Zero to SRE

Kim Schlesinger

@kimschles

Your company can transform a junior engineer to a mid-level engineer within one year.



Kim Schlesinger

@kimschles

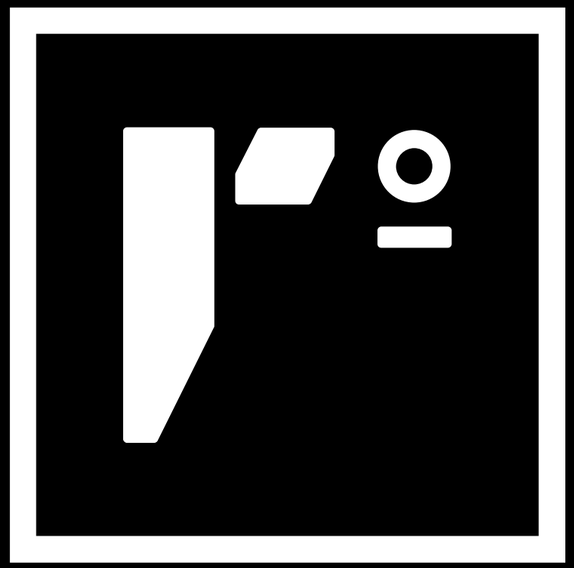


@kimschles



ersity

@kimschles



Reactive Ops

@kimschles



My Journey to Site Reliability Engineering



Steve Kinney
@stevekinney

Follow

Being able to take junior engineers and turn them into badasses is a competitive advantage.

10:14 AM - 19 Sep 2018

19 Retweets **100** Likes



2 19 100

Zero to SRE

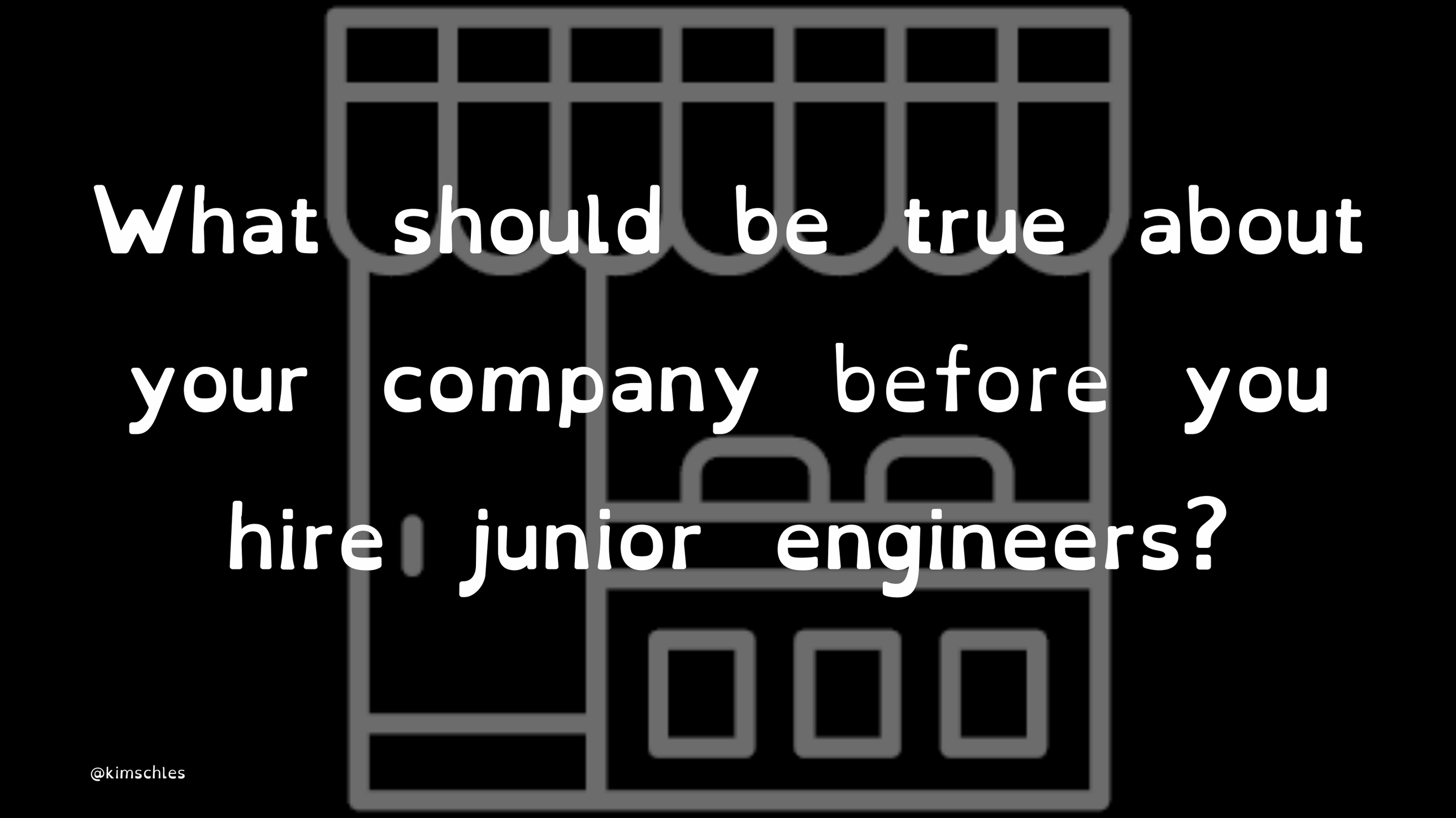
@kimschles

Agenda

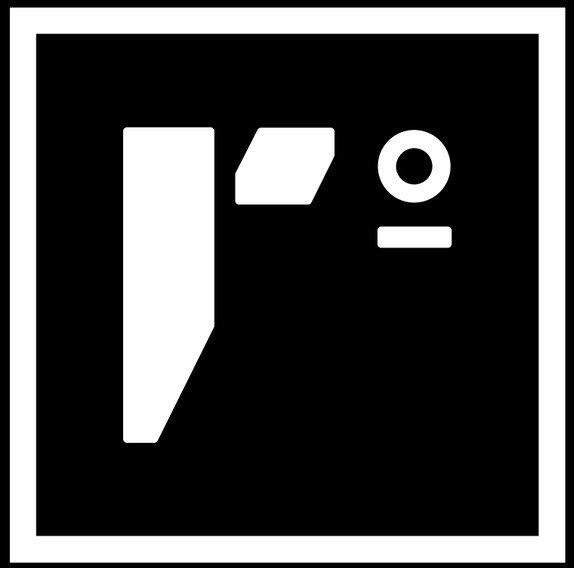
1. Your company's culture
2. Prepare for the junior SRE's day one
3. Provide them personalized resources and ongoing support



@kimschles


What should be true about your company before you hire junior engineers?

