**Maxime Alexander Sutters: Inspira Financial Questionnaire Answers**

2001 Spring Road, Suite 700, Oak Brook, IL 60523

**About the Role:**

We are seeking a talented and dedicated software engineer to join our dynamic team. In this role, you will be responsible for designing, developing, and maintaining software applications that align with our business goals. You will collaborate with cross-functional teams to deliver high-quality solutions that solve real-world problems.

**Key Responsibilities:**

- Design, develop, test, and deploy scalable and robust software solutions. - Collaborate with product managers, designers, and other engineers to gather and refine requirements.

- Write clean, efficient, and maintainable code, adhering to best practices and coding standards. - Debug and resolve technical issues, ensuring software performance and reliability. - Participate in code reviews and provide constructive feedback to peers. - Stay updated with the latest industry trends, tools, and technologies to continuously improve development practices.

- Document technical specifications, processes, and solutions for future reference.

**Requirements:**

- Bachelor’s degree in Computer Science, Engineering, or a related field (or equivalent experience).

- Proven experience in software development, with proficiency in one or more programming languages (e.g., Python, JavaScript, Java, C++).

- Strong understanding of software development lifecycle (SDLC) and Agile methodologies. - Experience with front-end and back-end development frameworks (e.g., React, Angular, Node.js, Django, or similar).

- Familiarity with databases (SQL/NoSQL) and cloud platforms (AWS, Azure, or Google Cloud).

- Excellent problem-solving skills and attention to detail.

- Strong communication and teamwork skills.

- Ability to work independently in a remote environment.

**Screening Questions:**

**1. Can you describe your experience with software development and mention the programming languages you're most proficient in?**

I have two years of experience on an infrastructure team at Amazon Web Services (AWS). My team built a system for processing petabytes of datacenter footage a day in real time to ensure the safety and security of datacenter workers and servers across the world. My responsibility was to update, maintain, and expand features related to our machine learning and computer vision infrastructure. My most proficient programming language is Python but I have experience in Java, C, C++, SystemVerilog, and JavaScript from my Computer Engineering degree at the University of Washington. It was there I studied classic CS topics like data structures/algorithms and networking and more low-level hardware-focused classes like FPGA chip development.

**2. Have you worked on both front-end and back-end development? If so, which frameworks have you used?**

The majority of my experience is in back-end development, though my augmented-reality guitar teacher (GuitXR) app involved HTML configuration and JavaScript libraries and frameworks like Three.js, A-Frame, and ML5.js. My team's back-end development stack at AWS was Python and AWS cloud, with infrastructure as code in the form of AWS CDK. I had the opportunity to build systems with Lambda, DynamoDB, S3, API Gateway, and other AWS cloud tools like code pipelines and Distributed Job Scheduler (DJS). And I built APIs with React and GraphQL to communicate data and requests between back-end and front-end teams.

**3. How do you approach debugging and resolving software issues?**

Most time spent debugging is spent understanding the problem thoroughly. Part of that is reproducing the issue, which for my previous team meant poring over logs and dashboards in S3 and CloudWatch, writing unit tests and mocked-out scripts to better understand edge cases in our systems' behavior, and sometimes visiting datacenters in person to conduct more in-depth evaluations of new features. If our current tests didn't catch the software issue it was my responsibility to write new ones that did, and build automated alarms to alert us if the same issue occurred again. My team worked together to make sure the bulk of our efforts were spent anticipating issues rather than responding to them. Once I had resolved the issue on my system and all new and old tests passed I submitted a code review for at least two of my teammates to comment on. I would submit revisions if necessary until the changes were approved, merge these changes to our mainline branch in our repository, and monitor as these changes progressed to wider audiences through our code pipeline. At each stage in our pipeline we had automated tests to make sure even if an unexpected issue occurred it would be limited to a beta environment or at least localized to a small subset of datacenters.

