1.「与上级/同事产生分歧，最后达成一致并取得成果」（情境1）

- **核心看点**：冲突 -> 坚持己见/寻求妥协 -> 行动过程 -> 最终结果/反思。
- **可覆盖的领导力题目**：
- **Have Backbone; Disagree and Commit**（所有围绕"坚持、冲突、妥协"的题）
- **Earn Trust**（你如何沟通、化解分歧，最终让对方相信你的想法）
- **Are Right, A Lot**（如果过程中涉及到你如何做决策、判断，对错结果的复盘）

Situation:

During the Online Coding Judgement System project—you can image it as a very simplified version of LeetCode—I was responsible for the backend development while my teammate focused on the front end. As the project going, I've found several places to do the optimization, and I want to implement them. So I turned to my teammate for advice.

However, my teammate was concerned that these changes would add additional work and delay our progress , as she preferred a quick finish to showcase this project on her resume.

Task:
Because I really want to learn sth. from my project, but not just do coding, I need to convince her that these optimization was worth the extra effort, though it may delay the progress a little bit.

Action:

1. **Open Discussion:**
   So, I had a conversation with my teammate to fully understand her concerns. She explained that a quick project completion was important for resume purposes because it is really hard to get a job or even an interview right now. And the time would be the key in this job market.
2. **Data-Driven Explanation:**
   Then, I expressed my view about the project, I want to make our system better, and closer to a real-world solution so that our work was meaningful. Also, I shared several examples from my previous project, where the optimization would greatly improve the performance. She wanted to be a full-stack SDE, learning these would benefit her career.
3. **Proposing a Compromise:**
   To balance our priorities, I proposed we could only optimizing the part that would not greatly alter our design, which is decoupling the code judging process by using a message queue. After evaluation, we thought it was not that difficult and risk of failure was low;

Result:

After reaching an agreement, I quickly implemented the change. The optimization just took three days, which didn't affect our progress too much. The decoupled judging process greatly improved the system QPS by 32% under stress testing. Both my teammate and I were very satisfied with this outcome.

有什么优化处：such as transforming the system into microservices, developed individual coding sandbox, decouple workflow.

原来的优化：Our initial design allowed the system to handle code submission, execution, and result retrieval in one flow. However, I identified an opportunity to significantly improve performance by decoupling the judging process using a message queue.

2「资源/预算严重不足，想办法节约成本或创新搞定」

- **核心看点**：问题 -> 资源限制 -> 创造性解决方案 -> 结果/成本节省/收益提升。
- **可覆盖的领导力题目**：
- **Frugality**（所有"省钱、省时、省资源"的题）
- **Deliver Results**（在资源不足情况下如何依然取得结果）
- **Bias for Action**（如果强调你在资源不够的情况下，短时间内快速做决策并行动）
- **Dive Deep**（如果过程中有较深入的数据分析或根因排查

Situation:

For my final year project, which was using machine learning to analyze 3D microscope images. By analyzed the distribution of protein, researchers can predict cell's life-cycle stage accordingly. At the beginning, I assumed that the core challenge was 3D image segmentation so I decided to work based on this. I built a complex solution that using 3D ML method to analyze while using several algorithms to make the dataset consistent. This solution works quite well from my side. And it was over half of the project timeline, so I decided to deploy it to lab's devices for testing.

However, I found that differences in lab devices led to dependency problems and many of them cannot run this solution. And most computers didn't have a GPU, so it took several minutes to run and the accuracy was only less to 80%.

(用户体验：而且，我觉得现在的deployment was difficult for the researchers that don't have CS knowledge)

Task:
I quickly realized that

(I failed to consider the real-world factors. It was not a coursework that under ideal situation, and now the computing resource was limited. I need to quickly adopt a new solution before the project completely fail.)
(the computational resources were limited. Also, it seems that the dataset was not enough to perform a good training.)
(the user experience was bad because of the difficult deployment and usage and long waiting time.)