@kimschles



Reactive Ops

@kimschles

Guarantee 20% Time for Learning

(2) Read Chapter 5 of How Linux Works

in list Next Up

MEMBERS



LABELS

RO-Time

DUE DATE

Mar 29 at 12:00 PM

ADD TO CARD

- Members
- Labels
- Checklist
- Due Date

Description

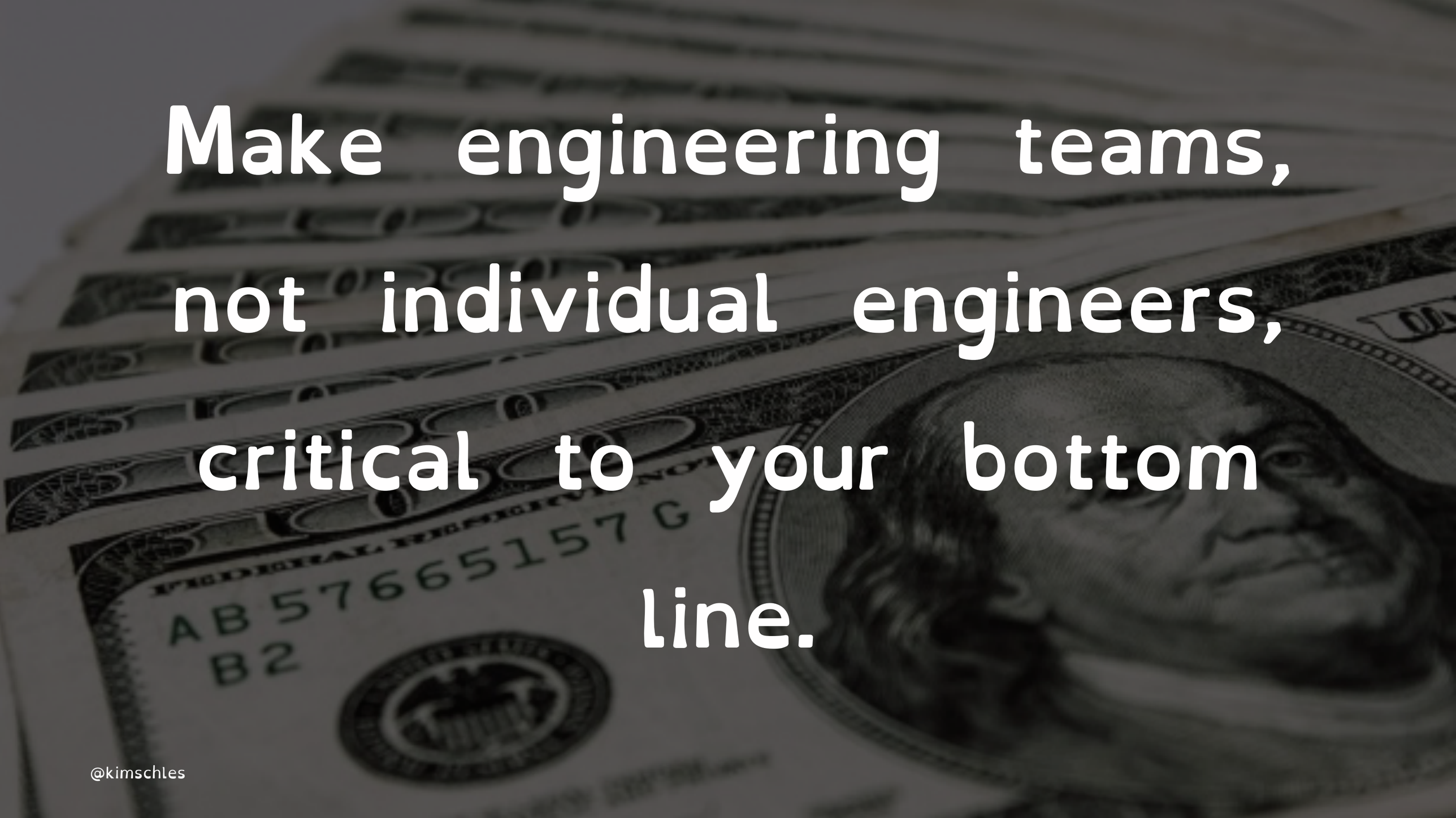
Edit

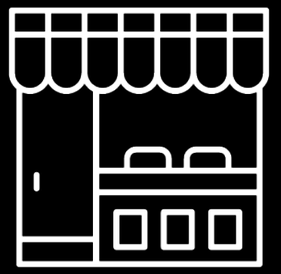
Chapter 5: How the Linux Kernel Boots

Create a Culture of Error

Expect Error and

Praise Risk Taking


Make engineering teams,
not individual engineers,
critical to your bottom
line.
@kimschles

