SOP:

My intelligent and highly independent 78-year-old neighbor still doesn't understand why his email stopped working. When he asked, "Hey CJ, can you come fix my computer?", I tried to explain the layers of software changes that had likely caused the issue. Halfway through, I realized that the problem was really a failure of the developers to build and release software in a way which ensures that changes don't break the application. My passion is to build release processes which ensure software systems behave in a reliable way, every time. This is still a problem which has yet to be solved well, as evidenced by the number of software bugs we encounter daily. Today, I lead the build and release team at Cboe Global Markets, shipping software nightly to servers tasked with processing nearly $100 billion in trades daily across exchanges worldwide. My long-term vision is to enable a near-constant, process releasing hourly to production while ensuring stability.

My motivation for pursuing a Master's degree in Computer Science is twofold: first, to deepen my technical skills to allow for the building of more advanced systems; and second, to develop my leadership skills to guide complex, large-scale and cross-functional initiatives. As such, the Rice MCS program appeals to me for its integration of technical and business coursework. When proposing large projects that require significant capital and personnel resources, I've realized the importance of communication, delegation, and ability to articulate impact in a way that business leaders can understand. I intend to pursue the Engineering Leadership specialization to grow into a more effective engineering manager and technical leader.

From a technical perspective, I'm particularly excited about coursework in cybersecurity and data visualization. Cybersecurity is a huge focus for my company, and the systems I manage are inherently high-risk due to delivering the code for production machines. I hope to leverage

this course to harden our systems and permissioning models. Data visualization would allow me to improve our web dashboards to more clearly communicate deployment and system status in a real time format.

Although I have seven years of professional programming experience, my undergraduate background in mechanical engineering means that I have gaps in certain computer science fundamentals that I want to close through additional education. I'm able to solve issues eventually through persistence, but I lack certain concepts to do so efficiently and gracefully, every single time. For example, when troubleshooting a network issue, a colleague dug in examining system calls, routing tables, and TCP syn-ack logs, quickly isolating the core problem (packet drops on spare firewall). My initial approach relied on using iperf3 and looping through required ports, costing valuable time. Rice's MCS program will empower me to understand the whole system, and to be surgical when debugging rather than relying on stumbling into the issue.

Professionally, I believe my background has prepared me well for the rigor of this program. I manage a hybrid CI/CD cluster spanning 35 on-prem and up to 200 ephemeral cloud instances that support thousands of daily commits to a monorepo. We release C++, Python, Java, JavaScript, database changes, and Linux package upgrades to thousands of servers every day. This allows us to achieve 100% uptime across our global exchange environments. We write everything as code, turning every bug into an integration test to ensure our systems can't regress. Lastly, my code is highly scalable and resilient, supporting hundreds of developers with builds, releases, and patches with only three other build and release colleagues to support me.

The system I inherited suffered from years of accumulated tech debt, namely being out of date by over three years with major version releases, integration suites taking 2.5 hours, frequent

node crashes due to running out of disk, builds failing inconsistently due to lingering environment files, and hundreds of outdated and end of life plugins.

Over the past year, I fixed all the above issues and:

- Reduced pipeline duration by 60%, saving over $650,000 annually with cloud caching.
- Converted our infrastructure to config as code with weekly rolling updates.
- Created a machine learning model which predicts the most relevant tests, utilizing historical results to run a dynamic smoke test suite per exchange, saving money and improving cycles/day.

Rice's MCS program represents an opportunity to formalize the lessons I've learned through experience, fill gaps in my foundational knowledge, and develop the leadership skills required to operate at an even greater scale. Through the delivery of stable, tested software, I hope to continue enabling trusted global exchange systems, and my neighbor is able to receive email. I believe Rice is the environment I need to take the next step toward that goal.

Describe a time when you faced significant challenges or setbacks in pursuing a goal or completing a project.
How did you demonstrate grit and perseverance to overcome these obstacles and achieve success?

