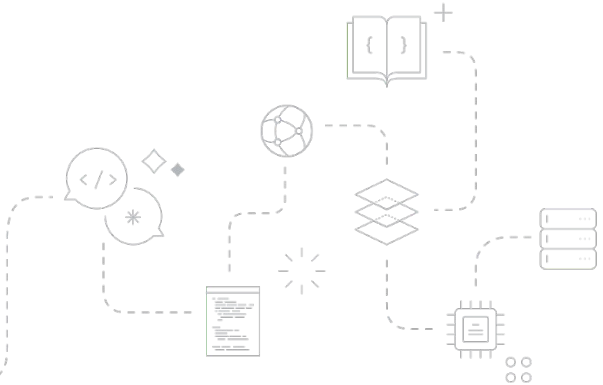
Hiring as Management

Stack Overflow



Hello, world!

A little bit about me

Stack Overflow

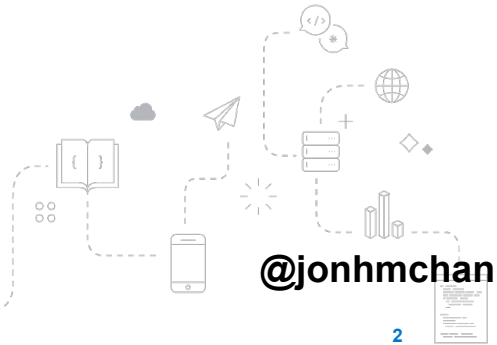
Marketing Engineering Lead

I run all engineering support for the Stack Overflow brand. I think a lot about content and growth across the company.

Bento

Founder

My passion project building tools for self-taught developers. I think a lot about how people learn, especially beginners.



@jonhmchan

Overview

What we're talking about today

Philosophy

`SELECT isms FROM philosophy;`

What does it mean to be a manager at Stack Overflow? How do we think about the values we hold that translates into hiring?

Interviews

`INSERT INTO company VALUES devs;`

We have a very transparent and rigorous process for interviewing developers at Stack Overflow. Here's how we do it.

Something I miss?

You can always get in touch with me at **@jonhmchan** or email me at **jon@stackoverflow.com**