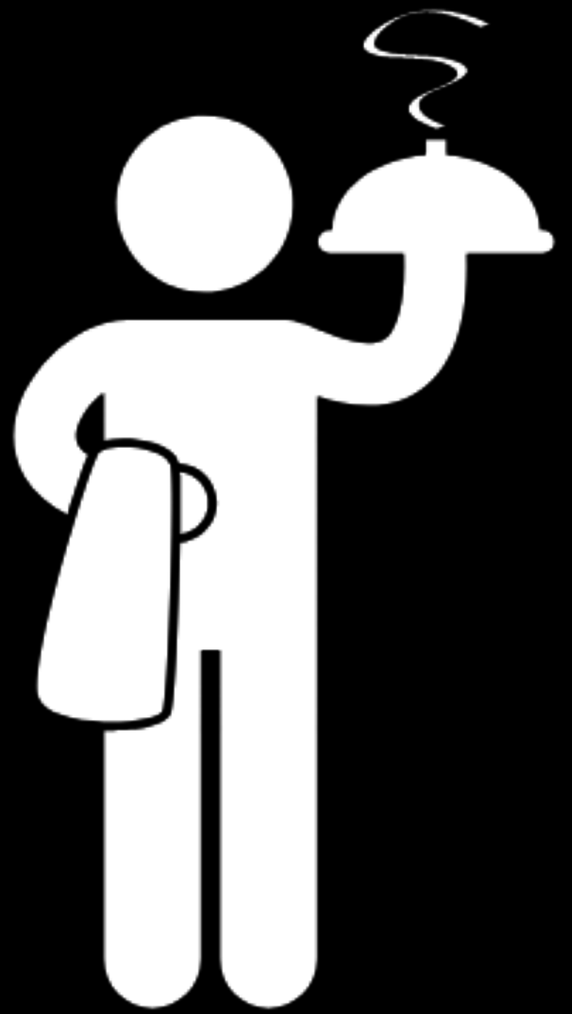


Guarantee 20% Time for Learning

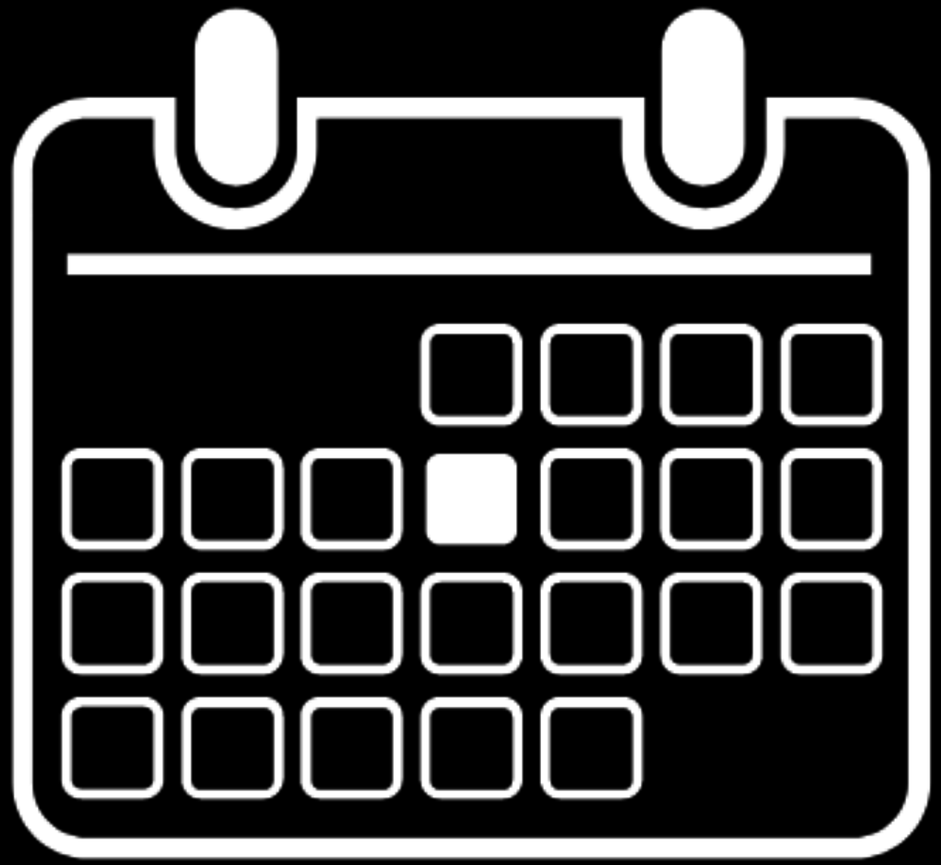
Create a Culture of Error

Leverage teams, not individuals

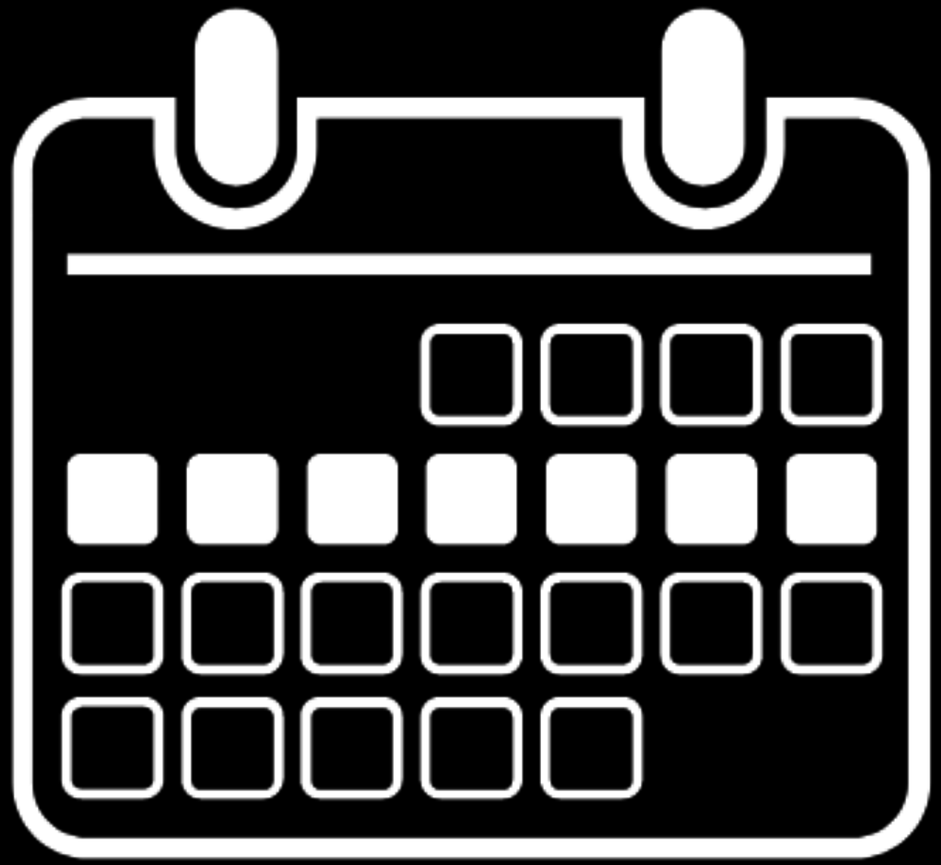




@kimschles



First Day



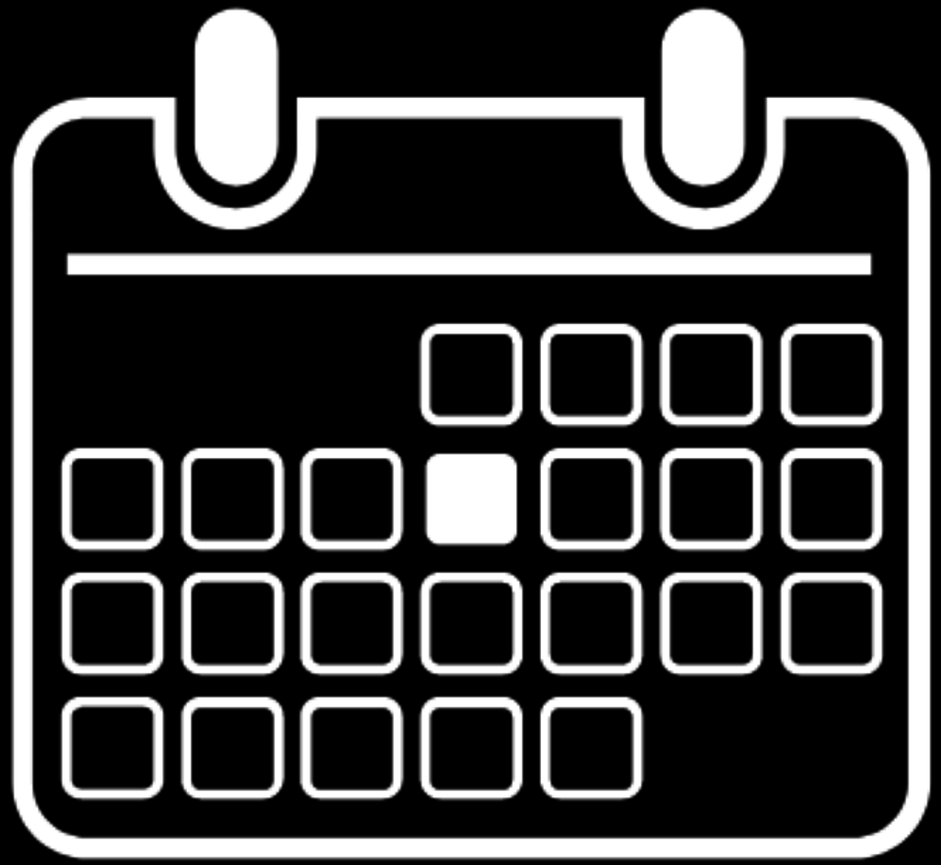
First Week



First 90 Days

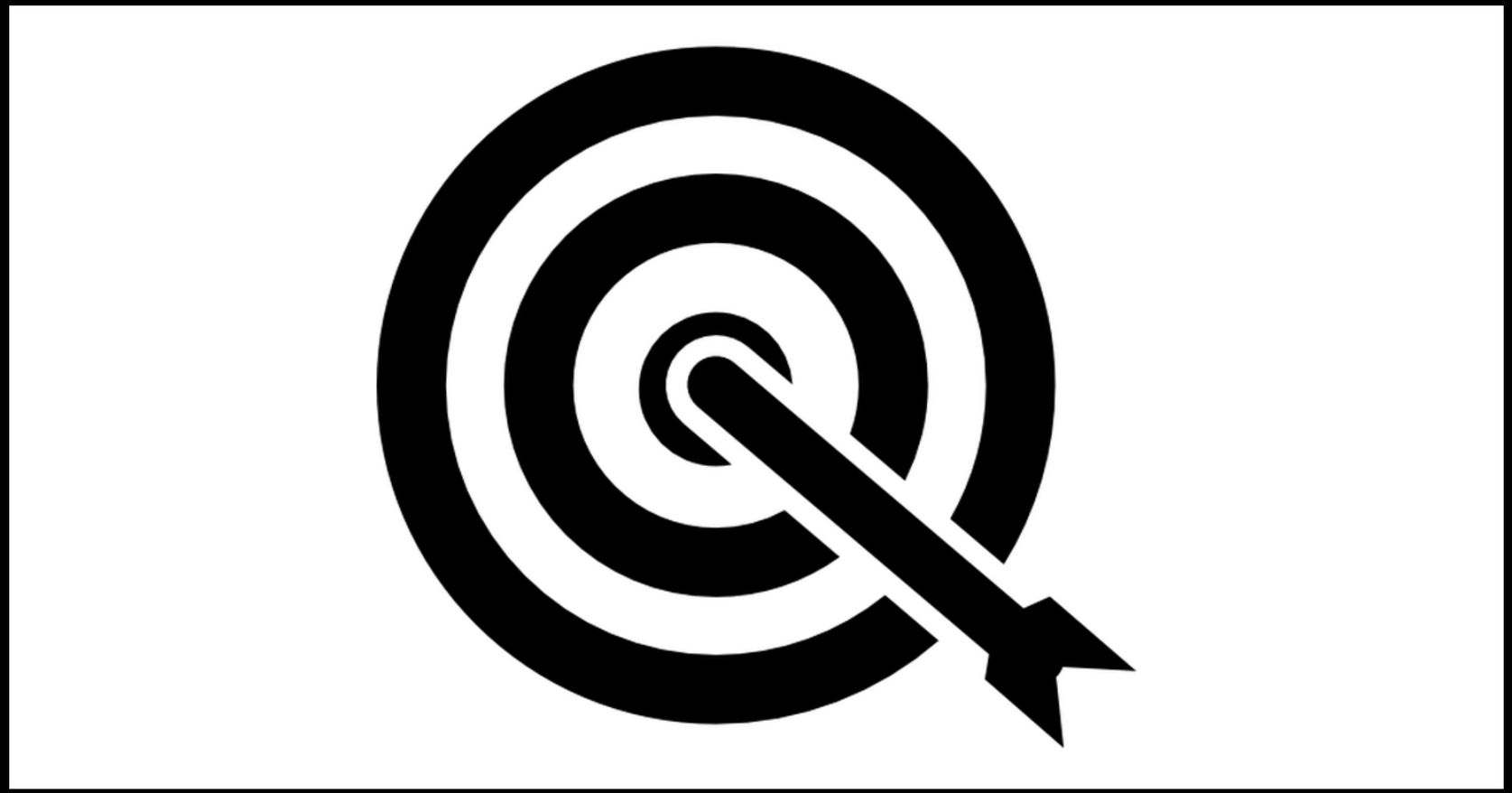


First Year



First Day

Clear and Measurable Expectations



Engineering Leveling Doc

@kimschles

INFRASTRUCTURALISTS

TRUSS

Engineering

Level	Below Expectations	Expectations	Above Expectations
1 - Novice Summary - Is not yet ready to undertake work for a client. Should pass through this level quickly.	* Fails to complete tasks as defined * Fails to communicate roadblocks in a timely fashion * Repeatedly makes the same mistakes after correction	* Completes tasks defined by supervisory personnel * Adjusts time expenditure and focus as directed by supervisor * Seeks guidance on what skills to improve and how to go about improving them * Attempts to characterize and resolve roadblocks through 5-15 min Google searching before reporting * Reports roadblocks quickly if not answered by the internet and asks for guidance * If a mistake is made, takes note and tries to avoid that mistake in the future * Actively seeks to develop & learn time/task management processes	* Anticipates when a task could use more clarification and asks for it prior to starting a task * Identifies ways to improve processes that are in place and recommends those to supervisor
2 - Advanced Beginner Summary - able to work as a junior engineer on a client team with more or less help from senior Trussel on team.	* Fails to complete assigned tasks * Fails to communicate task status * Fails to work effectively as part of a team	[All of the above, plus...] * Completes technical tasks in existing areas of expertise with some guidance * Breaks moderately-sized tasks down into smaller units of work upon request and reviews break-down with senior engineers * When encountering a new problem space, develops an idea of what there is to learn and where to start based on consultation with experts * Seeks to understand and employ collaboration practices including, but not limited to: working in sprints, collaborative git workflows, listening well in conversation, attempting to come up with hypotheses for how problems might be solved	* Begins facilitating team collaboration * Begins independently discovering and solving technical / process problems