So, my goal became to design a more resource-efficient ML workflow that could run quickly on various devices while still deliver accurate results within the given time.

Action:

1. **Deep Dive & Research:** Instead of rushing to work, I investigated of the available devices and the dataset in lab. Then I reviewed several literature and consulted professors with AI expertise to understand the trade-offs between computational-heavy method with accuracy.

2. **Resource-Conscious Redesign:** Considering the limited computing and training resource, I shifted my strategy from fully 3D method to a hybrid approach—using 2D analysis combined with metadata for restoration. This allowed for much faster processing, even on CPU, while still capturing essential spatial information. Also, this process allowed me to have more training data from the limited dataset.

1. **Rapid Decision-Making & Automation:** Moreover, I communicated with the researcher and found that he nearly knew nothing about AI. So, I simplified the dependency configuration and made simple user manual so that researchers can use it easily.

Result:
The redesigned ML solution can run successfully on multiple lab computers in less than 2 seconds. And the average accuracy was over 90%. This project was delivered when the resources were limited. Moreover, this innovative approach won the prize, and eventually led to a publication in BMC Biology, which is one of the top science journal.

3「客户/用户遇到难题或比较棘手的需求，超出期望地解决」

- **核心看点**：客户问题 -> 你如何理解、沟通 -> 你如何超预期完成 -> 效果与反思。
- **可覆盖的领导力题目**：
- **Customer Obsession**（几乎所有"客户互动"/"超出客户期望"的题）
- **Earn Trust**（你如何让客户或团队对你产生信任？）
- **Deliver Results**（如果故事包含你如何把客户反馈落地成结果）
- **Bias for Action**（如果情境中客户问题紧急，需要迅速响应）

Situation:
While working on a project for AMX, the clients were struggling with low accuracy, which was over 75%, with their models. Though their tried to improve the performance by retraining it with different parameters, the results didn't show great improvement.

Task:
As one of the Engineers, my job was to determine why the accuracy remained low after training, and find out the root cause, then collaborate with my team to identify and implement a solution for clients.

Action:

1. To better understand client's problem, I hold a brief but focused meeting with them to walk through their entire process for retraining that model. During this meeting, I carefully record their workflow and potential areas of concern.
2. **Collaborative Analysis:** After the meeting, I discussed the collected data with the team. Because we found that the training parameters and processes seems correct, we assume that the quality of datasets might be the underlying issue but not the training itself.
3. - **Validation Experiment:** To validate our idea, We made an experiment by retraining the model using a better dataset. This allowed us to examine our thought quickly without greatly delaying the project. And the result showed that dataset quality was the key factor in this issue.

Though we found the reason, we knew that improve or build a new dataset with higher quality requires extra budget and time, and the clients may not pick this advice in a short time. We should find another solution for them.
    - **Immediate Enhancement:** Since the training was correct and the dataset should not be changed right now, we started with their model to find potential improvement. Because client's model was developed based on BERT's model, so we decided to optimize the BERT and then recommend this strategy to the clients as another solution, which was more cost-efficient.
4. **Communicate Findings:** After optimization, the model accuracy increased by 13% without extra cost. We then organized a follow-up meeting with the client to present our analysis and two solutions. We recommended clients modify their models first, and improve their dataset in the future.

Result:
The client accepted our proposal. They thought our solution really exceeded their expectation, cuz it not only resolved their immediate problem with less price, but also set the stage for better model performance in the future.

4「发现/面对一个复杂问题，深入分析（Dive Deep），找出根因并解决」

- **核心看点**：复杂问题 -> 数据/信息收集 -> 深入分析 -> 对策 -> 后续跟进。
- **可覆盖的领导力题目**：
- **Dive Deep**（所有关于深度挖掘数据、发现根因的题）
- **Deliver Results**（如果最后解决了问题，达成某些业务目标）
- **Are Right, A Lot**（如果故事里包含决策过程，以及为什么它是正确或错误）
- **Earn Trust**（如果你在此过程中和团队/经理积极沟通，建立了信赖）

Situation:
It was the AaronFlow project and I finished most part of development. However, during stress testing with **wrk** (测flowsvr), I noticed that the QPS was only 200, which wasn't as high as expected and also, the CPU usage was highly unstable. After using commands and checking the performance manager, I found that there were sudden spikes in CPU load, indicating that something deeper was affecting the system.

