Remember: **SPECIFIC** (give facts and hit keywords), **HUMBLE** (don’t sound cocky, let interviewer make a good impression by being SPECIFC!)  
**S.A.R. Format: Situation -> Action -> Result**

|  |  |  |  |
| --- | --- | --- | --- |
| **Common Questions** | **Project 1 (Instrumentation - USAF)** | **Project 2 (Qualcomm)** | **Project 3 (Robot Project Capstone)** |
| **“Dealing with a difficult employee”** |  |  | **S:** -Worked on a project with 3 teammates -2 excellent but 3rd didn’t contribute much (quiet during meetings, didn’t contribute on email discussions, struggled to complete components). - it began weighing down on the team **A:** I did not want to write him up too soon. So I did 3 things to resolve the situation: - First: I wanted to understand why he was acting like this. Was it laziness? Was he busy with other classes in his schedule? So we had a conversation over coffee and asked opened-ended questions to see what was on his mind. He said he was interested in doing the presentation more than the project (which was the most time consuming). - Second: He stated he feared making mistakes. I reassured him that mistakes and last-minute changes are a natural and expected to occurring during the process and we can plan for it.  **R:** With his confidence raised, he contributed more to discussions. Became interested in writing code for the components. Overall, I was able to make the team more cohesive. The final product exceeded the professor’s expectations and ultimately we won the school competition. |
| **Mistakes/Failures** | **S:** - The system had a time requirement where we had to timestamp the data at 100Hz within NanoSec accuracy. Because the timestamp of the data was used for AI modeling, we needed to ensure the time was accurate otherwise the AI model we develop as a product to the customer would be misrepresented. - In my initial design I assumed an OS scheduler with Pthreads() would be appropriate where PThreads provides an interface to set the thread priority with pthread.SetPriority(). - [it turns that modern Linux kernels require we set other parameters as well(nice parameters or use RT Policies like SCHED\_RR (RoundRobin))](https://stackoverflow.com/questions/3649281/how-to-increase-thread-priority-in-pthreads).Even so this had inaccuracies within the 1ms (an inherit issue with the CPU scheduler upon a context switch) **A:** We ended up resolving this by using an external hardware interrupt. We had a Inertial Navigational System that had a GPS component. The GPS provided a Time Synchronization Signal Pins. From this we used it as an external hardware interrupt **R:** We met the timing requirements and was able to correlate both the data and GPS timestamps. |  |  |
| **Enjoyed** | - I enjoyed the system I developed for the autonomy team. - **reqs**:  - real time requirements  - developer each sensor driver - Coordinate all the sensors with threads and interrupts  **Results:** - we achieved 60% efficiency with threading optimizations (performed timing measurement and determine where to reduce overhead by SpinLock vs Busy Waiting).  - uses RPC to telemetry data to User Dashboard as a safety feature which leverages a wave radio relay and sends data to an Ground Station Laptop | SpinLock vs BusyWait  If critical section is small and wait is shorter than a context switch it may be worth SpinLocking.  BusyWait generally is ok but could be bad especially for CPU power management. Constantly checking forces causes cache synchronization traffic. |  |
| **Leadership** | **S**: We had a new hire that came in right before the pandemic. The lead software engineer became sick and had an emergency leave around the same timeframe for a month **A:** With the lack of a lead on the project I took it upon me to ease the new engineer. - I offered that he could reach out to me whenever he become stuck or had any questions. I asked open ended questions to get a feel of what he preferred **R:** Despite our lead I saw an opportunity for process improvement. This allowed him to read documentation, become familiar with the system and slowly easing him into his first task. He even thanked me by saying he felt very welcomed to the team. |  |  |
| **Conflicts** | -new hire, training, pandemic - advocating for digital fluency | - dynamic environment working with different teams every month on a rotation | **S:** -Our senior capstone project related to designing a robot which would compete in local competition **-** In short, we came up with different designs and the team was torn between two. We could not come to an agreement **A**: - this dragged out for a week and so I took this into my own hands and organized a meeting:  -prior to the meeting I organized a proposal: design 2 prototypes with core functionality  - During the meeting we discussed the concerns and agreed that making two prototypes was worthwhile. Of course, we risked deadlines but we all felt that this was a necessary step to achieve a **Minimal Viable Product** **R**: - we were able to get both prototypes built, tested and come to a conclusion - We selected the design with 4 wheels = more torque which was key in winning the competition |
| **What you’d do Differently** |  |  | - I would have ordered several spare parts  - sometimes the motors drivers would break. Ordering parts often led to a week of downtime waiting for the parts to come in  -sometimes the sensors had manufacturing defects so some of the sensors had different Signal To Noise ratios |
| **Technical Decisions** | - ship back a sensor (which requires a firmware update from the vendor to resolve timing issues) or use an external interrupt from the GPS Sync signal - the vendor noted either way was ok but we were unsure if this would be an issue with a potential client using the system in the future  - we ended up using the GPS Sync Signal as shipping the sensor had an approximate 1 month delay  - we noted in the documentation that the time source is based off the GPS so external customers who wanted to use our system/service came in understanding this |  |  |
| **Choices of Technology (Trade-offs)** | Web UI Dashboard: Prometheus vs InfluxDB  - ideal for timeseries data; both free  - both had proprietary query languages **Prometheus + Grafana -** easy setup - limited scraping data rate to 1HZ **InfluxDB + Grafana** - more effort to get setup - excellent for large data sets - more visual customization  In the end we went with InfluxDB primarily because the 1HZ limitation. With different customers coming in they may have different requirements and setting up the docker container frameworks starting from scratch would be working backwards. |  |  |
| **Genuine Questions** | | | |
| **Introduce yourself:** **Name:** Hello my name is Adrian Alonzo.  **Current Role:** I am an Electronics Engineer at Edwards AFB. I work in the autonomy team developing c++ aerospace applications for AI data modeling efforts. I have 3+ years working in Aerospace for a total of 4 years working in Defense.  **Background:** My undergrad background is Electrical Engineering at San Diego State University. I spent a semester working closely with a professor on mobile development and data analysis. In addition to that some friends and I attempted to launch a startup. I developed the android application that communicated to a SQL server. I am now a Computer Science graduate student at Georgia Tech Institute of Technology specializing in computing systems. My favorite classes include Software Design and Operating Systems. **Outside of Work/School:** - I like to contribute to an open-source software. Right now I am working on sprite graphics importer (Unity-Aseprite-Importer) which improves the toolchain for smaller game developers to help ship their products faster.  - Every now and then I like to partake in hackathons and Leetecode. | | | |
