Summary of my stories:

1. Infrastructure/Platform Experience

- Story 5 demonstrates strong CI/CD pipeline development experience

- Story 1 and 2 shows experience with complex system architecture and integration

2. Technical Leadership

- Story 4 shows implementation of automated testing systems

- Stories 2 & 5 demonstrate architecture planning and technical decision-making

- Stories 6-8 show strong technical trade-off analysis capabilities

3. Cross-Team Collaboration

- All stories demonstrate working across multiple teams/organizations

- Story 3 shows successful coordination across 10+ domain teams

- Story 1 and 2 shows collaboration across 2 VP organizations

4. Project Planning/Execution

- Consistent track record of delivering complex technical programs

- experience with technical design reviews and estimation

- Demonstrated ability to manage dependencies and risks

5. Communication Skills

- Clear communication of technical concepts

- Experience presenting to executives and stakeholders

- Strong documentation and planning skills

Other data points:

- Strong cost optimization experience (Story 8)

- Risk management capabilities (Story 9)

- Experience with strategic pivots (Story 10)

STORY 1: CDR (Critical design review) for Aerospace program

Situation:

As lead for a critical Department of Defense (DoD) Contract Design Review (CDR), I was tasked with coordinating a major milestone review that would determine our design's readiness for system development phase. The CDR required extensive preparation, documentation, and demonstration of design maturity across hardware, software, and system integration domains, with strict compliance requirements from U.S. government stakeholders.

Task:

Primary Objectives:

- Analyze and satisfy all contractual CDR entry/exit criteria

- Coordinate deliverables across multiple technical domains

- Manage complex stakeholder ecosystem including internal teams, Air Force, Navy, and external partners

- Secure formal approval to proceed with system development

- Meet strict documentation submission timelines (15-45 days before review)

Action:

1. Initial Requirements Analysis and Planning

- Performed detailed review of government contractual documents

- Mapped comprehensive CDR criteria into 8 major categories:

  A) CDRL Documentation Requirements

  - Identified submission timing requirements (15-45 days)

  - Created tracking system for submitted vs. pending documents

  - Assigned ownership and deadlines for remaining deliverables

  B) Hardware Design Maturity

  - Coordinated PCB design completion status

  - Worked with supply chain for prototype hardware procurement

  - Aligned with factory teams on build equipment readiness

- Created detailed hardware bring-up and testing schedule
- Established lab readiness plans

C) Software Design Readiness

- Verified design compliance with government requirements
- Coordinated design review modules readiness for review

D) System Reliability

- Planned Highly Accelerated Life Testing (HALT). This was done via simulation of UUT.

E) System Requirements Compliance

- Organized system-level requirements review
- Created traceability matrices

F) Test Equipment Readiness

- Aligned on test equipment plans to test flight hardware/software test capabilities
- Coordinated test equipment procurement and setup

G) Testing/Certification Planning

- Developed comprehensive test strategy
- Created certification roadmap

E) Implementation Planning

- Developed detailed build-out plan
- Mapped future milestone dependencies

2. Program Management and Stakeholder Coordination

- Established bi-weekly tracking cadence with all technical owners
- Created master workback schedule integrating all domains
- Conducted preliminary internal alignment meetings
- Held multiple engagement sessions with DoD customers for criteria alignment
- Scheduled and structured one-week review event
- Coordinated participation of 150+ attendees across organizations

3. Risk Management and Issue Resolution

Hardware Challenges:

- Identified parts delivery delays through regular tracking
- Collaborated with factory to optimize resource allocation
- Prioritized critical board bring-up activities
- Managed power/weight specification non-compliance (over by 1.5W and 14 ounces) :
  * Led detailed technical analysis
  * Developed exception request documentation
  * Created Path to Green (PTG) with owners and actions and ETA to bring closure
  * Secured stakeholder buy-in for path forward

4. Mechanism

- Implemented comprehensive one-week dry run review
- Organized detailed technical rehearsals by domain
- Collected and tracked implementation of feedback
- Conducted pre-review documentation audit
- Established rapid response process for review questions

