)
 - What gets covered in each step (e.g. coding, coding, design, culture fit/motivation/final)
- Internal **recruiting guild**:
 - **Standardize the questions** + **document good answers** and drill-down questions
 - Update this information regularly (internal wiki)
- Interviewer **calibration**
- **Probe vs. sell?**
 - **Adjust** based on candidate profile

B.

Interview process

Evaluate specific skills

Right approach:

- Ask **competency-based questions** that can only be answered by someone who has that skill
- Examples: write code, solve a specific problem, debug an issue
- “Describe a situation where you had to take immediate action because the service was down”

Wrong approach:

- Assume they have the skill because it’s listed on their profile
- Assume they have the skill because they claim to
- Think they must have the skill based on their education and experience
- Anti-patterns: “Describe your current project”, “Have you used React Native?”

Remember: never assume anything you haven’t probed



B.

Interview process

Have an interview plan

- Example for software developer generalist:

Interview stage	What's being assessed?
Phone screen (30')	<ul style="list-style-type: none">• Depth assessment based on previous experience (10')• Basic skills assessment based on small technical problem (15')• Initial selling (5')
Coding C1 (60')	<ul style="list-style-type: none">• Fundamentals (data structures and algorithms), coding, testing
Coding C2 (60')	<ul style="list-style-type: none">• Fundamentals (data structures and algorithms), coding, testing
Systems design (60')	<ul style="list-style-type: none">• Understand choices and trade-offs, make reasonable estimates• Architectural skills, scalability, technology options
Final (60')	<ul style="list-style-type: none">• Motivation, team fit, seniority• But also: sell, sell, sell!

Interview process

Have an interviewer checklist

- I'm familiar with the **candidate's profile**:
 - At least one question related to an item in her profile
- I know **what to assess**:
 - List of criteria I want to check
 - For each criteria I know how to drill down
 - I know what criteria I don't have to assess
- I have at least one **backup question** handy
- I know what **concerns** were raised during previous interviews:
 - I'm ready to drill-down on additional topics to **double-check**

B.

Interview process

Assess cultural fit, cultural add and motivation

- Cultural fit and cultural add should be assessed by all interviewers:
 - **“Do I want to work with this candidate?”**
 - “Would this candidate fit in well with our company culture?”
 - **“Would this candidate enrich our company culture?”**
- **Motivation** primarily assessed during final interview – questions to ask:
 - How does **success** look for you from a **personal perspective**?
 - Describe your **ideal role** (especially for junior people: architecture, coding, debugging...)
- Gauge level of energy & passion (all interviewers):
 - Be aware of candidate personality/cultural differences and how much they express this

Interview process

Do's and don'ts

- **DO** → **Be in control** of the conversation:
 - **Manage the time** carefully during the interview
 - **Steer** the discussion **politely**
- **DON'T** → **Topics you should not discuss** during the interview:
 - Any **sensitive topics** (personal status, political or religious beliefs, sexual orientation, etc.)
 - **Salary expectations** (except for final stage)
- **DO** → Be prepared to **handle difficult situations**:
 - E.g. candidate doesn't talk, talks too much, is visibly intimidated, talks about topic you're not familiar with, becomes aggressive, starts volunteering personal details...
- **DON'T** → **Things to avoid**:
 - trying to prove you're smart, not listening, dwelling on failures, being smug or impolite

Interview process

Provide an excellent candidate experience

- **Remember:** you're interviewing the candidate but they're also **gauging you and your company**
 - For a small company, **branding = candidate experience**
- **Before** the interview:
 - Let candidates know **what to expect**/how the interview loop is structured
 - Make sure the recruiters can answer any candidate questions
- **During** the interview:
 - **Adapt** to the level of seniority of the candidate, **listen**, be open & **friendly**
 - Give candidates **something to take away** from the interview even if you don't hire them
- **After** the interview:
 - Run **candidate surveys** for both hired and rejected candidates
 - **Keep in touch** with candidates who decline your offers, you may be able to hire them later

B.

Go/no-go and offer stage

B.

Go/no-go and offer stage

Use a scoring system for each interview stage

Score	Meaning
5.0	<ul style="list-style-type: none">• The candidate was incredible! Answered the questions better than I could• We should do everything we can to hire them!
4.0	<ul style="list-style-type: none">• Strong candidate, good or great answers to all questions
3.0	<ul style="list-style-type: none">• The candidate was good, reasonable answers to the questions asked• I'd work with them!
2.0	<ul style="list-style-type: none">• Said a lot but little of interest, or gave the wrong answers• Stop the process here!
1.0	<ul style="list-style-type: none">• Unable to answer questions or gave mostly incorrect/bad answers• I'll quit if we hire them!

- Make sure interviewers are well **calibrated** and write down **interview notes**
- Decide on the **criteria for when to stop** the interview loop

Go/no-go and offer stage

Offering and closing candidates

- **Turn-around time** for the offer matters:
 - Aim for 24h or **at most 48h**
 - Make offer conditional on checking references
- Understand package **expectations** and **market rates**:
 - Use your experience, your connections, and a benchmarking company to understand rates
 - Don't offer lower than market (it will come back to haunt you later)
- **Never forget to check references**
- Spend as much time needed **selling** your company/opportunity:
 - Accommodate candidate requests for further discussions with the founders/team

B.

Go/no-go and offer stage

Properly account for seniority

- Start with a simple **career ladder** (e.g. at least 3 levels):
 - **Copy an existing one & adapt** to your values (search for “engineering ladders” on GitHub)
 - Your teams/disciplines should own their respective career ladders
- Make sure new hires are **leveled consistently**:
 - Where do they fit on the ladder?
 - How do they compare to your existing team members
 - Is their **package** (salary + bonus + equity) appropriate?
- If this is not done right, I have war stories for you...

B.

Improving your recruiting skills and accuracy

Improving your recruiting skills and accuracy

Keep track of and learn from your mistakes

- We all make **hiring mistakes**, here are some of my past ones:
 - The software developer who couldn't write a single line of code
 - The database expert who couldn't design a simple schema
 - The ops person who filled the datacenter cage with garbage
- Try not to make too many mistakes & **learn** from the ones you've made:
 - **Check** how hires are doing after 3-6-9 months
 - When you've made a mistake, ask yourself **what you could have done differently** during the interview process
- **Do not use probation periods** to correct for frequent recruiting mistakes:
 - Fix the mistakes instead!

Summary

Your team is your key asset.

“The key for us, number one, has always been hiring very smart people.” [Bill Gates]

- Your company’s success **critically depends** on the team you build
- You & your team should **spend significant time on recruiting**
 - Hiring should be a key part of your culture
- Make sure you have a **great recruiting process** in place
- Always keep the **candidate experience** in mind
- You can **get better** at hiring over time

Be proud of great hires you’ve made and remember:

- Your best hires are people who are **better than you**

Balderton.

Q&A

Balderton.

THANK YOU.

LET'S TALK.

For any questions, please reach out to Dan: dt.eir@balderton.com

balderton.com

[@balderton](#)