EIGEN X

PTW19 PHILLY TECH WEEK PRESENTED BY COMCAST

An Insider's Guide to Getting Hired and Starting Your Tech Career

- **6:00pm - 6:10pm - have a snack, find a seat**
- **6:10pm -7:15pm - presentation**
- **7:15pm - 7:30pm – questions.**
- **7:30pm – please fill out evaluation forms**

- Brian Woelfel, Partner at Eigen X, IT consulting firm in Radnor (Salesforce, Tableau, Agile App Dev)
- 25 years experience C#, Java, Angular, NodeJS, MSSQL, Tableau, Project Management for QVC, Comcast, Lexis Nexis, Pew Charitable Trusts
- Don't know hardware, networking, or sysadmin. Not a lawyer or HR expert. Just my opinions.
- **bwoelfel@eigenx.com** (happy to answer any questions reach out)
- **http://linkedin.com/in/brianwoelfel** (connect with me, I'm linked to lots of recruiters and dev managers)

# Madlibs

- An old technology still being used today
- A cool new tech
- A thing annoying C.S. students quote their professors about. “My professor always said you should never …”
- Another cool new tech
- Bogus title that tries too hard to be cool, e.g. “Chief Officer of Hanging Out”
- A short word that’s a noise or about eating, e.g. “puff”, “chomp”
- A subgroup of millennials
- A hobby or collection

# Survey

- How many people will graduate soon (even next year) with some kind of tech degree?
- How many people have a tech degree already?
- How many people trying to switch from a non-tech career to a tech career?
- How many people are actively searching for a job right now?

# Job Hunting

## Job Hunting Tips

- The first thing a recruiters does when they hear of an applicant is check **LinkedIn**. Doesn’t matter if you hate it, you have to be on it when job hunting.
- Clean up some of your assignments on **github** and make them publicly available.
- Recruiters and hiring managers are Gen X’ers and baby boomers not millenials. Check your **emails and voicemails** diligently.
- **Networking** is always better than blind resume submissions. Attend meetups, hang out at college placement office, go to open houses.

# Should I Work For This Company?

## Is This Company Legit?

Watch out for:

- **Ghosting** – If they don’t respond, they're not serious about hiring you
- **Obvious Copycats** – “the next Facebook”, “the next google”
- **Immature Teams** “Source control? Sprints? QA? Nah we don’t do that”
- **Team Size** – healthy development teams are around 6 to 10
- **DEV:QA ratio** - 10 devs and 0 or 1 QA means “we don’t care about quality, just rush out buggy code”
- **Actual Dev Team** – if you’re the only programmer on some random non-technical team, you won’t get support for your career

## Avoid Stale or Lame Technology

- Go back to the 2000’s!
  - **Jquery-** great for building dynamic websites but makes messy code
  - **JSP** - java server pages
  - **Microsoft Access** –boss is an accountant who doesn’t understand technology
  - **Microsoft ASP** (from the 90’s), **ASPX** (early 2000’s)
- **Sharepoint** - all devs hate it.
- **Razor/Blazor** – don’t do weird macros in ASPX pages, use Angular or React
- **AngularJS v1 or v1.5**. Not so bad but out of date, but never versions much better
- **Subversion or Microsoft TFS**: source control war is over, git won
- **Wix, Squarespace, Wordpress** – good for designers and bloggers, not programmers
- **SQL Server Reporting Services, Business Objects, Crystal Reports** – should be BI instead
- Avoid these **dead technologies**: **Cold Fusion, Visual Basic, Classic PHP, COBOL**

## Should I Work For A Big Company?

- **Pro:** Career Security (kinda)
- **Con:** Slower career growth
- **Con:** Subject to whims of Wall Street
- **Con:** Boring projects
- **Pro:** Slow but steady promotions
- **Con:** Existential angst
- **Pro:** Boss Probably a Parent
- **Con**: Older Technology – always 2 versions behind

## Should I Work For A Small Company?

- **Pro:** Cutting edge
- **Pro:** Meaningfuller
- **Con:** Workload
- **Pro:** Learn fast and deep
- **Pro:** See how the sausage is made
- **Con:** Career insecurity (kinda) *"Hey gang, meeting in the conf room in 15 min, bring your badges and laptops, no reason"*

## Should I Work Remote?

Remote teams exist and can be highly effective— but there are lots of that whither and die

- **Pro:** Obvious-- you get to work from home
- **Con:** Out of the loop, can’t hear rumors or gauge temperature
- **Con:** Waste time on a problem instead of asking for help
- **Con:** Harder turn off work mode and relax, requires discipline

# Offers

## Did I Get An Offer?

Did you really get an offer, or did you just think you did?