Performance

`UPDATE skills;`

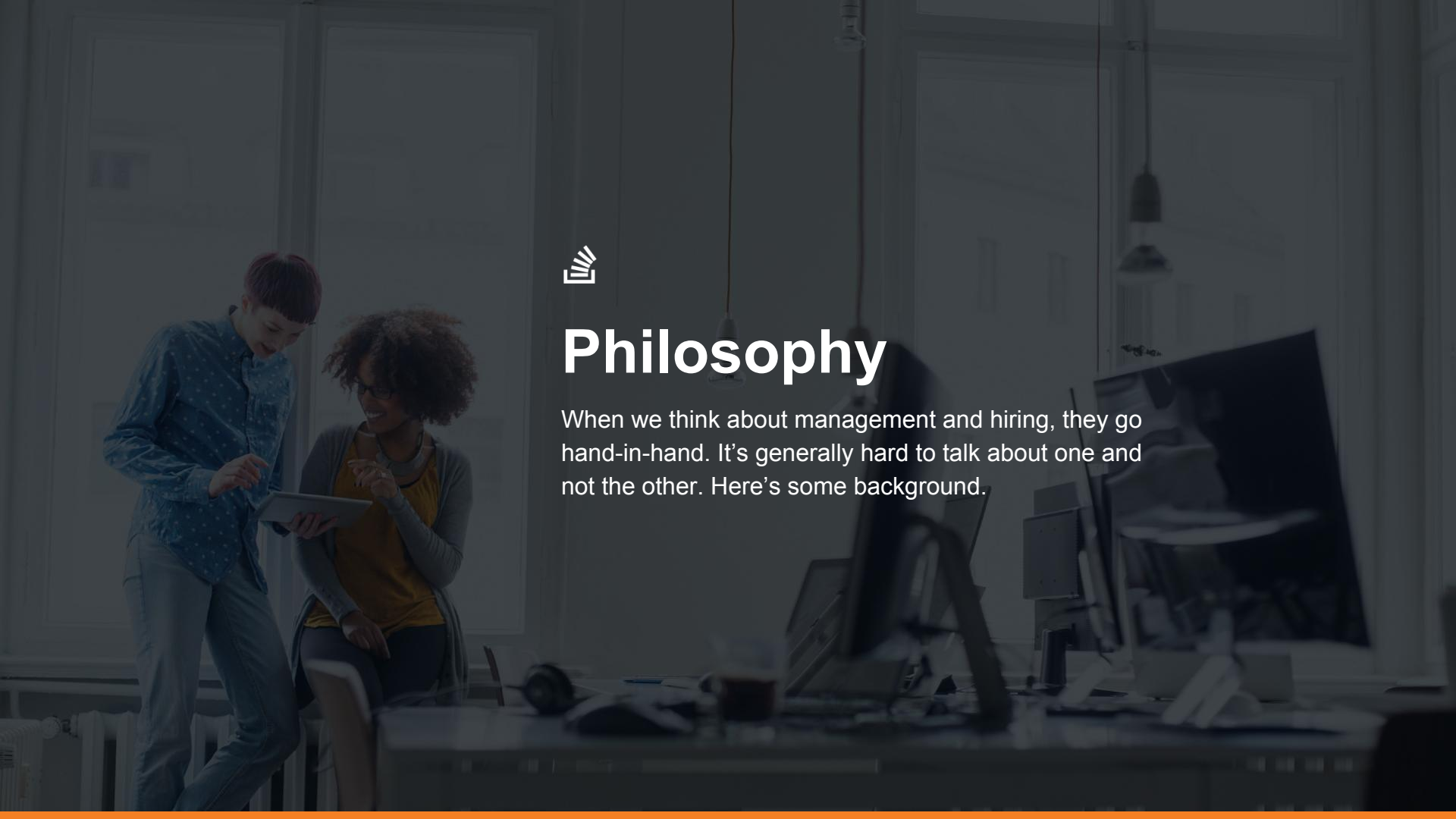
Now that you have developers, how do we track their performance and think about how someone improves?

Questions

`DELETE questions;`

There's a lot to cover here, but there's a lot I probably didn't answer just yet. Ask me anything, but I can't give you reputation.

@jonhmchan



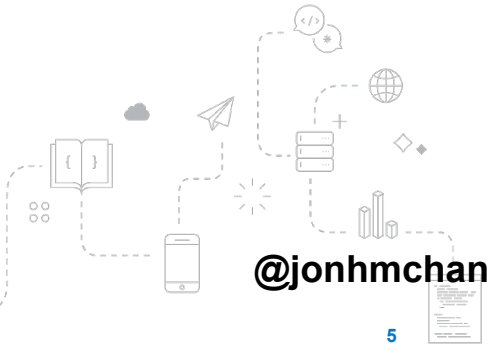
Philosophy

When we think about management and hiring, they go hand-in-hand. It’s generally hard to talk about one and not the other. Here’s some background.

About Stack Overflow

Who we are

- Over 250 people, **distributed** across the world
- Engineering organization as a whole with **55 people**, includes developers, SREs, design, and product
- **Three layers** of management after CEO: VP, manager/team leads, contributors

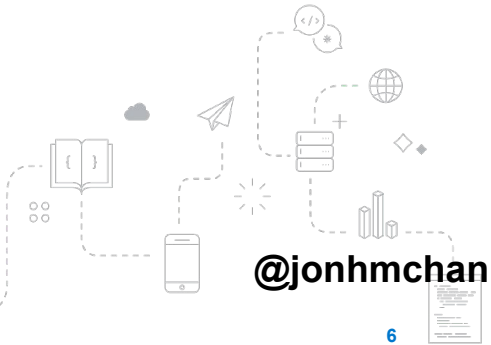


@jonhmchan

Pluralism

Management is a support layer

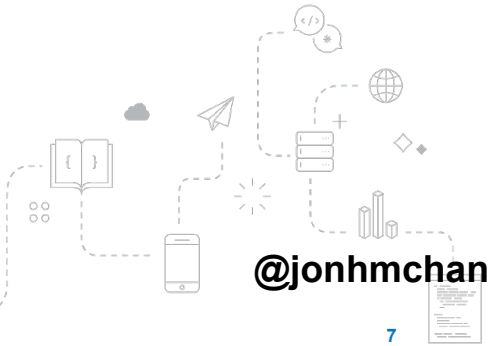
- **Joel Spolsky** talks about the history of management
- The ultimate thing to understand here is that management is a **support role**
- We loosely call this philosophy of management **pluralism** at Stack Overflow