Task:
Although I had introduced and adjusted MySQL connection pools to deal with connection overhead, the problem still exist. I was curious about this issue and I decided to dive deep into this and identify the root cause that affecting CPU stability and performance.

Action:

1. **Data Gathering & Initial Hypothesis:**
    - I started by analyzing the server side, (since the worker part(Aaron) processed tasks rapidly and its operation was light.)

    - After reviewing system metrics and checking various forums about this situation, I thought the problem may occur during multi-computer competition. Then I look into the code for this part and identified the use of the **for-update** may be the potential problem.
2. **Deep Technical Analysis:**
    - After reading the inner workings of the **for-update** command, I discovered that it can generate **gap locks**. It may delay query execution and affect performance, which it is also not recommended in the book called *High Performance MySQL*.
    - I then focused on the unusual CPU spikes. Because performance manger showed that, the CPU load spike when servers tried to fetched tasks. This led me to dive deeper into the MySQL. I found that MySQL used an automatic deadlock detection that checks for dependencies between queries. As the number of concurrent tasks increased, this mechanism caused high load on the CPU.
3. **Implementing the Solution:**
    - To address both the gap-lock issue and the detection overhead, I dropped the lock produced by the MySQL and decided to implement by myself.
    - After reviewing multiple forums and articles about multi-computer competition, I introduced a **distributed locking mechanism** to better manage competition. Also, I introduced **randomized delays** in task invocations to spread out the load.

Result:
The adjustments led to a great performance improvement: CPU became stable under load, and the system throughput improved by 5 times. By diving deep into the problem of **for-update** and the deadlock detection, I not only solved the immediate performance issues but also gained valuable experience.

5「时间或外部压力极其紧迫，你必须快速行动/决策」

- **核心看点**：突发紧急状况/极限期限 -> 快速抉择 -> 行动 -> 结果/教训。
- **可覆盖的领导力题目**：
- **Bias for Action**（在时间紧迫、信息不完全的情况下如何快速行动）
- **Deliver Results**（高压力或临时突发下依然完成或超额完成目标）
- **Are Right, A Lot**（如果当时信息不足，怎么权衡风险并做出正确决策）
- **Earn Trust**（如果故事中有协调多个团队、赢得他们支持的过程）

Situation:
In my Human-Computer Interaction (HCI) course, our team was working an AI-driven project that provided personalized cosmetic and makeup recommendations. Two days before our scheduled presentation, one of the professor shows her interest and believed that our project had the potential to compete for the award. And our supervisor suggest we could have a more detailed presentation, but not just using poster.

Task:
Although the deadline was tight and our team only had four members, we were really excited and wanted to get the prize. So we decided to take supervisor's advice to improve our presentation.

Action:

1. **Rapid Assessment and Decision-Making:**
    1. We immediately held a meeting to discuss about the form of presentation. Though we all agreed that an high-fidelity demo would make a great presentation, the limited time and lack of skills made it difficult to achieve within 2 days after evaluation. So, we decided to make a simple interactive demo while keeping the core features.
    2. Then, we began to find a tool that, could help us achieve this goal within DDL.
2. **Leveraging the Right Tool:**
    - After a brief searching, we discovered Figma—a popular platform among UX designers for creating interactive prototypes—which also supports real-time collaboration, so that we can separate the jobs and working on them together.
    - Meanwhile, I reached out to a friend with UX skills to give us with a brief Figma tutorial, so that we can start working quickly and get technical support during development. Although none of us were experts in Figma, we saw its potential to rapidly produce a good demo.
3. **Structured Planning and Collaboration:**
    - Since we already have a poster version presentation, we quickly decided the process and key features that needed to be shown. After that, we divided the tasks according to member's interest and understanding, and then started to construct the demo.