- “Hey sounds great I think it’s a really good fit and we are looking forward to working with you”
- "We're ramping up our hiring and we're very excited about our new client XYZ“
- “We’ve got a very exciting year ahead of us”
- Every hiring manager has to balance risk around when new projects start and when they should hire-- that’s their job. **Don’t let them offload their risk on you and when you’re not getting paid for it.**

## Job Offer Checklist

- Explicit **start date**
- **Title** (but don’t worry too much the wording)
- **Salary**
- **Vacation** days (2 to 3 weeks is normal)
- **Location** (maybe different from interviewed location?)
- Expectations on **travel**
- Expectations on **overtime** and after hours support
- **Benefits** start date
- **Retirement** plan start date
- Some kind of **signature** or company logo

## Negotiate The Offer

- Sorry, no clever advice about salary negotiation.
- **Gotta-haves** – if failure to achieve this accommodation will result in you declining the offer be as clear and specific as possible immediately up front
- **Nice-to-haves** – if it’s not a deal breaker, and you suspect most dev bosses will be cool with it, don’t pester the nice HR person or recruiter, talk to the actual dev boss
- **HR policy vs. reality** – the dev boss might say “sure you can leave a little early on Fridays no big deal” but the recruiter or HR person will quote “All employees are expected to work 8 hours a day not including 1 hour lunch….”

## Negotiate Tuition

- Will I get tuition reimbursement?

## Negotiate Tuition

- Will I get tuition reimbursement?

- No

## What If They Reneg On An Offer?

- A cover letter is a legal contract, and if they breach I should sue them right?

## What If They Reneg On An Offer?

- A cover letter is a legal contract, and if they breach, I should sue them right?

- No

# Quiz

# Quiz 1.1 Q

- **QUESTION:** A recruiter asks if you're looking to be a W2 or 1099. What does that mean?

# Quiz 1.1 A

- **QUESTION:** A recruiter asks if you're looking to be a W2 or 1099. What does that mean?

- **ANSWER:** W2 means full time employee (taxes withheld, probably benefits, usually 40 hrs/wk), 1099 means independent contractor (withhold your own taxes, contract stipulates hours or fixed cost, you're not an employee)

# Quiz 1.2 Q

- **QUESTION:** What are the three main frontend Javsacript toolkits?

# Quiz 1.2 A

- **QUESTION:** What are the three main frontend Javsacript toolkits?

- **ANSWER:**
  - Angular
  - React
  - VueJS

# Quiz 1.3 Q

- **QUESTION:** What source control is the most popular now?

# Quiz 1.3 A

- **QUESTION:** What source control is the most popular now?

- **ANSWER:** Git particularly on Github

# Quiz 1.4 Q

- **QUESTION:** What does REST API mean?

# Quiz 1.4 A

- **QUESTION:** What does REST API mean?

- **ANSWER:** A technique of building APIs so that frontend code (like Angular) to talk to backend code (like NodeJS) using HTTP commands like GET, POST, DELETE and transfer data using JSON.

# Quiz 1.5 Q

- **QUESTION:** What is Docker?

# Quiz 1.5 A

- **QUESTION:** What is Docker?

- **ANSWER:** Docker containers are a small copy of an operating system with some tools pre-baked into them to which you add your custom code, then it can be run anywhere.

# Start Career

## The First Day Skit

## First Day What To Expect

- Bring **driver’s license and passport or social security card**
- Have a **checking account** ready and figure out your **routing number** and **checking account number**
- Expect the first two or three days to be a **complete waste of time** as you fill out forms, get a laptop, badge, network account, and email
- Then you’ll **sit around awkwardly** for a few more days when you boss realizes they never planned anything specific for you to start on
- Get your **boss’s cell phone** number for when you’re stuck in traffic or forget your badge

## How to Survive Chaos

- For some reason, established bosses assume that new hires can’t talk to people outside their team. Talk to the end users directly and find out what they need. Follow up with written emails/wiki pages explaining proposed solution.
- Problems that you can't solve get explained to your boss and are not your responsibility anymore.

## How to Survive Chaos – Email Summary

Your company is a complete mess and your can’t tell what your boss is thinking?
Send weekly email summaries. Keep it short an on point.