Engineering

Level	Below Expectations	Expectations	Above Expectations
1 - Novice Summary - Is not yet ready to undertake work for a client. Should pass through this level quickly.	* Fails to complete tasks as defined * Fails to communicate roadblocks in a timely fashion * Repeatedly makes the same mistakes after correction	* Completes tasks defined by supervisory personnel * Adjusts time expenditure and focus as directed by supervisor * Seeks guidance on what skills to improve and how to go about improving them * Attempts to characterize and resolve roadblocks through 5-15 min Google searching before reporting * Reports roadblocks quickly if not answered by the internet and asks for guidance * If a mistake is made, takes note and tries to avoid that mistake in the future * Actively seeks to develop & learn time/task management processes	* Anticipates when a task could use more clarification and asks for it prior to starting a task * Identifies ways to improve processes that are in place and recommends those to supervisor
2 - Advanced Beginner Summary - able to work as a junior engineer on a client team with more or less help from senior Trussel on team.	* Fails to complete assigned tasks * Fails to communicate task status * Fails to work effectively as part of a team	[All of the above, plus...] * Completes technical tasks in existing areas of expertise with some guidance * Breaks moderately-sized tasks down into smaller units of work upon request and reviews break-down with senior engineers * When encountering a new problem space, develops an idea of what there is to learn and where to start based on consultation with experts * Seeks to understand and employ collaboration practices including, but not limited to: working in sprints, collaborative git workflows, listening well in conversation, attempting to come up with hypotheses for how problems might be solved	* Begins facilitating team collaboration * Begins independently discovering and solving technical / process problems

Novice	Is not yet ready to undertake work for a client. Should pass through this level quickly.
Advanced Beginner	Able to work as a junior engineer on a client team with more or less help from senior Trussel on team.

