# Preparing for Technology Roles and Opportunities

UCLA Extension
Coding Boot Camp
October 2016

# I am...

John Shiple

LinkedIn.com/in/JohnShiple

John@K2Teams.com

- (Former) Developer
- CTO and Technology Advisor

# What Will We Talk About?

- Common Industry Pain Points and Hot Buttons
- When I began my career, I wish I knew...
- Know Your Value
- Vet Companies Harder Than They Vet You
- Leverage Past Experiences to Carve Out Ideal Job
- Look for Feedback Loops
- Skill Track vs Management Track
- The Slippery Slope of Early Stage Startups
- The Dastardly Downside of Enterprise/Fortune 500
- How to Stand Out
- Where to Look for Jobs
- Networking Tips
- Recruiters
- Code Words
- Common Code Camp Problems and Perceptions

# Common Industry Pain Points and Hot Buttons

- Finite resources
- Cost, speed, quality tradeoffs
- Operational bottlenecks
- Old/dated infrastructure
- No infrastructure/support
- Time to market
- Holes in the team
- Inability to extend the product
- Lack of feedback and transparency
- Missed deadlines
- Details lost, not tracked
- Bad developers/analysts/marketers/sales people
- Build it fast or right?
- Build it or buy it?

# When I began my career, I wish I knew...

- **#1 Response: How to find a good mentor**
- Learn About and **Train Your Gut**: Learn to trust and tune your instincts and react accordingly
- **Trust, But Verify**: Trust others until proven wrong - but also check in regularly to feel comfortable that the work is going well and there are no issues
- **It Takes a Long Time** (to do anything of substance)
    - There is little value in predicting the future, especially the more distant that future gets
    - If you go "hot," by the time you achieve something good, it's probably not hot anymore
    - Don't rely on the forecast accuracy. Instead, determine if the development can be sustained for that length of time (Likelihood = 1/(deltaT^2))
- How equally important **career planning** is to job performance
- How important it is to **'manage' relationships** of your stakeholders (peers, managers, and subordinates)
- **Communication skills** outside technical circles tend to be a bit more tactful and gentle but a lot less effective as a result
- Hard-to-explain "**bad feelings**" about companies, vendors, or partners before the start of a business relationship turn out to be justified (tune your gut instincts!)
- Beware of **underperforming or toxic employees**, vendors, or strategic partners
- The **credentials/background of the team members** and the current state of the product matter. Those surprises can create significant road blocks
- **Greater revenue / profitability = great visibility and career advancement**
- **People/Process/Technology**: The more senior the role, the more important People and Process components are
- **Never stop networking**.

# Know Your Value

Calculate Your Value

- Write down prior and desired company/industry/vertical/region
- Download salary guides:
  - Robert Half Technology Salary Guide
- Plug your information into:
  - Comparably.com
  - Glassdoor.com
  - Indeed.com
  - Payscale.com
  - Angel.co
- Adjust by region/salary/expectation
- Adjust by source (e.g. Angel.co skews startup, Robert Half skews Enterprise)

Be wary of...

- Equity!
- Negotiate equity last, after your value is affirmed
- Know your pricing sources:
  - Salary guides skew corporate (more $)
  - Startup-oriented guides skew lower
- Common and “easy” push backs

# Vet Companies Harder Than They Vet You

Your Perspective a.k.a. “Reality”:

- The world is your oyster - they need you more than you need them
- Regardless of your perspective on the economy, the economy is ‘good’ for tech people
- Take as much time as you need/can
- Don’t do it for the money. Money, as a rule of thumb, is not a prime motivator for “tech people” (even though they get paid a lot)
- Listen and tune your gut (for hype, most notably)

Your Vetting Tactics:

- Do the 101 stuff (like reading the job description and your prospective employer’s website)
- Use this simple framework: Skills, Smarts, Culture (in that order)
- Use this simple framework: People, Process, Technology
- Learn to turn questions around
    - Make sure your questions are answered
    - Use the frameworks to turn the questions around