Accomplished last week:

- Fixed bugs with login screen (16 hours)
- Had meetings to go over web redesign (4 hours)
- Updated database index to improve performance (15 hours)

Planned for this week:

- Finish test scripts

Questions/Comments/Concerns:

- Need to get license for server
- Marketing team hasn’t responded yet to my email questions from 4/15/19

## But I’ve Got Baggage!

EVERYBODY'S got baggage including your boss and boss’s boss. Medical issues, childcare issues, family stuff, marriage/divorce, home repairs, etc.

- Focus on what accommodations you need from your boss
- Don’t dwell on personal details
- For small stuff, an informal question to your boss might get a better answer than a formal question to HR

# Midcareering

## Midcareering Tips

- 10,000 hours (3-5 years) of blood, sweat and tears and late nights to master your craft
- Build relationships with people **outside your immediate circle**
- Build your brand and become a **trusted professional** in your area
- If you only clock your hours in your cube with your headphones in and avoid people, there won't be any advancement.
- What else can you do outside your 9-to-5? Attend meetups? Write blog articles?
- No one else will tell you if your current job is a dead-end with no hope of advancement. Burden is on you to actively advance your career through **learning, networking, and taking risks**.

## Red Flags At Your Company

- Sharp uptick in **closed door meetings**
- Bosses **can’t make simple decisions**
- New set of employment papers for you to sign
- Instructions to document **your own job description**
- You hear people say “**due diligence**”
- Sudden urgency around making sure various client documents and agreements are signed and up to date
- Delays in payroll or expense reimbursement "**it's the bank's fault**".
- Unexpected company meetings where CEO says “**Everything’s Great**!”

## Merges and Acquisitions (M&A)

Okay so something's happening, but it's not immediate bankruptcy or dissolution. How can this benefit you?

- **Churn** can be bad or good or **GREAT** for your career. You get a first impression do-over!
- You’re gonna lose some fun stuff
- Observe the veteran’s body language
- **Avoid tribalism**
- Find an excuse to chat or have lunch with all the senior devs and middle managers you can

## Come In My Office And Shut The Door

So things aren’t going well, but you’re not fired.

- If a boss gives you a Performance Improvement Plan or puts you on probation or wants to work with you on improvements – **SOMEONE IS THROWING YOU A LIFELINE, TAKE IT!**
- A bad boss will ignore things too long then explode, a good boss will give you honest feedback and specific instructions on what should be improved.
- Swallow your pride and take it seriously
- The role of HR is to represent the interests of the company

## I Just Got Fired– What Should I Do?

- **Calm down**, it happens a lot
- You're not going to talk your way out of it
- It's **not a career killer**
- Future employers are probably too busy to care
- Have a **bland stock answer** ready. "It was time for me to pursue other opportunities" or "the position wasn't a good fit for me"
- **LinkedIn** updates, network, brush up on latest tech

## Will I Get A Severance Package?

## Will I Get A Severance Package?

No

# Long Term Career

## The Millionaire Skit

## Career Options – Tech

- **Help Desk** - everybody's got to start somewhere and it's a good place to learn skills, but it's tough to get promoted out
- **Senior Developer** - very reasonable to accomplish, but gets scary when you eventually hit 50
- **DevOps** -- If you like networks and servers and shell scripts and can program but think most programmers are idiots this might be a good fit. Build Engineer, CI, CD, Build automation.
- **Enterprise Software Specialist** - learn Salesforce, SAP, Peoplesoft inside and out. A lot of material to learn but relatively good job security. Gets boring quick.
- **Tech Guru** - spend a lot of time marketing your brand and attending events and advertising yourself.
- **DBA --** nobody thanks you when the databases DON’T crash.
- **AI/AR/ML --** every year for the last 20 years the tech media says “This is the year of AI!” It’s finally happening in niche areas, but it’s tough to get a job in here because most corporate efforts aren’t well thought out
- **Mobile –** very difficult to keep up to date on all the latest mobile kits and risks of vendor lock-in (Swift)

## Career Options – Architect

- **Software Architect** - often times this just means 10+ years experience and you're more expensive than a senior developer. You do boss-like work without boss salary.
- **Solutions Architect** – put together larger systems that integrate multiple technologies for a specific business purpose
- **Enterprise Architect** – have to master every technology out there, not just the fun cool JavaScript frameworks but all the buzzwords that only used in Fortune 100 systems like *Enterprise Service Bus* and *Tenant Security Framework*
- **Cloud architect** - specialize in building large scalable solutions in AWS, Azure, or the various other providers. Tough to keep up, good job security.
- **Network architect** - not really my area. Every large company has a complicated network that needs lots of attention and expensive hardware and monitoring.