Engineering

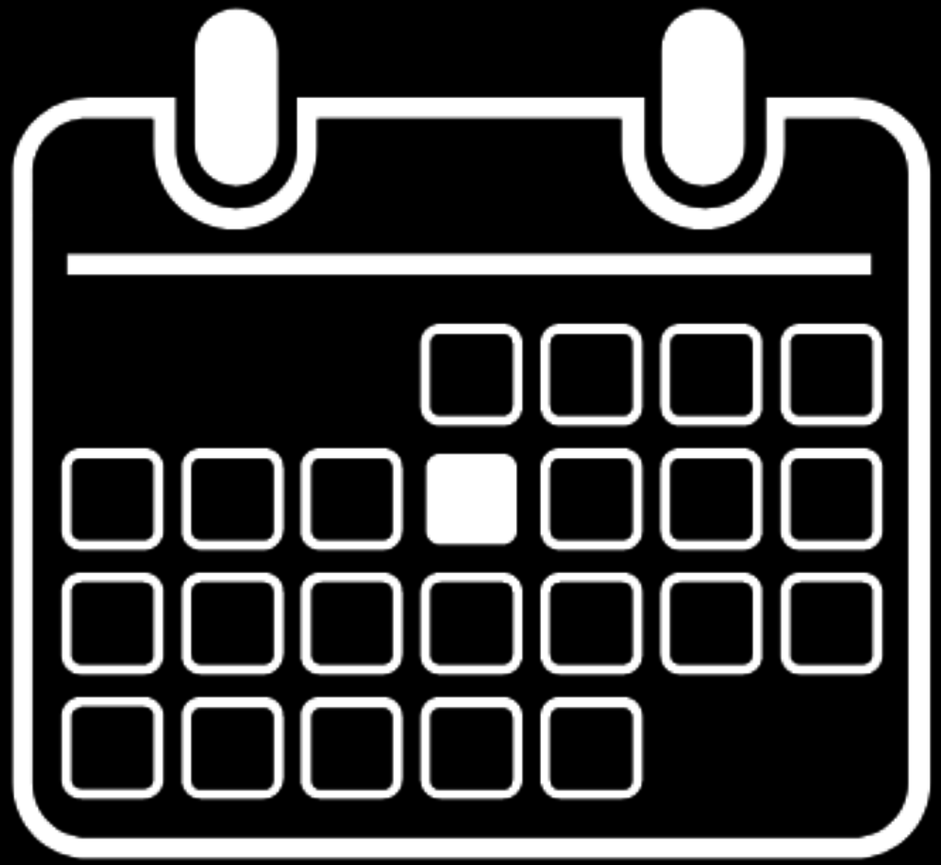
Level	Below Expectations	Expectations	Above Expectations
1 - Novice Summary - Is not yet ready to undertake work for a client. Should pass through this level quickly.	* Fails to complete tasks as defined * Fails to communicate roadblocks in a timely fashion * Repeatedly makes the same mistakes after correction	* Completes tasks defined by supervisory personnel * Adjusts time expenditure and focus as directed by supervisor * Seeks guidance on what skills to improve and how to go about improving them * Attempts to characterize and resolve roadblocks through 5-15 min Google searching before reporting * Reports roadblocks quickly if not answered by the internet and asks for guidance * If a mistake is made, takes note and tries to avoid that mistake in the future * Actively seeks to develop & learn time/task management processes	* Anticipates when a task could use more clarification and asks for it prior to starting a task * Identifies ways to improve processes that are in place and recommends those to supervisor
2 - Advanced Beginner Summary - able to work as a junior engineer on a client team with more or less help from senior Trussel on team.	* Fails to complete assigned tasks * Fails to communicate task status * Fails to work effectively as part of a team	[All of the above, plus...] * Completes technical tasks in existing areas of expertise with some guidance * Breaks moderately-sized tasks down into smaller units of work upon request and reviews break-down with senior engineers * When encountering a new problem space, develops an idea of what there is to learn and where to start based on consultation with experts * Seeks to understand and employ collaboration practices including, but not limited to: working in sprints, collaborative git workflows, listening well in conversation, attempting to come up with hypotheses for how problems might be solved	* Begins facilitating team collaboration * Begins independently discovering and solving technical / process problems

Attempts to characterize and resolve roadblocks through 5-15 min Google searching before reporting.

Breaks moderately-sized tasks down into smaller units of work upon request and reviews break-down with senior engineers



@kimschles



First Day

Prepare a technical learning plan that is aligned with the day-to-day work of your engineers.

Example Topics for the Learning Plan:

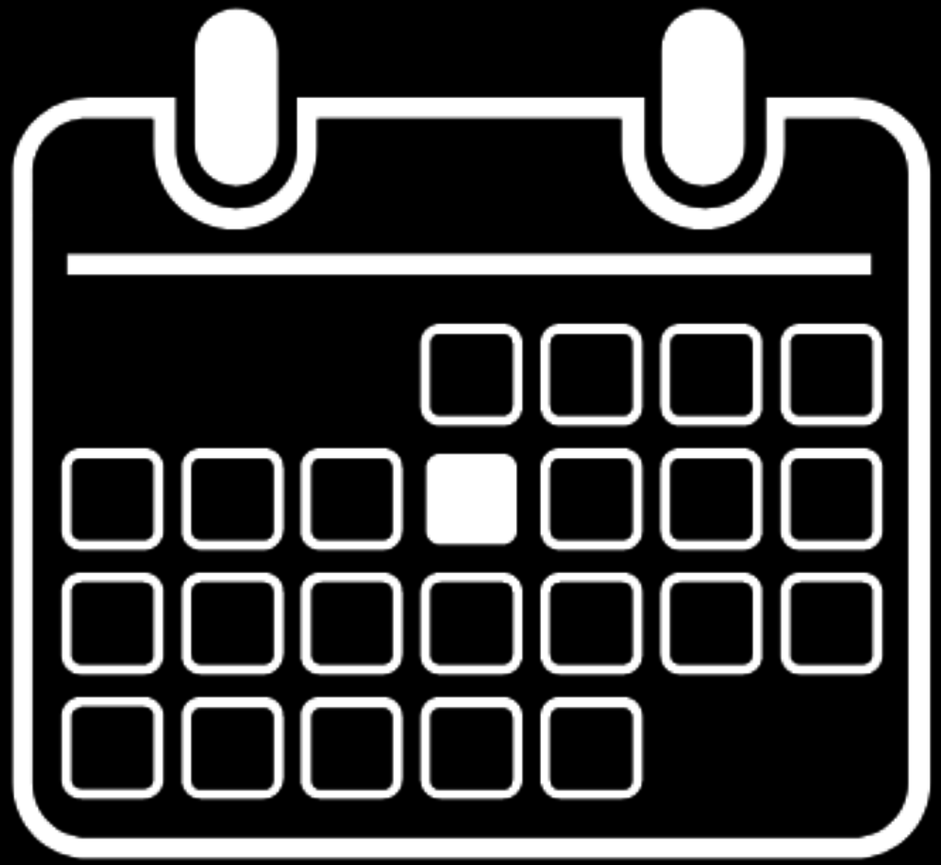
- git and linux commands
- yaml
- VPC with AWS and GCP
- Docker and Kubernetes