Result:

- Finally, our team successfully created an interactive demo within two days. And the presentation impressed our professors and other students. As a result, our project not only received high praise but also the "Most Creative Design" award from our department.

Lessons Learned:
Under limited time, a structured plan and effective job allocation are very important. But, I would like to change the way we separate the task. We allocated the tasks according to our interest. However, this requires us to learn the entire process, which increased learning and communication cost. So, I would like to allocate the jobs according to functions. Everyone focus on a single implementation. This way, we could streamline the workflow and improve efficiency.

6「犯了错误或决策失误，之后如何补救和复盘」

- **核心看点**：错误决策/执行 -> 造成的影响 -> 采取的补救措施 -> 反思/改进。
- **可覆盖的领导力题目**：
- **Are Right, A Lot**（并不总是对，错误的决策如何复盘与学习）
- **Earn Trust**（你如何对利益相关者坦诚、挽回影响）
- **Dive Deep**（如果你在纠错和复盘过程中用了深入的数据分析）
- **Deliver Results**（如果你在有限时间内尽量把损失降到最低，也是一种“交付结果”）

Situation:
During my junior year as an undergrad, I worked in a team of three to develop a wearable navigation device, which designed to provide users with direction without distraction. Because this project involved both hardware and software development and we had to order various components from the website. We started developing the software first, while waiting for the order.

However, after completing and testing most part of software, we found that the pandemic disrupted the deliveries, and many core components, even the motherboard, would not arrive on time or even out of stocks.

Task:

Instead of waiting, I suggested we must take immediate action and make a corrective plan to mitigate this impact, cuz kept waiting may just let our efforts useless.

Action:

1. **Immediate Assessment and Communication:**
    - Firstly, I immediately checked all existing and available devices in the lab. And we carefully analyzed the gap between our initial plan and the current available resources.
    - Then, we discussed and assessed possibility to move the software to the remain platform. We also found that some features would need to be dropped due to hardware limitations. Therefore, we narrowed down our target users from anyone to those who takes the similar routes every day.
2. **Team Collaboration and Deep Analysis:**
    - After that, we wrote a report about this situation, and discuss with our supervisor, to seek approval for modifying our project goals and requested guidance for revising our timeline.
3. **Developing and Implementing a Mitigation Strategy:**
    - After getting the approval and evaluating the new project timeline, we decided to do the transfer while only maintaining the key features.
    - Also, we re-designed features according to limited device, and combined with Karman Filter Algorithm to compensate for the low precision and the unfinished feature, so that we could achieve similar functions under this condition.
4. 有啥改动
    1. Transfer from Raspberry Pi to Arduino, and using Python instead of C++;
    2. Initial design, we planned to using hi-precision GPS and map API to get the path, and using vibrator to indicate the direction.
       Now: Existing is not that advanced, and frequently lost its location. Also, we shift our target people to the disabled, whose daily routine were quite fixed. Therefore, we adopted fix paths for directions, and used acceleration sensor to count the steps and direction to judge whether the used was on the right track.

Result:
Despite the setbacks, we successfully completed the project within the revised timeline. The final product, though cannot specify the path, was able to provide non-disrupted navigation. Moreover, we improved the device's positioning precision under low signal conditions by 10 times, and the accuracy increased by 6 times by using the Karman Filter. We were satisfied with this outcome.

Lessons Learned:
This experience taught me never to underestimate potential risks. I learned the importance of conducting a thorough risk assessment, seeking guidance from experienced mentors, and maintaining open, honest communication with all stakeholders. The incident reinforced that proactive collaboration and a willingness to adapt are crucial when unforeseen challenges arise.