## Career Options - Security

Security is a complete mess that no one understands it. So many encrypted tokens and certificates flying around and getting redirected and MFA. It's amazing that the entire internet hasn't collapsed under the weight of all the security protocols.

- **Security specialist** - probably much less exciting than it sounds. Most of the time you’re telling people what they’re doing wrong, not building fun stuff.
- **Pentest/White Hat** - everybody's heard stories of the dropout who's making millions getting paid to hack into big company networks, but the body of knowledge is huge.
- **Auditor** - there are various flavors of technical and security auditors but they are all borne out of accounting traditions first - so it's mostly checklists.

## Career Options - Other

- **Social Media** - the labor pool is very saturated, not much money left.
- **Recruiting** - works best for those who are big talkers and have a very large social/professional network. You pretty much already know if you'd make a good recruiter or not. It's like the realtors of tech

## Career Options - Industry

- **Adopt an Industry** as opposed to a technology, e.g. real estate, legal, or non-profit, and be a specialist
  - You’re usually tied to a big vendor but you can easily job-hob to clients or other vendors.
  - You know everyone in this space
  - You become an expert in that industry
- **Higher Education/Government IT** -
  - Things move SLOOOOOOOOOOOOW
  - Budgets are smaller
  - Personal politics are more intense
  - Tech is older
  - Working hours are better, people are slightly nicer, slightly more job security

## Career Options - Creative

- **UI/UX –** mix of psychology, design, art, Photoshop, Illustrator, Blender, HTML, CSS, business analysis
- **Gaming –** labor market is saturated with real and wannabee game developers. Gaming companies have very tight profit margins and high demands on employees.

## Career Options - Indy

- **Independent Contractor**
  - Lots of risk
  - Some companies won’t pay for 90 days, do you have 3 months wages saved up?
  - If you’re full time at a client, you’re not selling and landing your next gig
  - No payroll withholdings, you pay your taxes quarterly
  - Miss a week or two? Massive hit to your income.
- **Tech Entrepreneur** - all your time will be spent networking and shmoozing investors and spouting buzzwords and no time for hands on tech.

## Career Options - Consultant

Nobody's a management consultant at age 22. Nobody's an tech expert yet either at that age. Most people migrate to consulting after 5-10 years in the industry

- **PRO:** you get exposed to **new technologies**
- **CON:** you have to **work harder to keep up to date**.
- **PRO:** you have **more control** over your destiny.
- **CON:** your **life revolves around the billable hour** and getting the next client lined up
- **PRO:** it's **entrepreneurial** but you get salary and benefits and some stability
- **PRO:** When you get paid for your opinion suddenly **people start listening more**
- **CON:** Sometimes you show up and realize **the client knows more** than you do
- **GIFT/CURSE**: Burden and responsibility of **estimating project's budget and schedule** falls more directly on you.

## Career Options - Project

- **Project Manager** - realistic career transition for tech people. Sometimes you can carve out “technical project manager” as a stepping stone if you can’t get promoted to “dev manager”.

  A programmer can’t weasel their way into being an accountant or stock broker, but you can a PM.
- **Product Manager** – boss off several project managers, plus you have to bring charisma and leadership to deal with the board room and dev team
- **Project Management Office** (PMO) – can only wield soft power

## Career Options - Sales

- **Sales** - where programmer's souls go to die
- **Sales Engineer, Solutions Engineer** – hands on technology, get to work with clients and prospects, set up servers and install stuff and solve problems. Always second banana.
- **Evangelist** - hired gun to promote a specific technology and pretend you love it on social media. Everybody hates them.

## Career Options - Management

- **Management**- good outcome for devs with a few gray hairs. Not much authority, dealing with difficult personalities, lose touch with tech because you don’t do it every day. The good ones shield their people from exec and sales nonsense and let devs be semi-independent.
- **Dir/Sr VP** - second/third level manager. Mostly managing budgets and trying to figure out how to explain to execs why projects are late. The good ones make a positive tech culture.
- **Exec/CIO/CTO** - very few of these people are super technical, and those that are-- are mostly network engineers. Office politics, reviewing budgets, dealing with vendors, soothing the CEO, soothing the CFO, soothing the COO.