**4. Describe a challenging software project you’ve worked on. What role did you play, and what was the outcome?**

My first large cross-team project was sending mismatched datacenter halls to another team's system to be registered as work assignments for a team to follow up on and correct in our mapping tool. Specifications and requirements were outdated and the API I needed to use to send work assignments to datacenter employees required a major update before it would work as described.

I needed to make the case for cross-team work and secure buy-in from my manager, my contact on our partner team, and his manager. This required thorough investigation of the design's claims while maintaining a cordial working relationship with my contact on the partner team.

I soon uncovered missing and deprecated functionality not mentioned in the initial design document. I built out our AWS service and completed our side of the API calls while asking the other team questions and validating the gaps in the design. I wrote a new design based on the current state of our partner team's API, the mocked out inputs I tested with my team's side of the project, and expected and actual outputs.

Armed with the new design I scheduled meetings with my manager, partner team's manager, and the customer (the manager of datacenter contract guard force).The customer meeting was important because what we promised - locations listed in the work assignment - was different than what we were set to deliver - a generic link to our mapping tool in the work assignment. I scheduled a 1:1 call as soon as I had a clear idea of the new design.

In the end the necessary changes approved and my partner team contact received more development time to update his team's system. Though the new design differed from the previous we were able to keep the customer informed and deliver a solution that met their requirements.

**5. What is your experience with version control systems such as Git**?

I used Git for version control extensively for my own notes and documentation even before studying at the University of Washington and working at AWS. My experience maintaining repositories and tracking changes has only grown since. At the University of Washington I used GitLab to keep track of changes and submit assignments on a class/project basis and at Amazon Web Services I used code.amazon.com to view branches, search across the entire company's codebase, and follow up with engineers based on Git blame attributions attached to specific code changes.

**6. Have you worked with Agile methodologies? If yes, in what capacity?**

Yes. At AWS I participated in standups, sprint planning, issue grooming, retrospectives, operational excellence (OE) meetings, and larger-scale discussions of operating plans at our organization. Though our team focused on internal infrastructure we included user stories whenever relevant. Managers, tech leads, engineers, and applied scientists worked together to accomplish as much as possible in our two-week sprints.

**7. Can you provide examples of how you’ve collaborated with cross-functional teams on a project?**

The answer to question 4 is relevant here. To add another example there was a SQL query that I noticed was taking a long time to run and not only that was failing to produce the data we required. I dove into the issue and found our business intelligence partner team had updated the structure of their database and failed to update all of the previous database's consumers. I followed up with them to design a new query that fit our data access pattern better and ran more quickly than the last.  
  
Half my team was also made up of applied scientists. We worked together frequently to research and develop new features like automatically recognizing if someone had experienced a workplace accident or was carrying a server past a camera.

**8. Have you deployed applications to cloud platforms (e.g., AWS, Azure, GCP)? If yes, describe your experience.**

Question 4 applies again here. For the cross-team datacenter hall mapping project I deployed a new Lambda service, maintained short term logs in a DynamoDB database auto-scaling on-demand, and stored the daily results in S3 long term storage. I used Quicksight and Cloudwatch logs to display and track the performance and wrote metrics and tests to raise alarms when performance fell below an expected baseline. These I added to existing dashboards in Cloudwatch and on the AWS account responsible for the new service. I also SSHd directly into individual hosts to check specific regions and make sure that the service was working properly in the AWS regions in which we were rolling out. I worked with our distributed job scheduler (DJS) as well to instantiate AWS resources through Python scripts on a regular schedule.

**9. What strategies do you use to ensure code quality and maintainability?**  
  
Test-driven development and iterative changes are key to keeping code readable and functional. I write comprehensive unit and integration tests to define expected behavior and make maintenance as painless as possible for myself and future developers. My code is modular and reusable with appropriate encapsulation. Whether working on a team or contributing to open source projects I participate in code review and provide feedback to other contributors. And most importantly I understand the problem we're solving top to bottom and write documentation at multiple levels of abstraction. Knowing the customer's needs and checking in regularly is key to make sure we're solving their problem and not just our own.