7「协调跨团队/跨部门目标不一致，最终达成共识并取得阶段性成果」

- **核心看点**：多方目标冲突 -> 沟通/谈判 -> 达成一致 -> 后续成果/教训。
- **可覆盖的领导力题目**：
- **Earn Trust**（和不同团队合作、对齐、互相信任的过程）
- **Have Backbone**（如果中间经历过分歧，你坚守了什么，或怎样“Disagree and Commit”）
- **Deliver Results**（最后如何保证项目如期完成）
- **Customer Obsession**（如果此过程与客户需求或体验有直接关联）

Situation:
During the development of our **Snake Game & Leaderboard**, our six-member team had a conflict regarding the scoring mechanism. The data team was pushing for a **complex scoring system** to make the game more engaging and to get more diverse data for advanced analysis, while I preferred a **simple scoring system** to ensure a quick launch, because I am the only one developer for this part and I was afraid to fail to meet the timeline. This conflict was creating tension and delaying our progress at first.

Task:
Because we only have two developers and we didn't want disagreement to hinder our progress, We need to resolve the conflict and guide the team toward an agreement, that balances technical feasibility with data requirement. And I needed to ensure that we can deliver a working Minimum Viable Product (MVP) on time without compromising on quality.

Action:

1. **Clarified the Project Timeline:**
    - I reviewed our Gantt chart and overall timeline with the project manager, confirming that implementing a complex scoring system require few days and risk missing our deadline.
2. **Facilitated Open Discussion:**
    - Then I organized a quick meeting to allow everyone to share their opinions and concerns. I actively listened to the data team, recognizing their desire to perform advanced analysis and enhance gameplay while emphasizing the importance of a timely launch.
3. **Proposed a Balanced Solution:**
    - To ensure we met our deadline, I proposed implementing a simple scoring for the initial release. At the same time, I applied OOD principles and the design pattern, so that it is much easier to introduce new features(—such as difficulty-based bonuses—in the future) to meet their requirement.

4. **Validated with User Feedback:**
    - To ensure our product aligned with user preferences, while I was working on initial launch, the data team can conduct a brief user survey about the gameplay at the same time. This data-driven step can help to clarify the advanced mechanism and support fast implementation in the future since a clear plan would decrease the communication cost.

Result:

- **Timely Launch:** As a result, We successfully launched the game with smooth gameplay on schedule, while leaving room for future enhancements.
- **Foundation for Future Updates:** And the advanced features designed according to the survey was quickly implemented, because the data team setup a clear plan and also we held regular meeting to make sure we're on the same page.
- **Improved Team Collaboration:** I think, this experience strengthened trust and cohesion within the team, as all members saw that their concerns were heard and respected.

Reflection:
This experience taught me three key lessons:

1. **Focus on Core Goals:** Prioritizing a timely, high-quality launch is critical—even when team opinions diverge.
2. **Design for Flexibility:** Building extensible systems from the outset enables future enhancements without major overhauls.
3. **Data-Driven Decisions:** Incorporating user feedback can validate our approach and help align divergent team goals.

---

8. 麻烦顾客

Situation:
During my final year project, I was tasked with developing a machine learning solution to analyze 3D microscope images. The goal was to save scientists from several hours of manual classification, which can significantly increased their efficiency. After efforts, I successfully built, trained, and demonstrated the model's ability to meet the project's goal.

Challenge:
It was at a final presentation. As I was introducing the project, the lab researcher remarked that this work did not meet his expectations because he thought he could not use it directly. He believed that current deliverable haven't meet the goal yet, which made me feel nervous and a little bit embarrassed actually, especially in front of the audience. But I quickly realized that the client may not be familiar enough with the GOAL of the project.