Your junior engineer should independently complete large and small projects as part of their technical learning plan.


Develop systems of support and accountability for your junior SRE.
@kimschles

ReactiveOps Apprentice Learning Plan

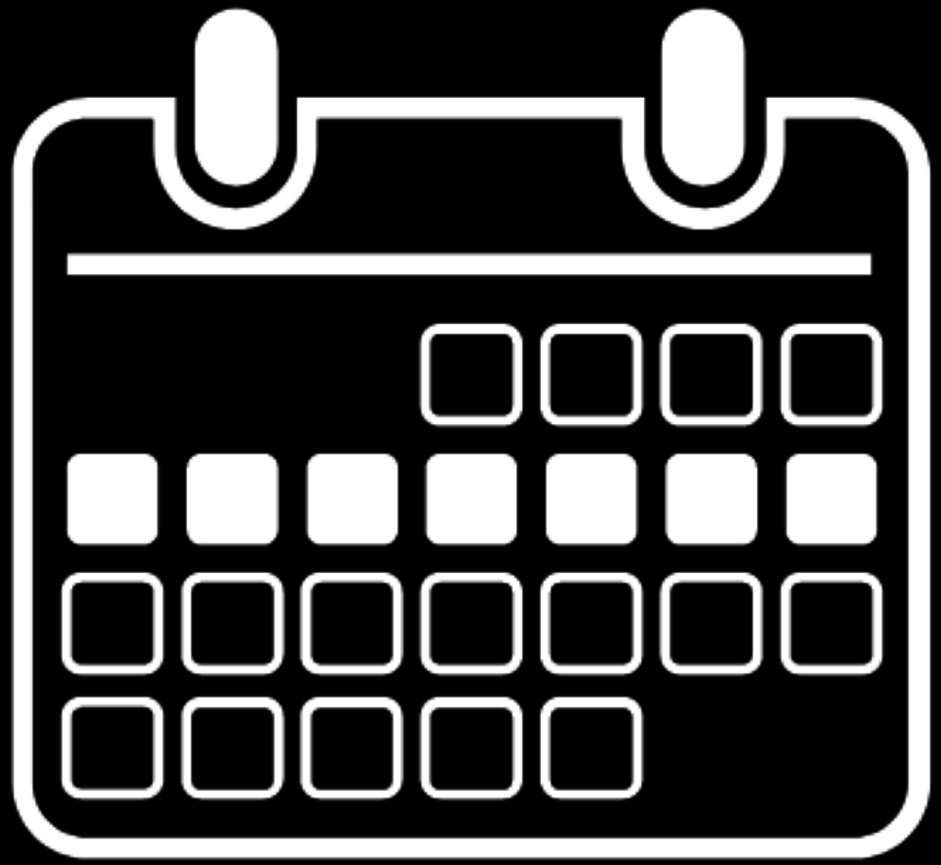
https://github.com/reactiveops/apprentice-learning-plan



First Day


Integrate your new hire into their team on day one.
@kimschles

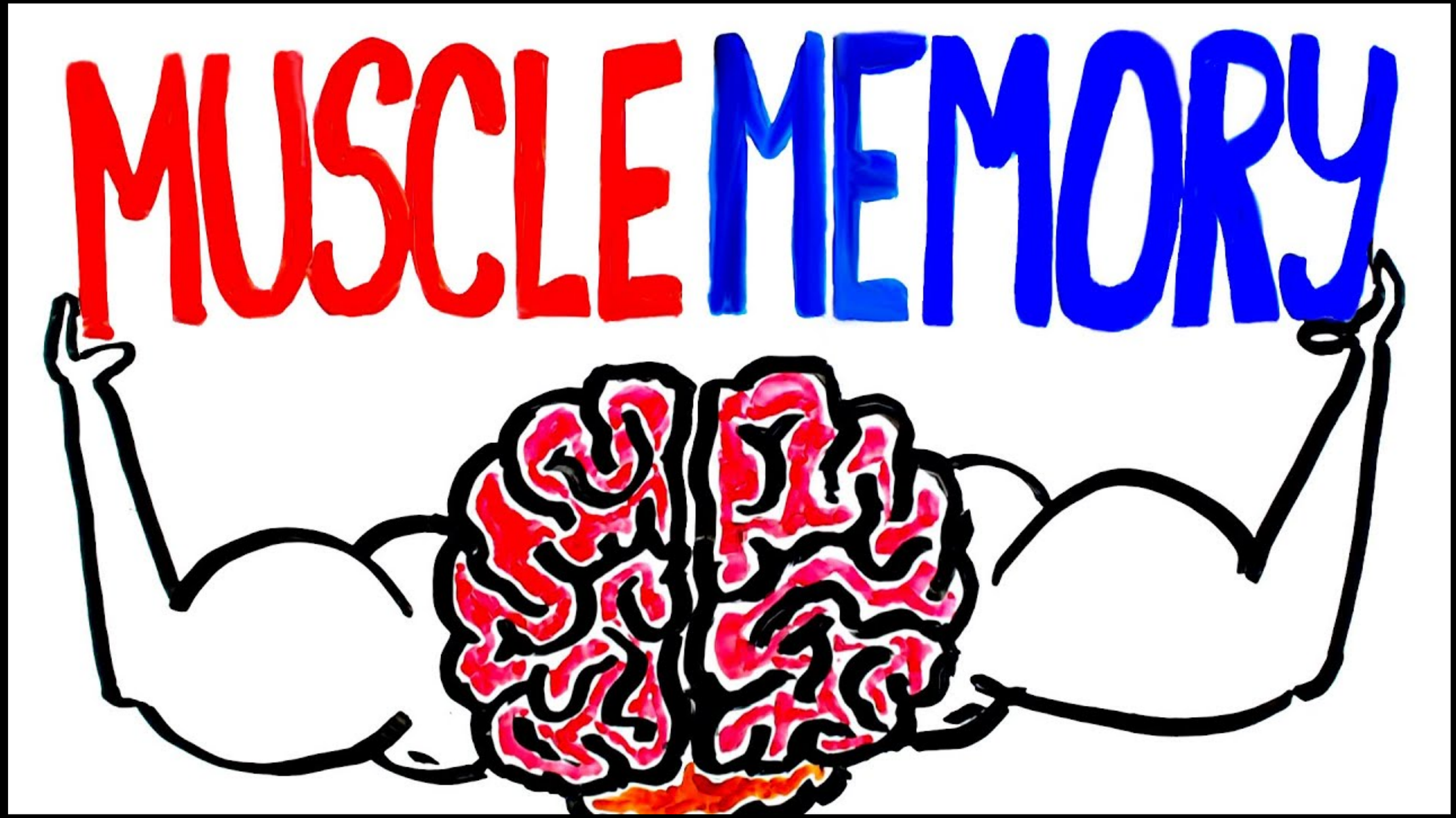
Be explicit about your desired timeline for the junior SRE's first independent contribution.



