

# Navigating Jr Dev to Sr Dev

Hi, I'm Jason!

# Bootcamp Graduate

**Been in industry for just  
over 4 years**

**What does that *look* like?**

May 2014 - Graduated Bootcamp, started as a TA  
Feb 2015 - Started at MasteryConnect as a Jr.  
March 2016 - Started at Mavenlink as a Mid  
Feb 2017 - Became a Team Lead  
May 2018 - Became a Senior

**What about those  
inbetween bits?**

**My Journey**

## We'll discuss:

- 1) Looking for a Job
- 2) Jr
- 3) Mid
- 5) Team Lead
- 4) Senior

# A few caveats

**This journey is different  
for everyone**

**Jr/Mid/Senior are  
nebulous terms that can  
mean vastly different  
things for each company**

**Circumstances can  
accelerate this timeline**

**Not every company  
values the same things**

Looking for a Job

# **A few rules of thumb**

**1) Show you can learn, be  
*excited to learn***

**2) Have some projects  
you can point to and talk  
about**

**3) Don't be a jerk**

**4) Don't be a jerk**

**How did I land  
my first job?**

# Connections

**Specifically I spoke at a  
lot of meetups**

**One day... I met someone  
looking for an employee!**

**Introductions are *always*  
the best way to get a job**



**This part will feel hard...**

# *Imposter syndrome*

## Focuses:

Learning the ropes,  
feeling comfortable

**1) Learn as much as  
possible\***

*\*this may include learning  
on your own time*

# Lots of options

- Courses
- Books
- Code katas
- Side projects

**This is the best time of  
your career to do these  
extracurricular activities.**

**When you're more  
advanced, just do them  
when you want to for fun!**

**2) Ask good questions**

**Whats a good question?**

**Hey Senior Engineer, I can't figure out X. Will you  
help me?**

Bad

**Hey Senior Engineer, I'm having a problem with X.  
It's supposed to be doing Y, but its not.**

Better

**Hey Senior Engineer, I'm having a problem with X.  
It's supposed to be doing Y, but its not. I think it  
might be because of Z. Will you help me?**

Best

**3) Ship some code**

- get confident writing code
  - solving problems
  - learning patterns

# 4) Find a mentor

I've had several mentors  
over the years, and each  
has taught me something  
different



# Ryan "Holdy" Holdaway

- Software Engineer at Mastery Connect
- Taught me that I knew more than I thought I did

# James Thompson

- Principle Software Engineer at Mavenlink
- Taught me about good design and working with the business

# **Andy Leavitt and JB Steadman**

- Director of Engineering and SVP of Engineering at Mavenlink
- Taught me about running teams and being pragmatic

**Find mentors that will  
help you learn what you  
need to at your level**

# Things to avoid

# 1) Paralysis

**2) Being *too hard* on  
yourself**



**Focus:**

**Be productive, build a  
name, work well with  
others**

1) Find something and  
own it

# **Some things I've owned over the years**

- Introducing React
- Intermittent Failures
- Production exception tracking

**2) Solve pain points for  
the team**

**Think of something  
annoying to you...**

... it's probably annoying  
to others too!

# An example

# Deploy emails

**10 pull requests go out...**

**Each pull request may  
have a user facing  
change...**

**... and developers might  
not have written a blurb!**

**Compiling this email is  
painful**

# Automated it!

**And there was much  
rejoicing**

The kicker?

**It was fun, not too  
difficult, and made me  
seem like a wizard!**

**3) Build notoriety**

**Mid Engineer gets shit  
done!**

**This is good!**

**4) Start to teach others**

**One of the best way to  
learn**

**More and more, this is the  
value that you add**

**5) Start to think of the user and collaborate with the business**

# Things To Avoid

# 1) Cockiness

# Rockstar syndrome

**Remember don't be a  
jerk?**

**Nobody wants to work  
with or promote a jerk**

A fork in the  
road?

# Management **vs** Individual Contributor

***Don't necessarily have to  
decide.***

**At some companies**  
**senior = team lead**

**I've enjoyed both roles**

**Team Lead**

## **Focus:**

**Be reliable, keep your  
team happy and growing,  
develop new processes**

*psst this role may be  
more rewarding for you  
than an IC*

**1) Grow your team**

**Let them spread their  
wings**

**Help them be effective  
and build their notoriety**

**The success of your team  
is your success**

**2) Estimate effectively  
and communicate often**

**3) Not just collaborate  
with the business,  
partner with the business**

**Solving problems for  
users is the whole point  
of software engineering,  
not- the fancy code you  
write**

**As a team lead you must  
understand this, and help  
your team understand it**

# 4) Experiment

**Constantly improve your  
teams process**

**Ship those processes to  
other teams**

**Nurture and accept  
processes introduced by  
members of your team!**

**Always try out a  
suggestion, even if you  
don't agree. Prove it  
works (or doesn't!)**

# 5) Deliver results

**Helps if you can estimate  
effectively and  
communicate well**

I never "missed" a  
deadline as team lead

# Things To Avoid

**1) Trying to do everything**

**2) Taking too much credit  
for your team's work**

**3) Hurting team morale**

**4) Not being up front  
about your teams  
capacity and deadline  
pressure**



**Focus:**

**Stellar work, stellar  
mentorship**

**1) Grow those you work  
with**

2) Be *the guy/gal*

**Reputation, reputation**

**...as a good engineer**

**...as a good mentor**

**...as a good human to  
work with**

**3) Be effective wherever  
you are**

**As I rotated off of team  
lead, I was put on new  
projects that were  
struggling**

**My job was to move those  
projects forward**

**4) Give technical  
guidance, but try not to  
build it all yourself**

**Remember that you want  
to level up others too!**

# Things To Avoid

**1) Building it all yourself**

# 2) Disappearing

**How to know  
when its time  
to move on?**

# Three questions

**1) Am I still learning new  
skills?**

**2) Is there opportunity  
for advancement?**

**4) Am I being  
compensated fairly?**

If the answer for two of  
those things is no, it  
*might* be time to move on

**Usually those are  
negotiable; a good  
company wants to invest  
in you. That's been my  
experience at Mavenlink**

# Random Tidbits

**Try to be as full stack as  
possible**

**Change teams frequently**

Try to keep it fun

**Give back to the  
community**

**Don't focus too much on  
the trendy stuff**

# Resources

The Software Engineering Job Ladder - Blog Post

Clean Coder - Book

Pragmatic Programmer - Book

Questions?