# Vet Companies Harder Than They Vet You

Common Questions and Concerns:

- Review the role coverage of your prospective employer (your role and those around you)
- Review the level of context switching in your prospective employer
- Get clear goals and expectations. If not, listen to and tune your gut
- Look for
  - Clear and transparent process management
  - How many people had your job before you in the last year (> 1 is bad, usually)

Warning Signs:

- Be Wary Of:
  - Being pitched on a cool product  and not cool technology
  - Your gut
- Avoid Irrational Expectations:
  - Next Facebook, Twitter, Google?!?
  - Pitches Product over Tech Challenges
  - Vague (or Revolutionary!) on Resource Status (aka Funding)
  - Poor Role Coverage
  - No Roadmap Management
  - Poor Process Management
  - Poor Technology Documentation

# Leverage Past Experiences for Your Ideal Job

Functional Skills that Increase Your Value (aka 2 or 3 in ones):

- Fullstack? With a Caveat!
  - Front-End vs Back-end vs DevOps
- Project Management Skills
- Data/Big Data
- Security
- Structure and Organization - applicable to many many areas

Cross-Functional Skills:

- Project Management
- Product and User Experience Design
- Testing and/or testing oversight (QA vs UAT)
- Specific Verticals

# Look for Feedback Loops

Feedback Loops to Review:

- Velocity and ability to iterate = faster results to experiment with
- Ability to course correct individually and company-wide
- Alignment with team and business
- Support and reinforcement of the learning and building of good methods and habits (indicator of future growth and success)
- Reference check: manager, peers, and team members

Product, Project, and Knowledge Management Status:

- How to know to build versus the company goals are?
- How does company communicate as a team? What tools and processes are used? How formal or informal are they?
- How does the company “write things down”? How to earn what others have done?

# Skill Track vs Management Track

Focus on Training and Cross-Training:

- Training = increasing your skill and expertise in your core functional disciplines
- Cross-training = increasing your influence and expertise in your related functional disciplines

Level of Company Support

- Level of Enlightenment
- Training support
- Career support

Ongoing Education

- Specific Technology Skills
- Project Management
- Product Management
- Product Design
- Company-desired Skills that Align With Your Career Growth
- ?!?

# The Slippery Slope of Early Stage Startups

It's all about Expectation Management:

- Assess YOUR level of risk (not the company you are applying for's risk level)
- Determine how much of a haircut you are willing to take (5% to 10% tops)
- Determine if the equity offer (if included) is fair (AFTER you know your value)

Common Situation to Avoid:

"Work for free?"

"No, Equity!"

"Oh, work for free?"

"No, Equity!"

"Soooo, work for free?"

"No, Equity!"

*Repeat Ad Nauseum*

# The Dastardly Downside of Enterprise/Fortune 500

Again, It's all about Expectation Management:

- People
- Process
- Technology
- Responsibility
- Commute Time
- Benefits

Rationale for Enterprise is different than early stage companies and startups (e.g. stability)

Fortune 500 Feedback:

- Enterprise programmers are 'broken'
- "People come here to die"
- Ageism? Maybe. (Probably?)
- Skill Stagnation (Almost Assuredly)

# How to Stand Out

The Basics (aka You/Your Bio/Your LinkedIn):

- Business Resume (LinkedIn) vs Technical Resume
- Update your LinkedIn and past experience to reflect the job you want (aka a “Business Resume”)
    - Promote the value you provided
- Update your resume to reflect your technical skills (and present the value you provided) (aka a “Technical Resume”)
- Promote your 2- and 3-in-one skills
- You may never need your resume again...

The Details:

- Define YOUR purpose for yourself
- Write down what YOU are looking for
- Communicate intelligently
    - Review Linkedin. Your Summary is the first box at the top of your profile
- Expose your Github activity
    - Contribute to a major open-source package; ideally one your prospect uses
- Try to engage with (or speak?) at events
- Consider blogging (*shudder*)
- Don’t be overt/shameless/bad if you use these tactics only to get attention

# Where To Look For Jobs

Start with and escalate to...