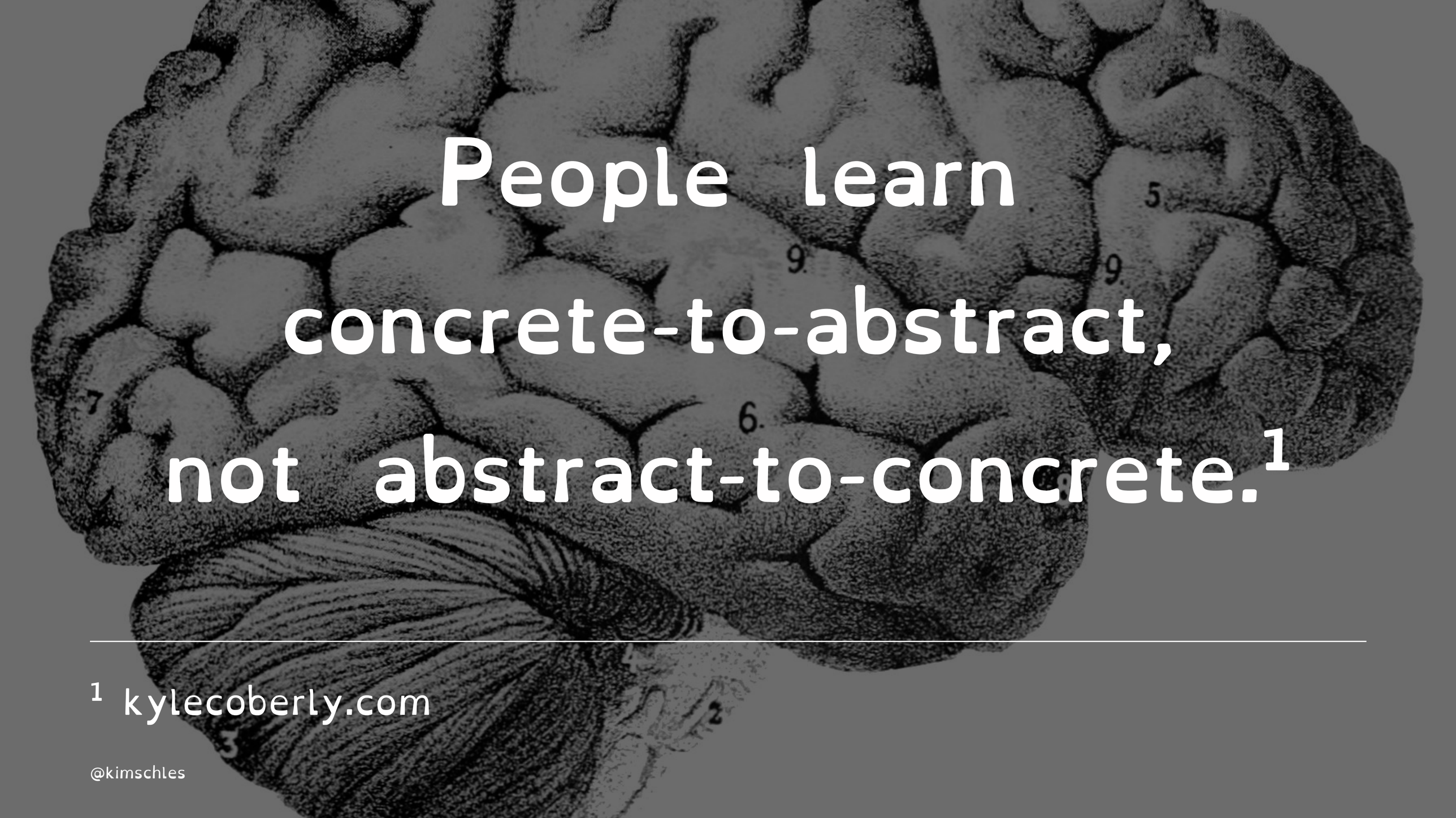
First Week



Pairing and Real Work



@kimschles



People learn concrete-to-abstract, not abstract-to-concrete.[1]