Empowerment Organization

My personal interpretation of Pluralism

- You can put people into three buckets based on their ability to empower: **force multipliers, experts, net negatives**
- There are generally two **dimensions**: your skill in your craft and your ability to manage
- **Not specific** to engineering
- What that means is we **trust our developers a lot** to be self-driven

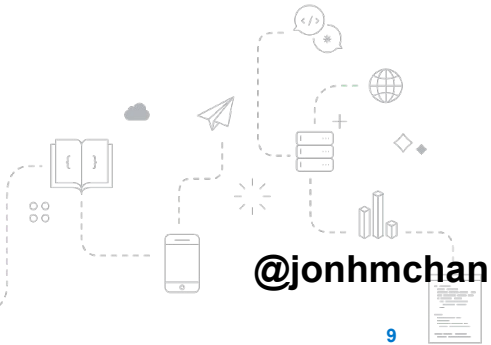



London
Denver
New York

Remote Culture

We've been distributed from day one

- Main offices are in **New York, Denver, and London**
- Quarter remote overall, development is **~65% remote**
- We really believe in remote: it vastly increases your hiring pool, flexible schedule, "private" office



@jonhmchan

Pluralism + Remote = Smart Hiring

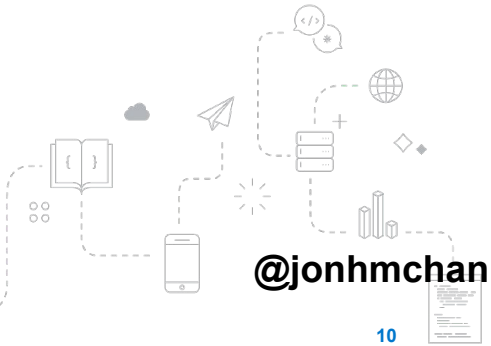
The two things we optimize for

Smart

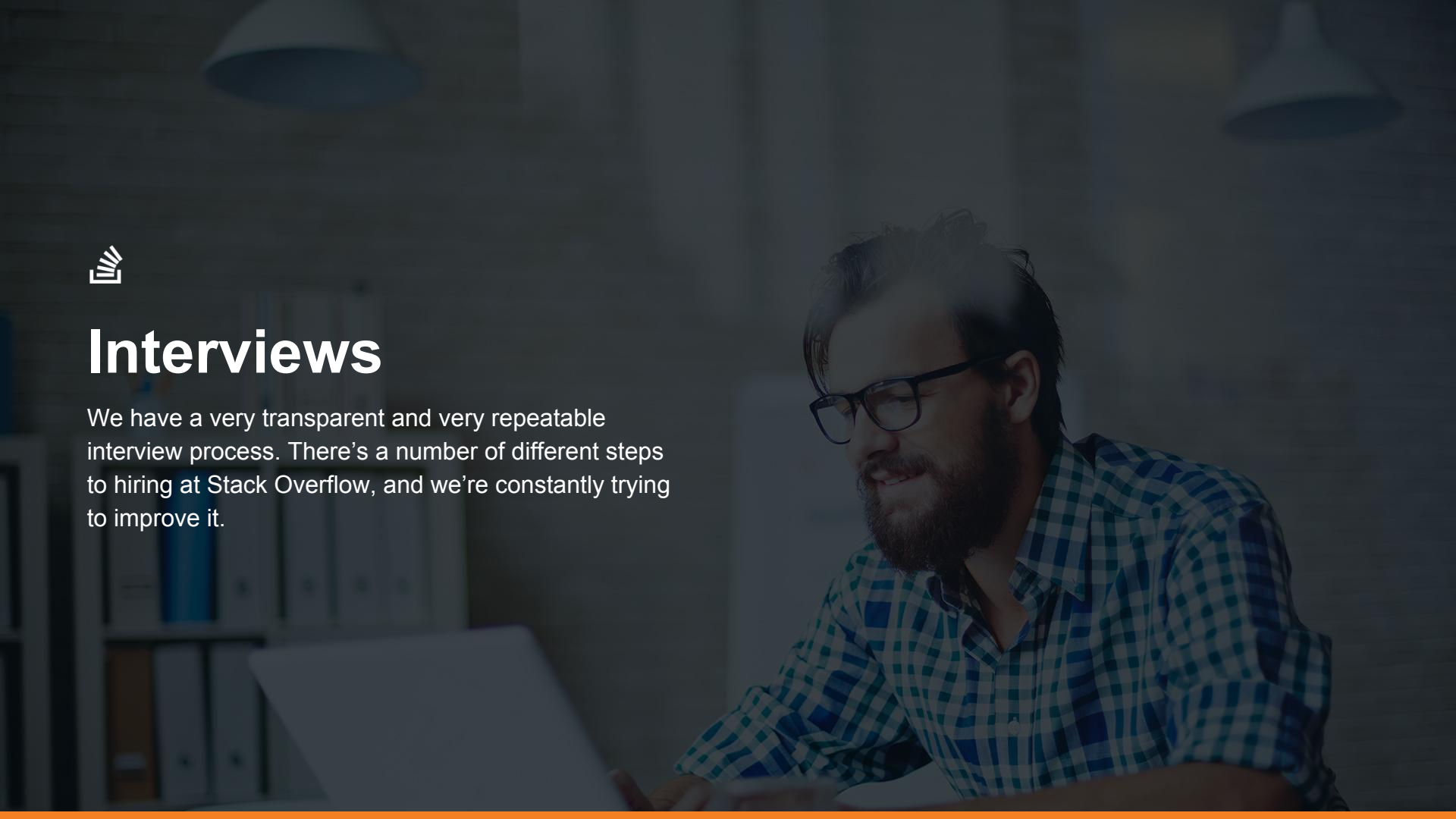
These are generally the things we interview for: you can usually tell from chatting with someone. Also what the technical parts of your interview are about (i.e. coding)

Gets Things Done

Are they self-driven, starters and finishers, and the person people ask questions to? Best indicator of “Gets Things Done” is a history of having gotten things done.



@jonhmchan



Interviews

We have a very transparent and very repeatable interview process. There’s a number of different steps to hiring at Stack Overflow, and we’re constantly trying to improve it.

Hiring Philosophy

Things we keep in mind

Binary Hire

