CSE 403
Guest lecture
Tips for thriving as a software engineer

Nathan Sjoquist

About me

Software engineer at Microsoft

CoreAI Applied Engineering

PhD from University of Cambridge

MS and BS from University of Washington

Goals of this lecture

- Share tips to succeed as software engineer.
- Share helpful experiences about working on a software team in industry.

Agenda

- Undergrad at UW
- Working at a startup (first job)
- Working at a non-profit (Allen Institute)
- Masters at UW
- PhD at Cambridge
- Working at a for-profit company (Microsoft)
- Job Searching Tips
- Interviewing Tips
- Questions

Undergrad at UW

Took CSE 403 with Gail Alverson in 2008.

Undergrad at UW

- Graduated in 2009 into the global financial crisis.
- Had lots of anxiety about interviewing and getting my first job.
- Didn’t know what to expect.
- Didn’t know what was required to work as a software engineer in industry.

Undergrad at UW

- Focus is on broad skills (critical thinking, problem solving, writing, logic reasoning)
- VLPA, English composition, foreign language, natural science, etc…

Computer Science department focuses on broad range of skills and subjects for undergraduate students

- Computer programming, foundations of computing, data structures, algorithms, theory, computer architecture, compilers, security, UI design, networking, operating systems, computer graphics, computer vision, AI, etc..
- Most of undergrad is made of introductory classes.

Undergrad at UW

What do you need to know for your first software job to best prepare for success?

Know the material from your introduction to programming and data structures classes very well.

- Most software job interview questions will come from this material.
- Most of the skills needed to be successful in a general software engineering job come from this as well.

Undergrad at UW

Do you have to be an expert at everything to excel as a software engineer after you graduate? No.

Find out what area you enjoy (and excel in) within computer science and focus on that.

Undergrad at UW

Is it difficult to quickly learn a new language needed for a software job?

No. It is quite easy. The core programing constructs and data structures are the same, the syntax and libraries are different.

Undergrad at UW

How to get your first real software job?

- Try to get an internship – easiest road to first software job.
- Use your connections (friends, etc…).
- Take whatever job is available (at first) to gain experience and make connections.
- Later, move to a different job if needed (very common).
- Work on side projects that are interesting to you.

STARTUP

Working at a startup

Worked at a startup trying to solve a natural language processing (NLP) problem:

How to create automated English subtitles for video conference meetings.

- Full stack software engineer.
- Organized and stored training data.
- Made data searchable and viewable through simple web UI.
- Designed a SQL DB.
- Used Java and Perl, SQL.

Working at a startup

- Very little thought was put into the engineering process.
- Writing code as fast as possible was the priority.
- Lots of technical debt.
- Long hours.
- Exciting, fun and interesting work.
- Interesting people.
- Very little testing.
- Entire company had five employees.
- Startup received some funding but was not ultimately successful.
- Learned lot about how to write software.

Working at a startup

What is the biggest difference between CSE undergrad and working as a software engineer in industry?

- Undergrad is primarily a solo effort.
- Software engineers work as a team.

Working at a startup

There are constant discussions between team members in all aspect of work – especially when difficult problems arise.

Your team is a huge resource.

Working at a startup

What do you if you encounter a new and difficult problem that you don’t know how to solve?

- Research the problem (search stack overflow, copilot in VS code, read documentation, etc..).
- Sketch out a design.
- Use a debugger.
- Try multiple approaches and don’t give up.
- Ask a co-worker, ask your team.

Working at a startup

What skills or knowledge did you wish you had developed more in college before starting your career?

- More programming experience.
- Design patterns, how to apply them.
  - Every project is different, but you will start to see the same types of problems repeatedly.
- More experience in how to design software architecture.

ALLEN
INSTITUTE
open for science

Working at a non-profit
Allen Institute for Brain Science

Full stack software engineer

- https://alleninstitute.org/
- Worked on open-ended research problems.
- Helped scientists implement algorithms.
- Worked on a small team with about 10 software engineers.
- Much less intense than the previous startup job.
- Very interesting work.
- Worked fewer hours (normal 40 hour week).
- Lots of unique benefits.