- Your friends and family
- Your personal network (LinkedIn, Stack Overflow, etc.)
- Local tech communities (like WeAreLATech.com and BuiltInLA.com)
- Local Meetups (check Meetup.com)
- Job boards
    - Local vs National comparison
    - Company stage matters
- Coding Meetups
- Hackathons

Evolve to...

- Speaking at events
- Organizing events and meetups
- Participating in the right events
- Owning/curating your own open source Github libraries (that other people use...and are not school projects)
- Recruiters(?)

# Networking Tips

Event Tips

- Could be overwhelming if not appropriately “armed” or “aware” (so have a plan)
- Pitch in and help out - they always need it
- When in doubt, act as a Connector
- Bring a buddy if that helps you

Have a Plan:

- Vet and review events
  - Some are great - very techie are good
  - Some are not - know the audience
- Target tech stacks and verticals you like
- Target individuals that you want to meet or connect with
- Have simple goals (like meet that 1 person or meet 2 ‘interesting’ companies)

# Recruiters

Finding good jobs is a LOT of work

Beware of...

- Sketchy and downright underhanded tactics from recruiters (not all! Just some...)
- New tactic on the supply-side: “bait-and-switch candidates"
    - Don’t help recruiters do this, even inadvertently
- Use them, don’t let them use you

Conflicts with...

- Organic, direct-hire approach (which requires a lot of time) (and which CEOs clearly prefer)

# Code Words

- Hands on = You Code and/or Do Way More Than We're Telling You
- Revolutionary = We Have No Money
    - Corollary: any description with excessive levels of Hype means: We Have No Money
- For Equity = For Free
- Kids Right Out of School = Ageism
- Product Passion = Irrational Expectations
-

# Common Code Camp Problems

Bad Actors in the Environment

- Glut of very early-stage developers in-market (so distinguish yourself!)
- Attitude
  - Too big for your britches
  - Salary expectations relative to experience
- All coding academy profiles are curated in the same way. CVs or LinkedIn profiles are very easy to spot quickly
- Mass, meaningless endorsements and referrals from all the people in the coding school

How to Differentiate Your UCLA Experience

- UCLA is Different! Tell prospects so
- Coding schools come in different shapes and sizes. UCLA has the highest incoming requirements
  - May need to show the Toddlers are NOT raising the Infants
- Validate the leadership and curriculum of the school to demonstrate that you come from a good program (it's the most important factor)
- Demonstrate your pragmatic knowledge and discuss what happens "in the trenches"

# When I began my career, I wish I knew...

- **#1 Response: How to find a good mentor**
- Learn About and **Train Your Gut**: Learn to trust and tune your instincts and react accordingly
- **Trust, But Verify**: Trust others until proven wrong - but also check in regularly to feel comfortable that the work is going well and there are no issues
- **It Takes a Long Time** (to do anything of substance)
    - There is little value in predicting the future, especially the more distant that future gets
    - If you go "hot," by the time you achieve something good, it's probably not hot anymore
    - Don't rely on the forecast accuracy. Instead, determine if the development can be sustained for that length of time (Likelihood = 1/(deltaT^2))
- How equally important **career planning** is to job performance
- How important it is to **'manage' relationships** of your stakeholders (peers, managers, and subordinates)
- **Communication skills** outside technical circles tend to be a bit more tactful and gentle but a lot less effective as a result
- Hard-to-explain "**bad feelings**" about companies, vendors, or partners before the start of a business relationship turn out to be justified (tune your gut instincts!)
- Beware of **underperforming or toxic employees**, vendors, or strategic partners
- The **credentials/background of the team members** and the current state of the product matter. Those surprises can create significant road blocks
- **Greater revenue / profitability = great visibility and career advancement**
- **People/Process/Technology**: The more senior the role, the more important People and Process components are
- **Never stop networking**.

# I am...

John Shiple

LinkedIn.com/in/JohnShiple

John@K2Teams.com

- (Former) Developer
- CTO and Technology Advisor