## Career Options - `else`

- You’ll be stuck in the same cube on legacy tech **FOR FOUR DECADES** or until the company goes bankrupt or you die

#Tech Landscape

## Languages

Not all IT jobs are programming and but it certainly helps to know them.

- **C#** – Used for mid sized and large sized backend corporate apps and Windows apps. MVC, Entity Framework, DotNetCore are important tools.
- **Java** – huge backend enterprise apps and Android development. Spring is the most popular toolkit.
- **Python** – Gaining in popularity. Heavily used in stats. Weird rules around whitespace. Django used for websites
- **NodeJS** – Recent entry. Not strongly typed. Lots of toolkits required to get anything done. 10 ways to do every task and each is a different paradigm. Growing in popularity
- **Ruby on Rails** – ROR devs can’t shut up about how great it is. Can prototype very quickly and has extensive toolkit.
- **PHP** – Older tech that's still around. Harder to use, language isn't as stable and secure, sometimes really hard to figure out what's going on. Many toolkits on top to make it easier.
- **C/C++** – closer to the hardware and for highest performance. Subtleties and nuances are fatal if not properly understood.
- Everything else? Fringe or niche. Honorable mention to **Go**. Yes Swift is niche.

## Frontend

Popular front end toolkits make sense of HTML, CSS, and API calls.

- **Angular -** javascript framework for building web and mobile frontends. Corporate america embraced Angular in the past few years. "TypeScript" is strongly typed JavaScript, but it's a really thin layer and not a big deal.
- **React -** similar toolkit supported by FaceBook. Seems to be more popular for mobile solutions. Does not use Typescript (sort of).
- **VueJS -** the most forgiving. It’s easier to roll this out incrementally.

Angular and React are tied for first. All others are fringe (knockout, ember). Deploying any of these to an actual website is very complex (fonts, javascript versions, toolkits, mobile, etc.)

## Business Intelligence (BI)

- We used to call this "reporting". People build entire careers out of visualization. Now it’s a combo of statistics, UI/UX, data warehouse, SQL, web development, and some MBA style buzzwords like KPI
- Vendors:
  - **Tableau** (most popular in corporate American IMHO)
    Consider looking at community.tableau.com
  - **Qlik**
  - **PowerBI**
  - Many, many, more. Microstrategies, Birst, …
  - If you can handle Excel pivot tables and charts, you’re fine.
  - If you know one can quickly learn another.

## File Formats

- XML, JSON and YAML are all file formats used for multiple purposes, and they're all basically the same. Nothing hard to learn, don’t be scared of these terms.

## Stacks

- **LAMP** - Linux Apache MySQL PHP
- **WAMP** - Windows Apache MySQL PHP
- **MEAN** - Mongo Express Angular NodeJS
- **MERN** - Mongo Express React NodeJS
- **Windows stack** Windows, IIS, SQL, ASP.NET
- Python, Django, MySQL
- **Ruby on Rails**
- **PHP frameworks**: Laravel, CodeIgniter, Symfony, CakePHP, Zend, Yii, Phalcon, FuelPHP, PHPPixie, Slim. Most devs who do full time PHP hate it

## Stats

- “**Data scientist**” doesn’t have too precise a meaning and covers several disciplines.
- **Rust** programming language for machine learning and data analysis, more stable, statically typed
- **R** scientific computing and statistics
- **Julia** – scientific computing, REPL
- **Elixir**
- **Python** - seems to be the most popular programming language for general statistical analysis
    - numpy numerical calcs
    - pandas tabular data sets

## Relational Database

- **Microsoft SQL Server** - preferred in mid/large corp clients, some shops are "Microsoft shops", good job security knowing MSSQL. Easist to install and administer, a dev can fake dba roles pretty convincingly for small projects.
- **Oracle** - been the top workhorse for decades. Preferred for large and very large legacy enterprise software. Nobody gets excited about Oracle anymore. Syntax and tools feel archaic. Still heavily used.
- **MySQL** - used in a lot of open source projects. Getting to be a bit outdate, mySql projects feel like early 2000's projects. Fun new exciting projects don’t use MySQL.
- **Postgress** - open source, more powerful than MySql, people are really excited about it but Corp America is still nonplussed but that's slowly changing

## NOSQL Database

Instead of normalized database tables, huge collections of disconnected JSON or XML blocks, and the program access the database has to make sense of it. Scales much better than traditional SQL. Works well with distributed systems.