Action:
To address the situation,

1. **Customer Obsession & Earn Trust:**
   I didn't point out his misunderstanding. Instead, I tried to stay calm and patiently explained that due to time constraints, this presentation was mainly focus on the ML workflow and its core principles. But if time permit, I would like to share the basic usage process, and he agreed.
2. **Ownership & Bias for Action:**

- After the presentation, I made a discussion with the client and clarified that our contract required the delivery of a working ML model—but not not a developed application. And he could directly use it with command.
- After listening his understanding of ML method, which he thought was similar to his software with GUI, I took full responsibility for any miscommunication and also I offered a solution: if a complete software was needed, I would like to initiate a new project to develop an application. Before that, I would simplified the dependency configuration and made a simple user manual so that researchers can use it easily.

Result:
The client appreciated the my explanation and advice. He also understood the difference between ML method and Software and even apologized for the initial misunderstanding. Recognizing the potential of my work, we launched a follow-up project that eventually led to a publication in BMC Biology and won an award.

---

9. 解决问题，提升效率

Tell me about a time you solved a pain point for the team (for-loop导入太慢，concurrent + pool加速，每次开发完数据库相关的功能就会慢)

Situation:
During my internship as an SDE intern, we were building a partner matching platform based on a user system. We had successfully completed the user query function at that time, and begin to test the system's performance. To do so, we needed to simulate an environment with 1M users. However, the existing data import process provided by the software was extremely slow and took nearly 5 minutes.

Task:
Since we frequently needed to manipulate database for testing and development, this inefficiency process became a major bottleneck. And I took the initiative to optimize the data import process

Action:

- I investigated the built-in function and found that it inserted records sequentially, leading to poor performance because of Transaction Overhead and Connection Overhead. So I decided to implement by my own.
- To improve efficiency, I used **batch insertion**, which greatly reduced the import time by 2 times. However, I realized that if any batch insertion takes too long or fail, it could block the following insertions because for-loop.
- To address this, I then decided to use multi-threads to do batch insertion. The idea was to process multiple batches concurrently, reducing the risk of bottlenecks.
- While this improved performance, it didn't fully meet my expectations. After checking the logs, I found that not all the threads were working and the default thread pool configuration was limiting concurrency. I then optimized it by using a **custom thread pool** to ensure all threads could perform the insertion correctly.

Result:
The optimization improve the data import time from 300s to 50s. This improvement not only resolved the immediate bottleneck but also accelerated the future development and testing workflow. Moreover, this success has made me even more eager to solve problems to improve efficiency.

10. 几个高优先级的任务一起做

Situation:
During my senior year, I was applying for graduate program and I need get high scores on my IELTS and GRE to meet requirement. Also, I chose a interesting final year project but was totally new to me, which was using machine learning to process 3D microscope images. Moreover, these things required to be done within one semester.

Task:
I had to manage these important tasks efficiently to meet deadlines and ensure quality results in each area, cuz this was essential for my academic future.

Action:
At the beginning, I broke each task into smaller pieces as daily goals, and also setup milestones. For the first two weeks, I made good progress: I remembered many new vocabulary, improved my language exam marks, and developed a basic plan for my project. However, I noticed that my efficiency decrease and felt tired to catch up with the plan.

I then sought advice from my academic advisor and peers. They pointed out that while my task breakdown was good, my daily schedule did not match my energy levels, which means my performance varied within the day.
To fix this, I observed my day and re-scheduled the plan accordingly:

- **Morning:** In the morning, my memory was sharp, I focused on memorizing new vocabulary and learning the courses that day.
- **Afternoon:** In the afternoon, I found that I can easily focus on a thing, so I used this time to practice for tests and worked on coding.
- **Evening:** As for the evening, when I felt most creative, I brainstormed and developed solutions for my project.
- To further increase the efficiency, I also used the Pomodoro workflow to stay fully focused on work. And I set an extra hour in the afternoon for exercise to recharge and review my day. Additionally, I reviewed my progress weekly and adjusted my plan as needed.

Result:
By efficiently adjusting my schedule and seeking advice, I was able to pass all my language exams and get admitted in the Cornell, and deliver a successful final year project that led to a publication in the top science journal. This experience taught me that handling multiple priorities isn't just about task breakdown—it also requires smart scheduling and maintaining physical and mental energy.