HIRE OR NO HIRE

Unless it's a strong hire, we don't hire at all - no "maybe" hires. We think it's better to let strong potential go than hire false positives.

Veto Power

NO HIRE MEANS STOP

Essentially everybody along the process has veto power. If someone gets a no-hire they are taken out of the process right then and there.

Full Stack

OPTIMIZE FOR LEARNING

We hire full stack developers (can build web projects end to end) and would be comfortable with picking up any part of that project.

Senior

BAR IS THE TEAM

We hire only what most other companies would consider senior developers. Bar is the rest of the team, optimizing for GTD.

Backgrounds

LANGUAGE AND EDUCATION AGNOSTIC

We don't mind that people don't have experience in our technology stack and we also don't mind that people don't have degrees

Developer Interviews

Our basic process

Resume Review	Basic Screen	Full Interviews	VP Interview
Recruiter Review	**Phone Screen**	**“Hard” Developer Interview**	**Team Match**
Developer Review	Code Screen	Architecture Interview	Fill In Gaps
		Product Manager Interview	**Sell the Opportunity**

@jonhmchan

stackoverflow

TEDPG

Developer Resume Reviews

Targeted

I LOVE STACK OVERFLOW

We want to make sure that developers really love Stack Overflow. Just a sentence or two on the cover letter is enough. 10K+ rep is a yes.

Experience

FULL STACK DEVELOPERS

4000 hours of full stack web development experience. That translates to about two years of full time work in the last five years.