Working at a non-profit
Allen Institute for Brain Science

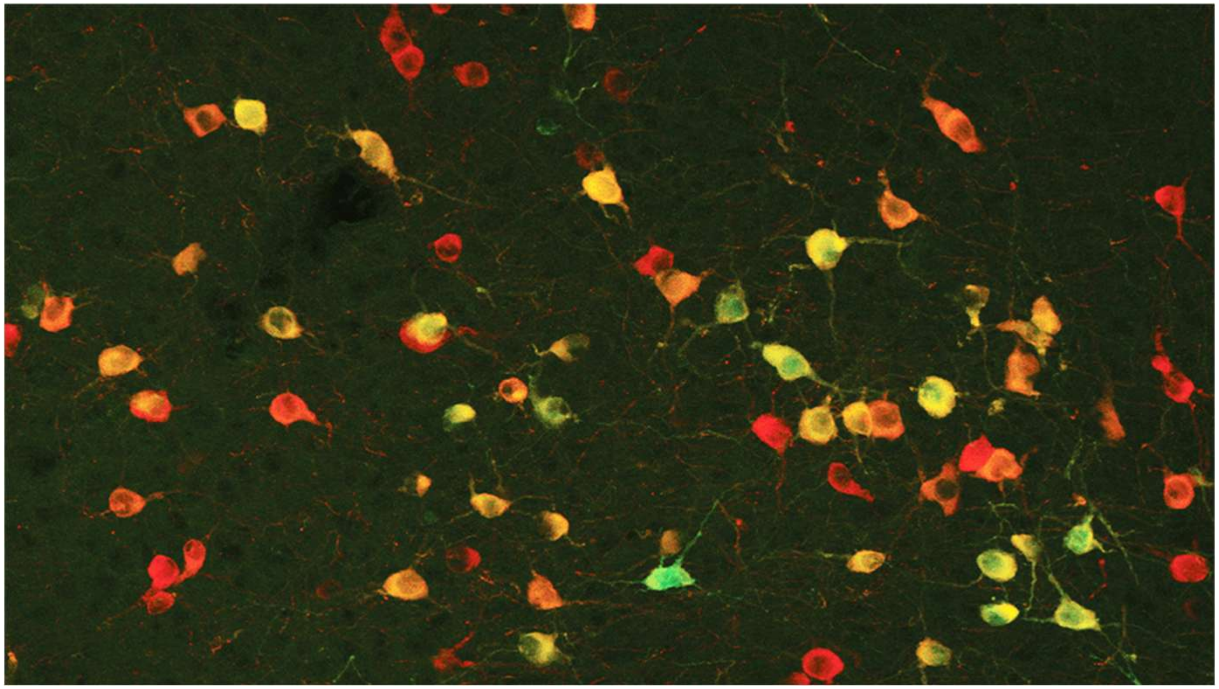
What did I work on?

- Built image viewers and custom video players.
- Built systems to store and manage huge amounts of lab data.
- Implemented image processing and computational neuroscience algorithms.
- Made digital atlases of the Brain.
- Wrote search (indexing of data) for website.
- Database design and ETL work.
- Some simple (internal) UI for internal systems.
- Many internal tools.
- Lots of segmentation algorithms (AI work)

Working at a non-profit
Allen Institute for Brain Science

Full stack software engineer

- Ruby on rails
- Python
- Postgres SQL
- C#
- JavaScript
- AWS
- Github (source control)



neurons labeled with fluorescence in a mouse brain.

Working at a nonprofit -
Allen Institute for Brain Science

How does the Allen Institute balance the need for rapid feature development while maintaining code quality? Are there specific strategies or processes that your team followed?

- Lots of testing.
- Full heavyweight scrum.
- Dedicated tester roles.
- Still had lots of technical debt (from old software).

Working at a nonprofit - Allen Institute for Brain Science

What are your favorite memories as a software engineer at the Allen Institute?

- Making good friends.
- Finishing and releasing difficult projects with excellent solutions.
- Figuring out what type of software was most interested to me (AI).

Working at a nonprofit - Allen Institute for Brain Science

What attracted you to AI?

I thought the field contained the most interesting problems.

Working at a nonprofit - Allen Institute for Brain Science

How do you decide when to specialize in a specific domain (e.g., mobile, backend, infra) versus remaining a generalist?

1. Focus on what you enjoy doing and what you are passionate about.
2. Focus on what you are talented at.
3. Accept the opportunities that are open to you.

Masters at UW

Professional masters program (PMP)

- Part time night classes.
- Provided more in-depth knowledge of CSE subjects.
- Could choose exactly what classes to take.
- Classes were slightly more difficult that 400 level CSE classes.
- Overall, easier than undergrad.
- Focused on AI themed classes (computer vision, machine learning, deep learning, NLP, general AI, etc…)
- Did a research project in computer vision with Linda Shapiro (https://homes.cs.washington.edu/~shapiro/)

Cambridge University

PhD at the University of Cambridge

Why a PhD?