| **What are your weaknesses?**  - can be a slow to learn new concepts; need to visualize things **S:** - I am a visual learner and so when learning new concepts, I need to visualize things.   I need to see the big picture. For Example: **A:** - So prior to a meeting I create diagrams. (this includes UML class, component, sequence and even wiring diagrams for the hardware)  - the diagrams organize and gather my thoughts.   - “Afterall it is always important to stretch before running a mile”  - This helps me ask important questions to narrow my focus–  - “Do I really understand the problem?” – if not then loop back and ask. If its with a customer then open the dialogue to gather more requirements  - or asking “How do we decouple the design?” – this helps think about isolating logical layers in the program design   Ex. separating sensor driver code from the networking layer. In most cases the two should not be tightly coupled **R**: In the end being a visual learner is my biggest weakness. It is something I identified as something to continue to improve and work on | | | |
| **What are your strengths?** -I embrace feedback. I always aim to listen and learn from my peers. So as a team member I am very open minded.  -I am patient. I understand that everyone has different learning styles, needs and backgrounds. This is important in building strong relationships and trust. With that I am dependable and trustworthy. -I am passionate in what I do. I always look for ways to improve myself and better integrate myself into the team. -I am Caring. I care about the product I work on and people I work with. I like to build strong working relationships that makes the overall experience better for everyone. | | | |
| [**Why Do you want to work here?**](https://novoresume.com/career-blog/why-do-you-want-to-work-here)(usually a filler with no follow ups. *Hint relocation to show you will stay long term*)  **Small company:** I want to be in a place where I'm not just a cog in the machine, and can have plenty of opportunities to grow. I want to exceed my current responsibilities and wear multiple hats **Big company:** I want to be in a place where my work has a dramatic impact on millions of users. I noticed your company values being open-minded which is what I seek for in company culture and I think I would be a great fit. I also have family around the area and plan to remain here long term. | | | |
| **Why should we hire you?** (Understand employer’s needs/ echo job description/ be prepared for follow ups) **Passionate:** - I am self-motivated and passionate. And this is reflected in my work ethic and quality of my code.  - I enjoy working in a fast-paced environment and seeing quick results.  - I simply love working with others who share the same passion.  **Motivator:** -I also am an excellent team player and motivator. it is always a pleasure working with others. The working relationships I develop is like a family. I want to look out for them just as they would look out for me.  - Lastly I have good communication skills.  - I know what questions to ask when I don’t understanding something.  - I can communicate both technically and non-technically using visual diagrams help deliver a message | | | |
| [**Where do you see yourself in 5 years?**](https://careerkarma.com/blog/where-do-you-see-yourself-in-5-years/)(helps employers answer if they can provide what you are looking for,  - I see myself as a hard-working experienced software engineer that is open minded. - I see myself as a good mentor with excellent working relationship made over the years - I see myself having the qualities of a leader and can take on responsibilities that are given to me | | | |
| **Agile:  -**a project development methodology used by software developers - **Phases:** requirements gathering and interface definition, design, development, test and integration, deployment and short/long term review  -includes division of tasks (1,2 or 3 hours of work). These boil down to features we wish to implement and are characterized into sprints (duration of time sometimes 2-4 weeks).  **-** Constant reassessments are made for what is needed now (as requirements and changes can be made from client/our-side)- within each sprint we deliver working software that we build on top of for the next sprint  [methodology 5 types](https://www.xpand-it.com/2018/10/11/top-5-agile-methodologies/): Crystal, XP, Kanban and Scrum (follow similar agile template but with twists) **Example:** Have a design an application payload for a customer platform. Depending on their test needs they will have various sensor and so different requirements **Create Sprint:**   - I would break down application into user Stories of smaller chunks of features   - User Stories are then broken down into tasks (tasks are placed in the backlog ex. backlog in Jira)  -then we create a sprint in X weeks  - drop in tasks and user stories for the sprint based on the prioritization set by the PM/Client **Developing Sprint:**  - each day coming into work and pick tasks for the day  - go into daily standup and explain what I will work on  - talk about: what I did yesterday, what I will do today and any task impediments preventing moving forward  - Completed task includes code, test, successfully build, pushed, submitted for peer review  - Once a task is complete then I would pick up next task and begin working on that  **Ending Sprint:**  - then I have meeting with clients/team to see what needs to get done for the next sprint **Process Repeats:**  **-** we want to design a payload to mine data with sensors x,y,z and a 2 week sprint - we have sensors order previously so we use those: INS (IMU/GPS) and a fuel flow meter - we look at hardware specs (voltage, current draw, device interfaces: UART, analog, digital) and if everything checks off all the boxes then we begin code design **-** we design each component using UML diagrams (component diagrams)   - then we begin code development. We have several days to finish the sensor development in the first week   - as each task is complete we unit test the code - in the second week say the hardware arrives on Monday we then take the code   **What is your favorite phase?** ~~- all phases of the software development lifecycle including requirements and interface definition, design, development, and integration of embedded software for space-based applications, and product maintenance.~~  My favorite phases are design and code development. I enjoy creating class diagrams and designing systems. I also enjoy translating that that design into code and constantly improving the design as development progresses. | | | |
| **Questions To Ask Interviewer** | | | |
| **"What is the ratio of testers to developers to program managers? What is the interaction like?  How does project planning happen on the team?"** | | | |
| **"What brought you to this company?  What has been most challenging for you?" What percentage would you say the portions of the systems are in-house (from other Boeing facilities) vs external customers. Are hardware simulators available? What procedures are put in place to adapt to the COVID situation?** | | | |
| **Insightful Questions** | | | |
| **~~"I noticed that you use technology X. How do you handle problem Y?"~~** | | | |
| **~~"Why did the product choose to use the X protocol over the Y protocol? I know it has benefits like A, B, C, but many companies choose not to use it because of issue D”~~** | | | |
| **~~Passions Questions~~** | | | |
| **~~“Im very interested in scalability, and I'd love to learn more about it. What opportunities are there at this company to learn about this?"~~** | | | |
| **~~"I'm not familiar with technology X, but it sounds like a very interesting solution. Could you tell me a bit more about how it works?"~~** | | | |