Technical Depth

DIGGING IN DEEP

Did they do something technically interesting and difficult? Build out their own web server or their own CSS preprocessor, etc.

Passion

MORE THAN 9 TO 5

We need evidence that they are really excited about programming (i.e. 50+ open source contributions, 10K+ rep, active technical blog)

Gets Things Done

LEADERSHIP

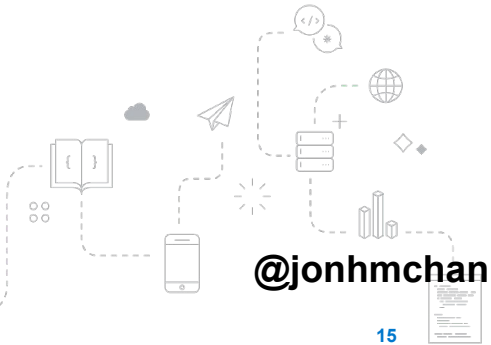
We want to see that they led or owned a project. Was this the person people asked questions to? If led a team, was it technical?

@jonhmchan

Code Screen

Can this person code?

- **Introduction**
  - Who I am, how this works, think out loud
- **Warm up question**
  - They do this fast, without issue
- **Meatier question**
  - Everyone can understand, not everyone can solve
- **Questions they have**
  - I generally make no judgment on this except if there are *no* questions