- To become an expert in a field of computer science.
- Necessary to gain the skills and credentials needed to enter academia.

PhD at the University of Cambridge

What research is like?

- Lots of reading of research papers.
- Lots of writing.
- Lots of programming.
- Lots of presentations within the university and at conferences.

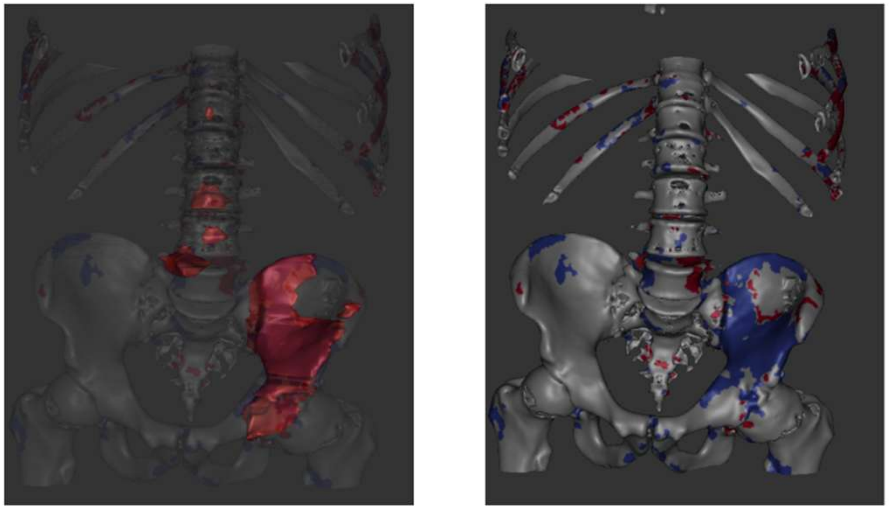
PhD at the University of Cambridge

How to get your degree?

- Pass general exam (breadth of knowledge).
- Write a thesis containing original research (depth of knowledge).
- Defend thesis at oral defense (viva).

PhD at the University of Cambridge

- Project was in AI (computer vision)
- Designed a novel algorithm to detect metastatic bone disease (spread of cancer into the bones) within CT images.



(a) Using an alpha blend to see the ground truth 3D volumes (shown in red) which are located within the bony surface.

(b) Possible sclerotic lesions are shown in blue while possible lytic lesions are shown in red – found using the assessment method.

Microsoft

T

Working for a for-profit company (Microsoft)

Hired after my PhD onto the AI platform team at Microsoft

- Software engineer on CoreAI applied engineering team.
- Confidential AI projects.
- Startup feel.
- Very interesting work.
- Microsoft is an excellent company to work for.
- 100% remote (all meetings are on Microsoft teams).
- Flexible work hours.
- Mix of research and engineering.

Working for a for-profit company (Microsoft)

What you do in your job?

- Design work (create requirements, design APIs, class entities, interfaces, flow charts, etc…).
- Write lots of code (python, C#).
- Attend meetings.
- Learn new things all the time (read academic papers, read documentation, search stack overflow, query VS Code's copilot).
- Lots of internal demos.

Microsoft – typical day

20% design work

60% programing

10% meetings

10% learning

Microsoft – scrum

Daily standup.

Very lightweight project management framework (not too many meetings for software engineers).

Startup feel.

Lots of prototyping and building things quickly.

Microsoft

- No dedicated testers
- Software engineers write their own tests (unit, functional, system, etc…)
- Use continuous integration as much as possible.
- Use Git (Microsoft DevOps) for source control.
- Almost everything (all jobs) is run within a docker container and pushed to a VM cluster on azure.

Microsoft – working on a team

- Have four other software engineers on my team at various levels.
- Report to one principal program manager (PM).
- Work closely with other engineering teams.
- Work with data scientists.

Microsoft

How does Microsoft balance the need for rapid feature development while maintaining code quality? Are there specific strategies or processes your team follows?

- Pull requests (PRs) are reviewed for each feature.
- We write tests (unit, system, etc…).
- My team does a lot of prototyping of new ideas so there is less focus on maintaining code quality.

Microsoft

We use Azure for almost everything:

- Azure machine learning (to run our core jobs on clusters)
- Azure storage (to store files and data)
- Azure functions (to provide a web API)
- Azure key vault (to store secrets)
- Azure managed identities (to securely connect resources)
- Azure application insights (for logging)
- Azure SQL server (for database)

Microsoft

How do you use AI (in general)?