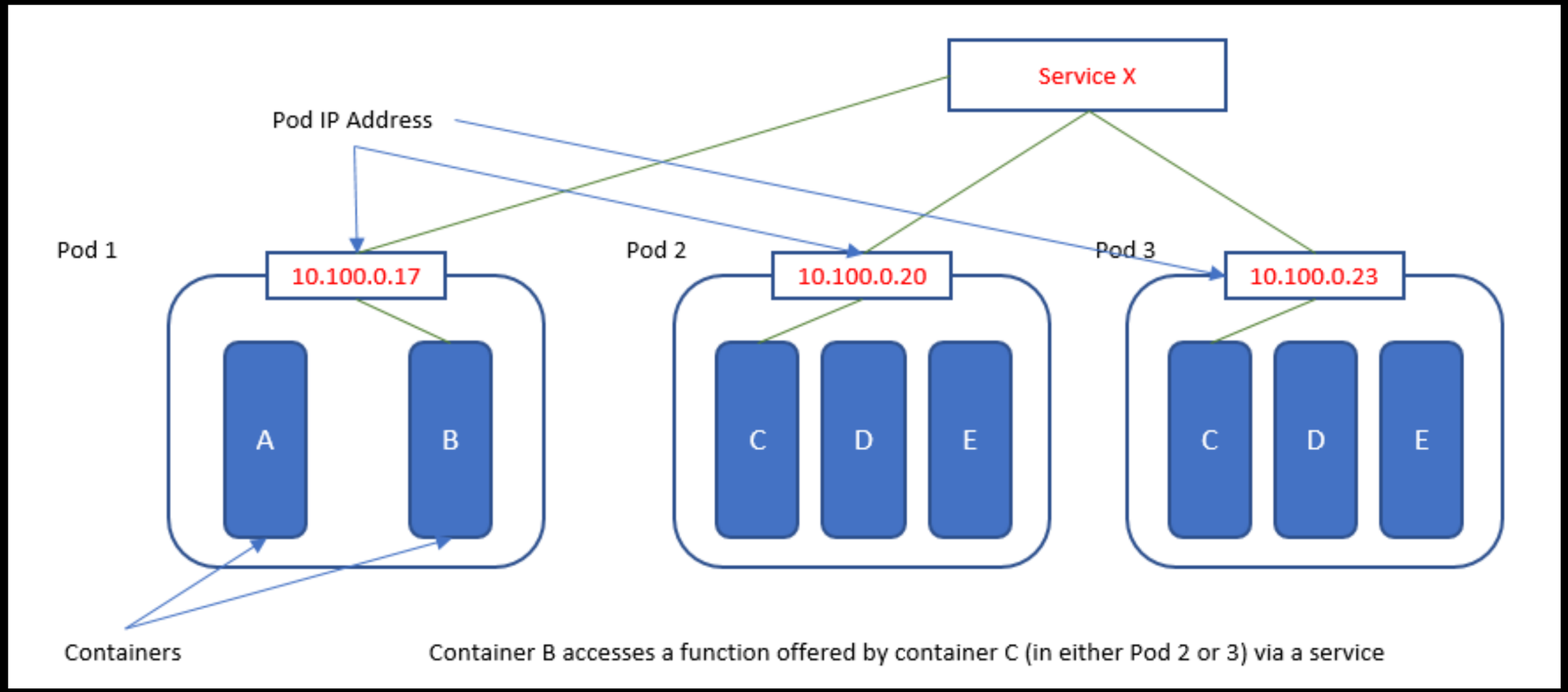
---

[1] kylecoberly.com

Concrete

```
kubectl get services -n kube-system
kubectl get pods -n kube-system
kubectl get pods -n kube-system -o wide
```

Abstract


Service X
Pod IP Address
Pod 1
10.100.0.17
A
B
Pod 2
10.100.0.20
C
D
E
Pod 3
10.100.0.23
C
D
E
Containers
Container B accesses a function offered by container C (in either Pod 2 or 3) via a service



Regular one-on-ones with their manager



First 90 Days

90 Day Trial Period

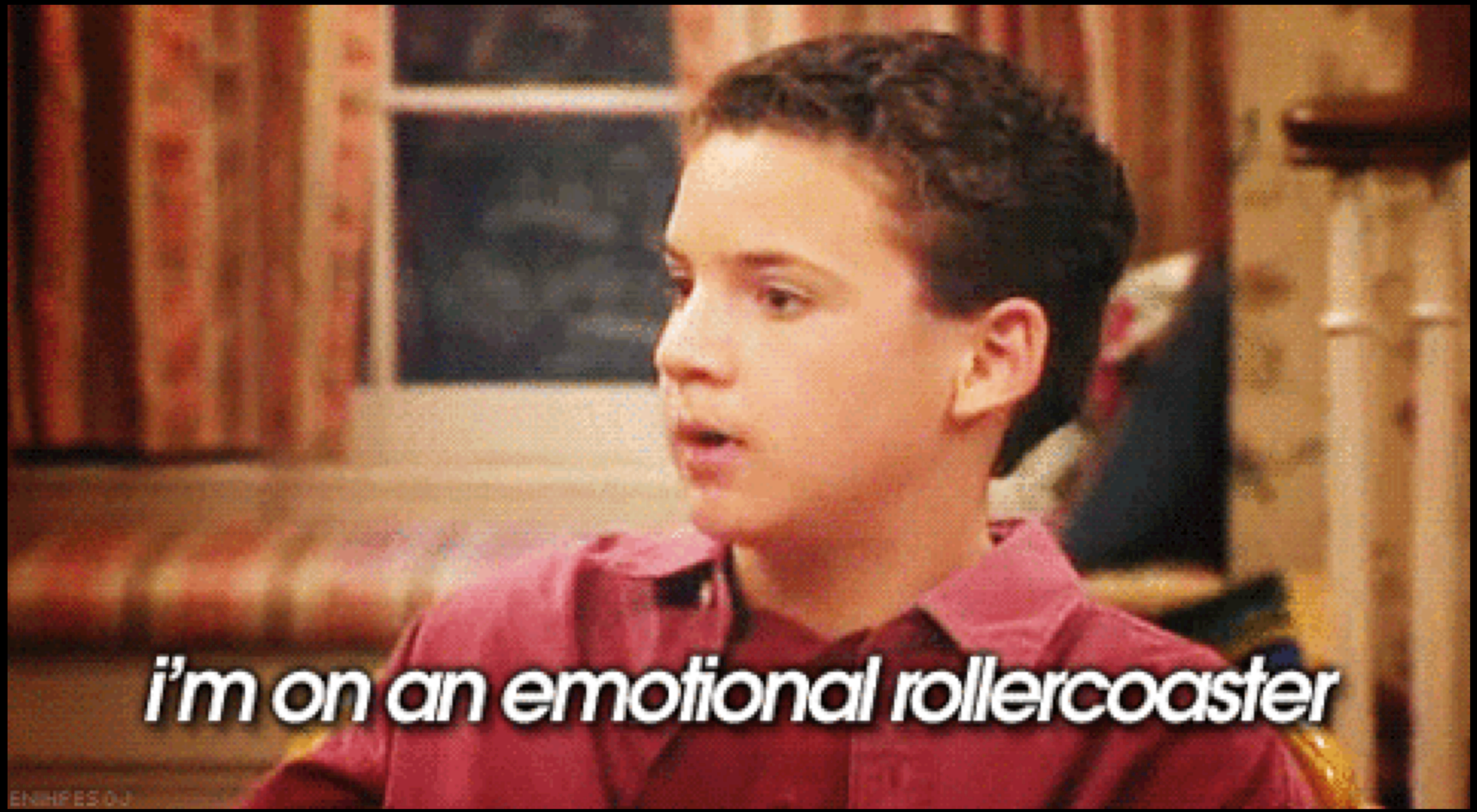
Review the new engineer's progress as compared against your leveling doc.





Encourage formal learning through exam prep, a book club, or an online course.

Hire more junior engineers!



@kimschles



First Year

Review the new engineer's progress as compared against your leveling doc.

Promote your SRE when they've leveled up!



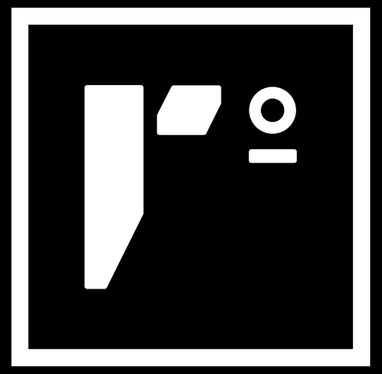
@kimschles

Your company can transform a junior engineer to a mid-level engineer within one year.

Credits

- Photos by WOCinTech Chat
- Culture of Error by Doug Lemov
- Steve Kinney
- Kyle Coberly
- Truss Leveling
- ReactiveOps Apprentice Learning Plan

We're Hiring!



Reactive Ops

https://reactiveops.com/careers



kimschlesinger.com

hirediversity.us

@kimschles