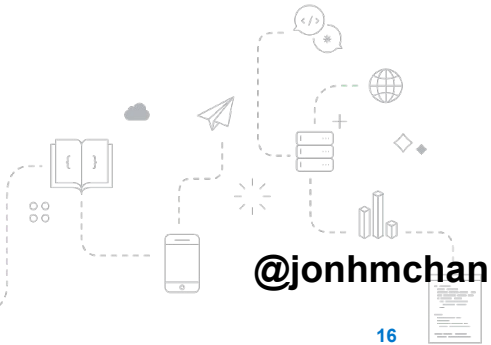


@jonhmchan

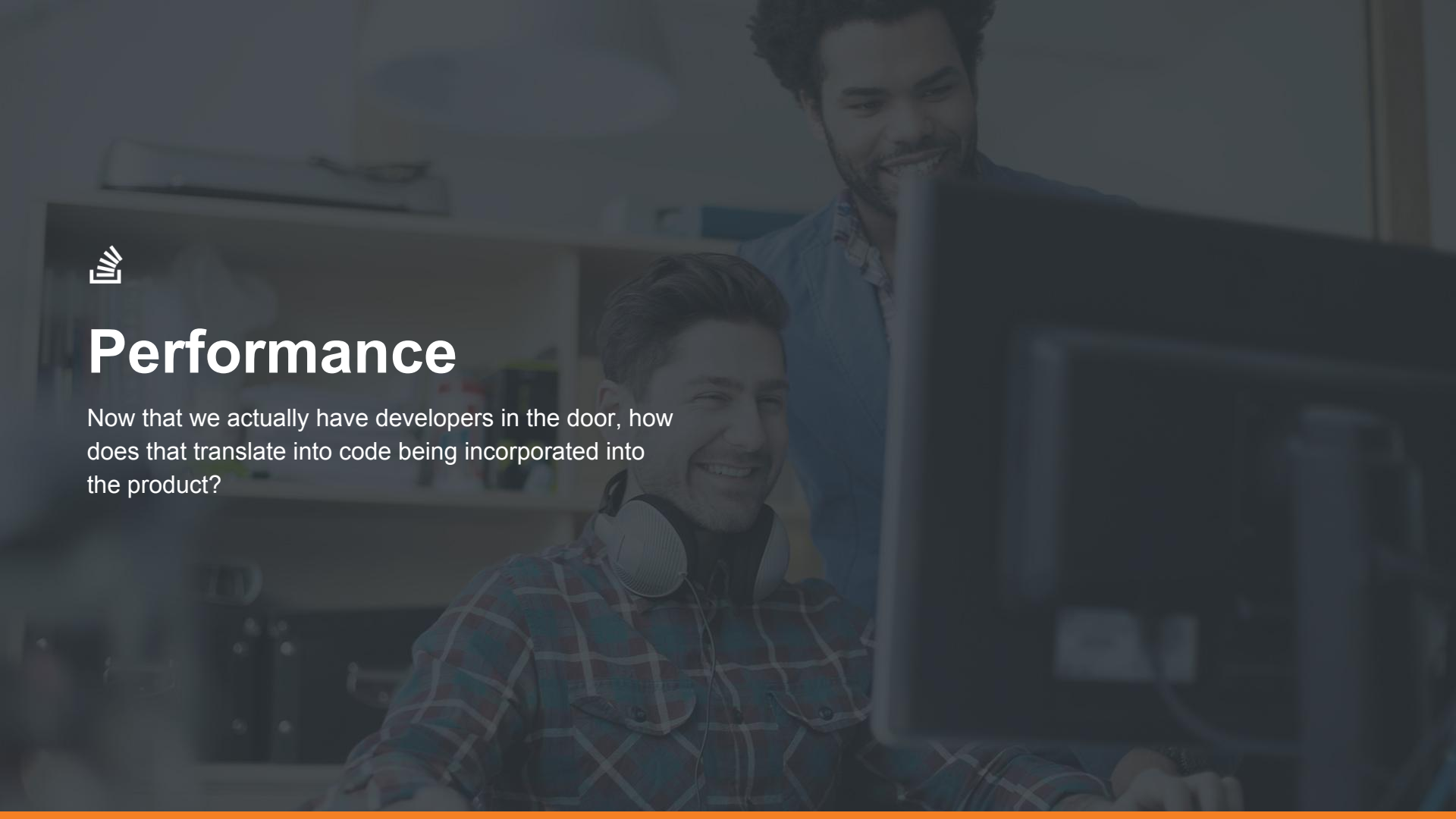
The Rest

Can this person code?

- **“Hard” Technical Interview**
  - i.e. recursion, pointers, low-level questions
- **Architecture Interview**
  - Tests for full stack understanding
- **VP Interview**
  - Team fit, fill in the gaps, sell the company



@jonhmchan



Performance

Now that we actually have developers in the door, how does that translate into code being incorporated into the product?

Now What?

How we track performance of developers

Stack University

ONBOARDING

Like our hiring process, we have a pretty thorough and repeatable onboarding process for our developers.

Weekly Updates

AGILE-LIKE

Developers are largely left to their own devices. There's a weekly cadence in how we do work with some additional PM tooling.

Salaries & Performance

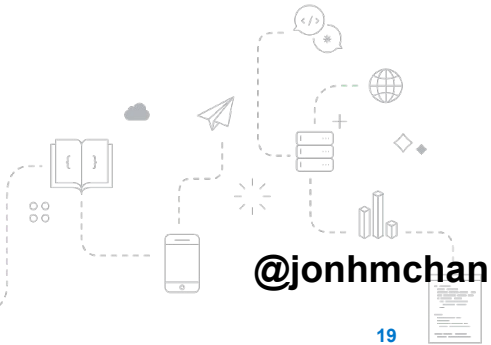
BE MORE AWESOME

We have a very transparent and standardized approach to salaries and performance. Every developer knows how this works internally.

Stack University

Onboarding Developers

- Assigned a **mentor** from your team
- **Standard checklist** that takes about six to eight weeks that covers:
  - Overall company policies
  - Machine setup and day one push
  - Optional “Ping Pong” app
  - 30 minute chats
  - Bug Duty
  - Graduation Project

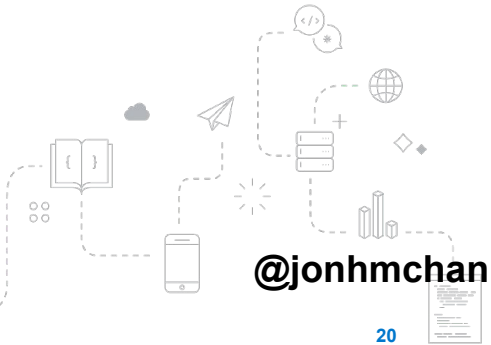


@jonhmchan

Weekly Updates

Our “agile-like” approach

- We have an “agile-like” approach to product development, on a **weekly** basis
- Standard **format**: what did you do last week, what you’re doing this week, discussion items
- Some mixture of **Google Docs** and **Trello** to organize these updates
- Additional **PM driven** processes are used, mostly in Trello