- Use copilot in VS code all the time.

Microsoft

What IDEs and text editors do you use?

- VS code (Microsoft integration)
- Sublime (very fast, lightweight, customizable)
- VIM (when using a Linux terminal)

Job Search

Job Searching tips

Lots of types of places that a software engineer can work at

- Software companies
- Hardware companies
- Not-for-profits
- For-profits
- Start ups
- Academic institutions
- Other companies in general
- Etc…

Job Searching tips

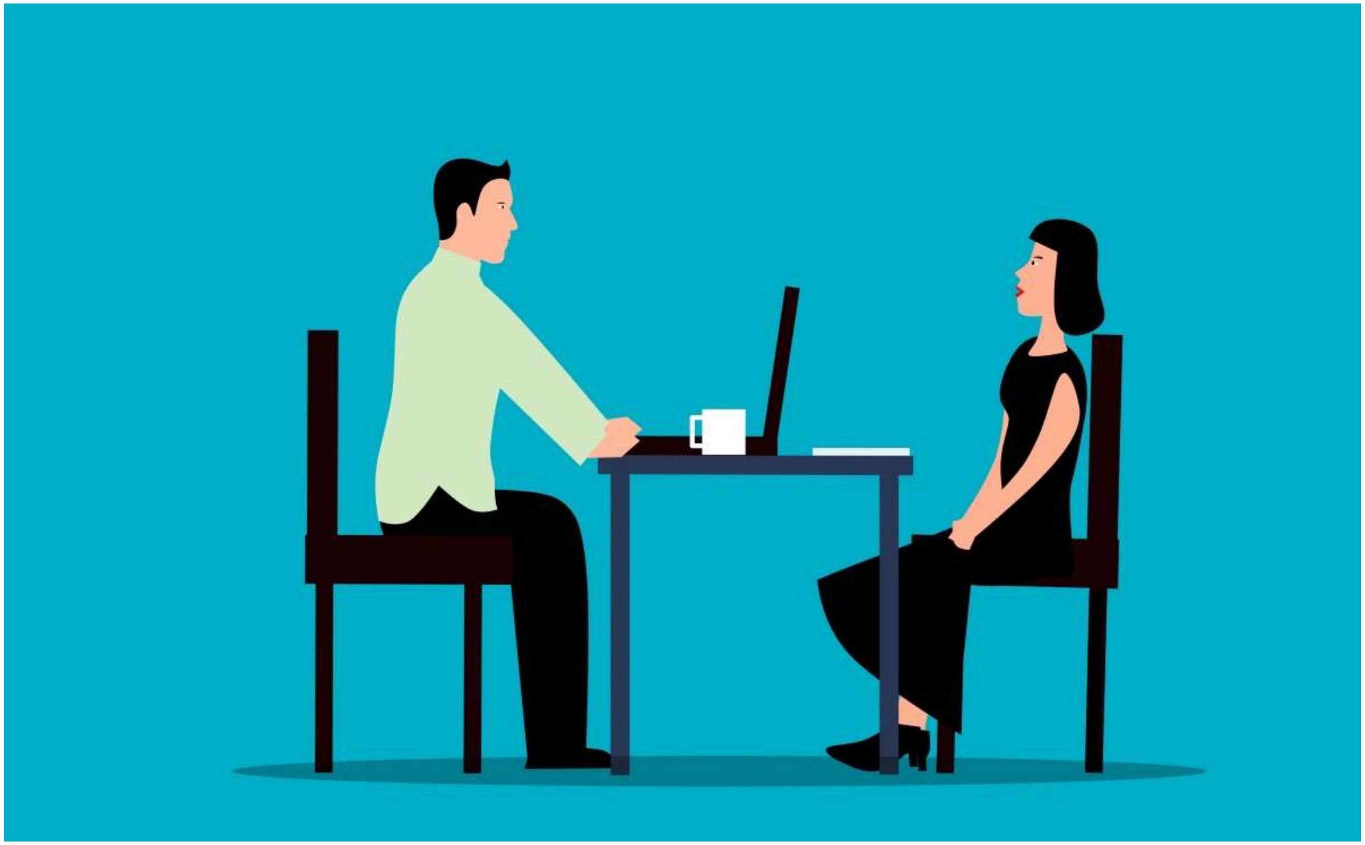
Different role types:

- IC – individual contributor (non-manager)
- PM – program manager (manages a team)

Job Searching tips

- Take what jobs are available to you at first.
- Keep trying.
- Keep learning.
- Make and use connections.
- Use LinkedIn (https://linkedin.com/) to manage connections.
- Know what your ethics are.