My first job out of college was at a healthcare startup that worked to improve hospital revenue cycle systems through the usage of programming. I created code that sat in the middle of the claim generation process and automated new insurance requirements for billing. This prevented denials, ensured revenue flowed steadily, and released humans from sitting and clicking checkboxes for what amounted to thousands of hours of their lives.

Claims are the primary way hospitals are paid. If they aren't sent out or are sent out incorrectly, revenue stops. The claim generation process had no tests, and the custom "roll-up" sits upstream of additional vendor code, which we had no control of. I was responsible for modifying claim generation logic, including the "roll-up" script to bundle various line items in specific scenarios. An error in the "roll-up", or malformed xml, would cause all claim generation and submission to halt across all hospitals. Even worse, if a piece of data was put into the wrong field, insurance would deny the claims, which might not be discovered until several days and thousands of claims had passed. This depleted cash reserves until the hospital was able to appeal the denials and regenerate the claims. Early attempts frequently caused outages despite manual testing in cert and user validation, leading to embarrassing and costly failures.

As the seventh employee of Spark Change LLC, I was deployed on-site with a major healthcare client within my first month. The client operated 12 hospitals and generated $7.3B annually; the site was also my company's only profitable client, as it was the first to hold a recurring contract with us. At this particular site visit, I was presented with a scenario in which the hospital wanted certain parameters to address a three-fold insurance billing issue. Being brand-new, I had no

idea how to fix this problem, one which the hospital's COO personally told me to "figure it out; this is what we pay you money to do." My boss, already overloaded with other things to sort out as CEO of a fledgling company, echoed this response, telling me that I just had to find a solution. While this situation certainly sent a junior software engineer into a panic, one of my personal strengths is working under pressure, regardless of the circumstances.

I decided to build a solution, an integration test framework which extracted these edge cases into test files bundled with known-good outputs. For each change, I generated claim data through the new claim generation logic, comparing with the old output and generating concise diff reports showing the before and after. Additionally, I ran the framework through all claims generated in the last 14 days to surface any hidden edge case logic.

This approach transformed our development process, dropping development cycle time from several hours to minutes, decreasing system breakages, and increasing client confidence in our team. As a junior developer, this utility allowed me to deliver changes in 1-2 days, which previously required a specialized vendor team of principal engineers. Most importantly, this experience gave me an inventive spirit that I now call on daily and an ability to stand behind my own work, and trust my skills, despite lack of experience. This situation taught me that there is no time for, and no benefit to, being afraid; I now view all actions in my career as "forward, always forward."

OPTIONAL: Within the bounds of the law, Rice University is committed to cultivating and nurturing an academic community that is rich in diversity in all forms. If you'd like to, please comment on how your previous experiences might connect with this commitment. Additionally, if you are interested in engaging with activities or academic pursuits that relate to diversity, please tell us here.

I view diversity as including rural, socioeconomic, and vocational backgrounds that are often underrepresented in academic and technical spaces. Growing up in rural Kansas, my computer courses consisted of typing and math games. Nonetheless, I found the command prompt while attempting to fix my parents' computer and was hooked at my first echo statement. Additionally, I was able to hone my problem solving through my work on our family's farm, repairing equipment, welding, and eventually working in a CNC shop as a junior machinist.

Later, a teacher introduced me to Scratch, and after seeing my passion, gifted me a C++ book. This small intervention has provided me with an extremely exciting and fulfilling career. As a late programmer, I maintain a strong sense of empathy towards others in the community who may not have had programming opportunities from a young age.

At Rice, I hope to contribute by supporting peers from nontraditional backgrounds. Previously, I volunteered teaching students Scratch through Code Ninjas and plan to continue those efforts to pay it forward to the next generation. I am also interested in teaching long-term–I have been invited to speak at the Pittsburg State (KS) University AI Symposium, reinforcing my desire to bridge technology with communities that have, historically, been distant from it.