@jonhmchan

Weekly Updates

Status Document Example

LAST WEEK

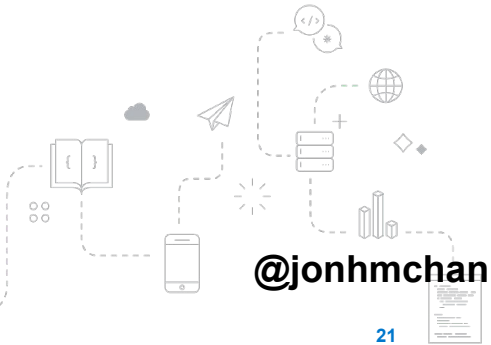
- (Jon) Built out feature X for project A
- (Jon) Gave a talk about diversity & inclusion

THIS WEEK

- (Jon) Two developer interviews
- (Jon) Internal testing for feature X

DISCUSSION

- (Jon) Tabs vs spaces

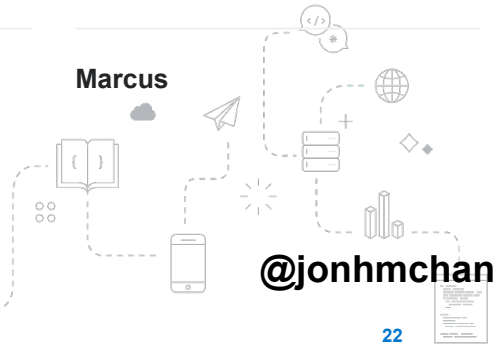


@jonhmchan

Weekly Updates

Trello Board Example

FOR CALL	WORKING ON	DONE	ROUND ROBIN
(Jon) Our Fall 2016 Roadmap	(Jon) Blog migration	(Jon) Localization popup	Jon
(Marcus) ESP Evaluation	(Tiffany) Events pages	(Tiffany) Mobile fixes	Tiffany
		(Marcus) React evaluation	Marcus

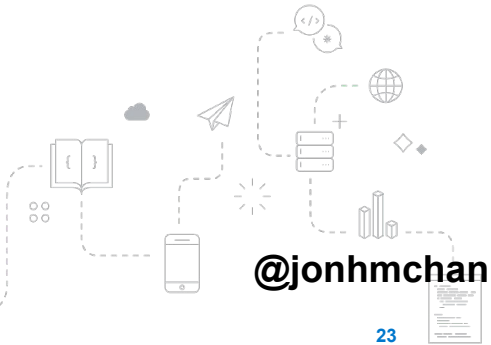


@jonhmchan

Salaries and Performance

Our “Be More Awesome” Chart

- Standard rubric to assess someone’s skills known as our **Be More Awesome** chart (BMA) every six months
- Each skill is graded from **B to A+++**
- Skills are plugged-in to a formula with **experience**, **role,** and **adjustments** to calculate salary
- **Completely transparent** and even released soon



@jonhmchan

Be More Awesome

“All Engineering” Sample

Iteration

Implements quickly and correctly. Demonstrates regular, incremental, and visible progress. Avoids coupling and over-architecting. Adjusts well to feedback and changing priorities.

Public Artifacts

Creates publicly-viewable artifacts intended for the benefit of others. Examples: participates on Stack sites (including meta), writes blog posts, contributes to open source or apps, speaks at meetups or conferences.

Self-Motivated Learning

Learns diverse technologies, techniques and topics out of curiosity. Dives deeper into known stacks. Uses learning to improve our code and processes.

Ideas

Consistently coming up with new, useful ideas. Asks questions and fights status quo bias. Thinks independently. Creates new RFCs and constructively participates in others’.

Communication

Conveys concepts, suggestions, and goals within and across teams. Is proactive about keeping others up to date. Articulates and persuades. Doesn’t “fall off the map”. Uses tools (Trello, chat, calendar, email, docs...) effectively.

@jonhmchan

Thanks!

Any questions?

You can find me at @jonhmchan & jon@stackoverflow.com