**Mutex**: locking mechanism to synchronize access to resource. “I will try to lock access to resources”. 1 task can acquire the mutex. It means there is ownership associated to the mutex and only the owner can release the lock.

**Semaphore**: a signaling mechanism. “I am done, you can carry on”. For example if you are listening songs (assuming 1 task) on your mobile and at the same time your friend calls you, an interrupt is fired with an Interrupt Service Routine (ISR) signals the processing task to wakeup

**DeadLock:** When you have 2 processes waiting to access a resource and that resource lock is never released.  
Tip: To avoid this always pair a lock with an unlock. Avoid double locking!  
If using in c++ utilize the lock objects (unique\_lock or guard\_lock) uses scope to determine life of object. (Upon destructor, the lock is guaranteed to unlock the mutex. Good for crashes)  
 - uses Resource Acquisition is Initialization (RAII)  
  
**Race Condition:** When you have 2 processes that rely on a single state change as the output. For example two processes occurring that increment by 1. If no mutex is used OR operation is not atomic then race condition occurs.

- atomic = the instructions executed will start and complete without the interruption of another thread  
 - in this example we want only one increment to occur at any given time. if an increment occurs we expect 1 count

For those projects, and all your projects, be able to talk about the challenges, mistakes, technical decisions, choices of technologies (and tradeoffs of these), and the things you would do differently.

References:

* [35+ Common Interview Questions and Answers [Complete List]](https://novoresume.com/career-blog/interview-questions-and-best-answers-guide)
* <https://www.job-hunt.org/job_interviews/answering-why-work-here-question.shtml>