**10. Have you ever set up or worked with CI/CD pipelines? Please elaborate.**  
  
Software teams at Amazon do not separate devops and engineering roles. At AWS I was regularly responsible for instantiating new CI/CD pipelines in AWS CodePipeline, staging rollouts by region, and monitoring progress of code changes in my team's pipelines. We pushed changes multiple times a day with a pause Friday afternoon to Monday morning to minimize our response time to incidents. This made quick patching much easier than it had been in the past.

**11. How do you stay current with emerging software development trends and technologies?**  
  
I am a voracious reader of tech content on various outlets. I find white papers posted on LinkedIn, Twitter, Reddit, YouTube, and ArsTechnica, follow projects that I find interesting on GitHub, message friends at other companies to get a high-level view of the industry, and advocate for new tools and approaches if our customer or team needs have outgrown what we are currently using.

**12. What tools or technologies do you use for task management and team collaboration in a remote setting?**

At Amazon I kept track of issues with SIM and ticketing systems, managed sprints in another system with points to track effort by task, messaged teammates over Chime and Slack, and kept a whiteboard by my desk opposite the webcam in my home office to sketch out ideas during virtual collaboration with teammates. Learning how to work asynchronously with teammates meant not even a time zone across the world could interrupt our productivity.

**13. Can you explain how you would optimize the performance of a slow-running application?**

Question 3 is pretty relevant here. Like any software issue, poor performance is a departure between a developer's understanding of the code and its actual function. Ideally whoever wrote the application would have already set up metrics and logging to understand exactly which part was running more slowly than expected. If not then I would add timed logs with a debug flag to make sure they only run while testing and not in production code.  
  
If the issue was an inefficient database query I could reduce the scope of queries and add paging to allow processing of large inputs without needing to wait for the whole result to finish. If no specific code optimizations were possible I could scale cloud or on-premise resources, but this again would ideally be already covered by an on-demand scaling model in our cloud provider settings. If our application was running on a single on-premise server I would look into spreading the workload across multiple virtual machines or containers in a distributed computing model.

**14. How do you handle conflicts or differing opinions within a development team?**  
  
Empathy plays a huge role. I do not always have the whole picture and listening to the perspectives of all my teammates gives us the best chance of coming to an informed decision together. Data is always great for backup but even if a teammate is still collecting data I still take their concerns seriously. We might consider escalating to a manager or further if we encounter a true deadlock that can only be solved by a decision by the company as a whole.

**15. What do you consider the most important factor for success in a fully remote engineering role?**

Communication. Keeping each other updated regularly, collaborating asynchronously (asking a question instead of asking for permission to ask a question), and documenting our work.

**Salary:** $90/hr. And are paid semi-monthly. The paydays are the 15th (or the last working day before the 15th) and the previous working day of each month. Training is $70/hr. Hours of work: 40 hours, and employees are entitled to overtime pay for all hours worked beyond 40 in a work week. You will be eligible to receive the following benefits.

**Benefits**

Health benefits include Medical, Dental, and vision.

Comprehensive In-House Training Program

Competitive salary and benefits package.

Opportunities for professional growth and career advancement.

Collaborative and inclusive work environment.

Access to ongoing training and development programs.

Tax-Deferred Annuity 403(b) 401(k)

**Additional Information**

At the Company, we value the following:

1. Investment in you

2. Offering opportunities for career growth and personal development training

3. Open-minded management

4. Empowering employees by listening and responding to ideas, issues, and approaches Flexible work-life balance

5. Offering a generous PTO allotment, holiday shutdown, and flexible work options

6. Diversity of cultures, perspectives, backgrounds and experiences of all our team members. Through our diversity, equity and inclusion initiatives, our focus is to strengthen our company culture to create more inclusion and belonging for all.

7. Benefits for our team members. Excellent medical and nonmedical benefits, 401(k) matching, pet insurance.

The Company provides equal employment opportunities to all employees and applicants for employment and prohibits discrimination and harassment of any type without regard to race, color, religion, age, sex, national origin.