Interviewing



Interviewing tips

For a software engineer there are generally four types of interviews:

Recruiter Call
Behavioral
Coding
System design

Sometimes these are mixed together, and sometimes they are separated.

Sometimes given a take home programming assessment.

Interviewing tips

Recruiter Call –

- A recruiter contacts you and asks if you want to have a chat.
- The recruiter will ask about your background (tell me about yourself question).
- The recruiter will ask general questions about what you are looking for in a job.
- The recruiter will tell you about the company and about the interview process.
- Treat this as an interview – the recruiter can end your interview process right here.

Interviewing tips

Behavioral

Questions are asked to see if you fit in with the values of the companies.

- Study the company’s core values.
- Tell a story from your own experience and tie it back to the company’s core values.
- Search for these types of questions before the interview and practice.

Interviewing tips

Behavioral example

Question: tell me about a time you had to work with a difficult co-worker?

Interviewing tips

System Design

Usually asked a very open-ended question.

i.e. design an online auction site

You will use a shared (online) whiteboard for diagrams.

Interviewers purposefully leave out necessary details – ask lots of questions!

Interviewing tips

System Design

Have a plan that you apply to every question:

1. Narrow the scope and ask clarifying questions.
2. Write out functional requirements.
3. Write out non-functional requirements.
4. Design APIs. (refer to #2)
5. Design entity classes.
6. Diagram the entire system and talk it out.
7. Deep dive into something you are familiar with.

Interviewing tips

Coding Interview

- Usually asked two coding questions in a 50 min session.
- 5 min for intros at start.
- 20 min for each coding question.
- 5 min to ask questions.

- Usually use a shared (online) text editor.
- Usually do not actually run the code.

Interviewing tips - what the interviewer does

Coding Interview –

The coding question and an example input is pasted into the text editor by the interviewer.

i.e. given a list of strings, return the first repeated string

# input =[‘a’, ‘b’, ‘c’, ‘b’]
# output = ‘b’

The interviewer explains the question and output.

Interviewing tips – tips

1. Coding Interview – Read the question and explain the question back to the interviewer to verify you understand what is being asked.
2. Ask clarifying questions – ask about edge cases, inputs, outputs, etc…
3. Talk though the most basic solution you can think of that solve the problem (often an inefficient solution).
4. Talk though the most efficient solution to optimize space and runtime complexity.
5. Ask the interviewer if they are OK with your solution and if you should start coding.
6. Code the solution.

Interviewing tips – continued

```
def find_first_repeated(input: str):
    seen = {}

    for item in input:
        if item in seen:
            return item
        else:
            seen[item] = True

    return False


inputs =[‘a’, ‘b’, ‘c’, ‘b’]

find_first_repeated(input)
```

Interviewing tips – continued

Coding Interview

7. Walk through code with a basic example.

8. Walk through code with an edge case example.

9. Reason out your big-O space and runtime complexity.

10. Talk non-stop through the entire interview so the interview knows what you are thinking.

11. If you get stuck, the interviewer should give you a few hints.

12. Respond to any questions your interviewer should ask you.

Interviewing tips

- Almost all coding questions can be solved from intro programming and data structures classes.
- Study and practice lots of example interview questions (https://leetcode.com/).
- Study data structures and common algorithms (arrays, linked lists, trees, strings, search, sorting, etc…)
- You can use any language you want. Tip - use the language you are most comfortable with - however a high level language (i.e. python), is much easier to quickly code in.
- Study basic syntax of language – doesn’t have to be perfect
- Always create a function as part of your solution.

Interviewing tips

General thoughts:

- Interviewing is a skill.
- Practice as much as you can.
- There are many good resources to help you (https://interviewing.io/learn#interview-guides, https://leetcode.com/, ‘cracking the code interview’ book and system design interview books).
- If you don’t get and offer, you can try usually reapply to the same company in about 6 months.
- Companies generally set difficult interviews to screen out bad candidates and, in the process, screen out many good candidates.
- Keep trying.

Interviewing tips

Once you get an offer:

- Do not accept it without negotiation.
- Companies expect you to ask for more than the initial offer.
- Best way to maximize your offer is to interview at multiple companies and have them try to outbid each other.
- Use https://levels.fyi/ or https://www.glassdoor.com/index.htm to lookup common salaries by company.

Final thoughts

- Do what you are passionate about.
- Take risks – you will find it much more fulfilling.
- Don’t be afraid to fail, just keep trying.
- Money isn’t everything – do what you love.

Questions?