

Tam Vo
CSC 131
Dr. Abeer Abdel Khaleq
19 November 2024

Guest Speaker Session

On 18 November 2024, Ronit Sonawane, a Software Engineer III at Charter Communications with a Master's in Information Technology from the University of Denver, was a guest speaker for our CSC 131 course. Ronit is aiming towards the software architecture route.

Ronit emphasized the importance of LeetCode paired with Communication Skills to obtain a software development job. He was able to be familiarize himself with Data Structures and work with them via Python to excel in LeetCode and the interview process. Creative projects are a great way to emphasize skills during an interview. Majority of the skills will be learned on the job, because the field changes so fast and everyone has different software they maintain, it's necessary to be able to read code and write maintainable code.

Ronit has worked with volatile codebases before, where it would take about 6 months to push a feature. He said it was because many services have a list of dependencies that they need to coordinate and maintain so to appropriately solve the dependency issues, they can only request and wait for updated changes in different departments.

Typically, when you start out as a junior developer, you will start out slow, gaining more responsibilities over time. With the layoffs and AI, software developers are increasingly being pressured to have more output while maintaining a healthy work-life balance.

Technical Debt is a great way to showcase your skills and take the initiative because no one wants to take it on. Senior Engineers focus on more important features than cleaning up the codebase. Ronit was able to refactor code where he was able to make it modularized instead of having to relearn the logic and reduced breaking changes.

1-on-1s are important to career growth and self-development. It is where you and your lead can talk about status and future projections of the company and your career.

Guest Speaker Questions

1. How do you deal with technical debt?
2. How have you dealt with technical debt and what was the impact?
3. How do you combat a project with a volatile code base? (6 months to add feature)
4. How do you balance solving problems on your own vs asking a senior?
5. How do you manage time between 1on1s vs working on the code base?
6. Do you have capacity for a mentee?