- **MongoDB** seems to be the favorite. Tight integration with NodeJS. Define indexes not map-reduce functions.
- **DynamoDB** hosted by Amazon. Great for misc key value pairs
- **FireBase** – free cloud hosted database great for prototyping
- **Cassandra** – stores HUGE databases, more SQL-ish
- **CouchDB** – best for accumulating, occasionally changing data

## Cloud

- **Amazon Web Services** (AWS) - the leader in cloud computing. EC2 (Elastic Cloud Computing) most popular. You can't avoid AWS, at some point in your career you'll be involved with it.
- **Microsoft Azure** - strong second place contender. Microsoft focused but not exclusively, lots of linux offerings and other tech as well.
- **Google** - offers lots of services, but most companies use AWS or Azure for cloud computing. They do a lot of big things that aren't as obvious
- **Docker** – mini virtual OS’s with tools installed, you add your stuff on top. important but kinda easy to learn. It only takes a few tutorials and free AWS things to say "Yeah I've done docker“
- **Cloud Foundry** – Hides operating system, you get language specific build packs to add your stuff to.

# Terminology

## HR Terminology

- **W-2** – The tax form full time employees get
- **1099** – The tax form Independent contractors get
- **Exempt** – fixed hours per week, no overtime
- **Non-exempt** – paid hourly without fixed hours
- **I9** – sign this form when you get hired to confirm you can work in U.S.
- **W-2 hourly** – short term employment, only paid for hours worked
- **Corp-to-Corp** – you have an LLC (company) that gets paid
- **H1B** – foreign worker visa
- **Biweekly** – paid every two weeks
- **Semimonthly** – paid twice a month

## Contract Terminology

There are legal document terminology employers like to toss around

- **RFP** - big organization requests bids from vendors to supply some service.
- **LOI** - letter of intent- a company *may* do something in the future
- **NDA** - non-disclosure agreement.
- **SOW** - statement of work, outlines who's doing what and how much they'll get paid.

# Conclusions

## Nobody Cares About…

- How clever or **aesthetically pleasing your code** is
- **Your opinion** about why this programming language or platform is better than the others
- How your **old professor thinks** things should be done
- How the **root cause of your bug** is the operating system or toolkit. Nope it's **your responsibility** to come up with a solution

- *I do care about evaluation forms, please fill them out to help improve for next time*

## Other Important Stuff

- It's okay to be in a "**warm recruiting pipeline**“. Some day you’ll get the call and be ready.
- There is **no social class structure.** You can say “hi” to execs.
- Most problems roll downhill to you, that's life. Management's inability to make a decision or their unrealistic optimism in setting delivery dates isn't something you can control. However, you can and should **clearly communicate what you're realistically going to be able to finish in the short term** and what things you need specific help with.
- If you have six month’s of savings, you can enter every work dilemma/crisis/fiasco from a stronger bargaining position, otherwise you just have to put up with it.

## Be Friendly To These People

- **Senior dev people** on your team who aren't your boss – you will learn more from these people than you learned in college
- **Security guard/receptionist**
- **Administrative assistant** in your department – they can answer almost all of your first day questions
- **Executive assistant** to your team’s exec – some day you’re really going to need something signed or approved and they will slide paper across the desk
- **IT staff** that helps with laptop/network issues – when you drop your laptop a third time you really want them to be nice to you

## Misc Useful Stuff

- **Confirm your PTO early as possible**. Book hour Dec PTO right after thanksgiving. Very easy to change a day here or there as plans change, but bosses don't like to be told Dec 19th "oh yeah I forgot to tell you I'm out of town and not back till Jan 3rd"
- It takes years of experience (decades for me) to accurately **estimate schedule and budget** projects, and still is really tough. Agile helps a lot but is not always an option.
- Who's opinion does your boss value? Certainly not you-- the new hire. Carefully **watch how people with good reputations got their good reputations**. It's not a secret - hard work, deliver on what you promise, take extra effort to do extra research on a problem, be willing to be honest even when it’s unpleasant, don't be full of hot air. They're not necessarily smarter than you.

## More Misc Stuff

- Vacation is doled out in “hours accrued per pay period”. If you have two weeks vacation and semi-monthly payroll:

$$10\ PTO\ days \times \frac{8\ hours}{day} \div \frac{24\ paychecks}{year} = 3.33\ PTO\ hours\ accrued\ per\ pay\ period$$

- During the recruiting process, whatever the company says about agile or scrum or Kanban you should just nod politely and say "Yes". Whether you love or hate agile, no one is going to listen to your opinions as a new hire.