Zero to SRE
Kim Schlesinger
Your company can transform a junior engineer to a mid-level engineer within one year.
Kim Schlesinger
TEACH FOR AMERICA
<div>ersity</div>
My Journey to Site Reliability Engineering

@kimschles
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
Zero to SRE
Agenda

1. Your company's culture
2. Prepare for the junior SRE's day one
3. Provide them personalized resources and ongoing support
What should be true about your company before you hire junior engineers?
Reactive Ops
Guarantee 20% Time for Learning
(2) Read Chapter 5 of How Linux Works

Description

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.
Guarantee 20% Time for Learning
Create a Culture of Error
Leverage teams, not individuals
First Week
First 90 Days
First Year

@kimschles
First Day
Clear and Measurable Expectations
Engineering Leveling Doc
INFRASTRUCTURALISTS

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

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

https://github.com/reactiveops/apprentice-learning-plan
First Day
Integrate your new hire into their team on day one.
Be explicit about your desired timeline for the junior SRE’s first independent contribution.
First Week
Pairing and Real Work

@kimschles
MUSCLE MEMORY
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

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!
I'm on an emotional rollercoaster.
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

@kimschles
We’re Hiring!

https://reactiveops.com/careers
Thank you.