5. Review Execution

- Managed 5-day review schedule

- Facilitated technical presentations across domains

- Organized real-time sidebars for issue resolution

- Tracked and categorized all action items

- Maintained detailed minutes and decision log

Result:

Immediate Outcomes:

- Successfully completed 5-day CDR with 150+ participants

- Secured Conditional "Go" decision from Air Force and Navy stakeholders with exception for weight and power non-compliance

- Limited post-review actions to 10 well-defined items

- Established clear path forward for power/weight compliance

Long-term Impact:

- Created reusable framework for future milestone reviews

- Strengthened relationship with government stakeholders

Metrics:

- Met all documentation submission deadlines

- Achieved 90%+ design readiness across domains (only except was weight and power)

- Maintained program schedule despite technical challenges

- Created efficient action tracking and closure process

STORY 2: AUTOMOTIVE SEARCH APP DELIVERY

Problem Statement:

Deliver Amazon's first automotive Search 1P application integrating ASV, Alexa Local Info, and Vega capabilities across multiple VP organizations. Critical challenge: Continue development despite pending customer data partner decision to maintain program timeline.

Action:

1. Scope Definition and Architecture Planning

- Led architecture workshop to define initial scope and technical requirements

- Established scalable framework focusing on Navigation-Search application integration

- Prioritized backend enablement as foundation for future iterations

2. Cross-Team Alignment and Dependencies

Alexa Local Info Integration:

- Conducted deep dives to identify critical cloud API dependencies

- Aligned on two core capabilities with product to begin work:

  - POI (Points of Interest) Search
  - Auto-suggest functionality

Partner dependencies:

- Identified three key workstreams as partner dependencies to align and get commit:

  - Keyboard implementation
  - IPC mechanism
  - Inter-app communication protocol

3. Strategic Influence and Prioritization

Built compelling case for partner teams by:

- Highlighting strategic importance of penetrating automotive market

- Demonstrating value of functional prototype over strategy discussions

- Emphasizing broader impact on Quattro program milestones

  - Quantifying blast radius of enabler components

4. Execution

  - Developed simplified first iteration of automotive keyboard (en_US) by away team agreement

  - Implemented core features while maintaining flexibility for future partner integration

Outcome:

Successfully delivered first 1P Search application demonstration to customer, showcasing Amazon's Automotive search capabilities while:

- Maintaining development momentum despite data partner uncertainty

- Creating reusable components for future automotive applications

- Delivering MVP within resource constraints

Leadership Principles Demonstrated:

- Dive Deep

- Invent and Simplify

- Bias for Action

- Deliver Results

- Frugality

Teams Involved across 2 VP orgs:

- Quattro Search

- Quattro SCM

- Quattro launcher frameworks

- Quattro QA

- Vega UI framework

- Vega Script

- Vega Accessibility

- Local info (cloud)

STORY 3: L10 Localization Challenge

Problem Statement:

The Quattro team faces challenges in L10N localization due to their inexperience and lack of a plan. Key issues include:

1. No clear OP1 commitments for localization efforts

2. Unaligned cross-team dependencies

3. Absence of a testing strategy for locale-specific features

4. Insufficient QA capabilities for handling locale testing

Action:

To address these challenges, I took ownership of the situation and implemented the following steps:

1. Developed a localization plan, including:

   - String translation prioritization

   - Locale support roadmap for feature completion

   - Identification of testing gaps in QA

2. Initiated multi-pronged approach to secure testing resources:

   - Submitted intake request to AGQ for support

   - Proactively explored alternative options, anticipating potential AGQ delays

   - Leveraged LR mechanism to report Red status, creating urgency for additional solutions and decision making

3. Established partnerships with other orgs to fill testing gaps:

   - Reached out to other project teams to secure external vendor for L10 localization testing

   - Collaborated with DS2 localization team for parallel testing efforts

   - Implemented workaround with Quattro QA to provide English screenshots to localization team

4. Developed long-term scalability solutions:

   - Challenged Quattro QA to prioritize automation initiatives for Q3 2024

   - Justified automation need based on 25+ locale support requirement for customer

- Implemented efficient screenshot generation process using automation

5. Ensured continuous improvement and resource optimization:

  - Ongoing efforts to confirm 3P vendor onboarding post-tech SOP support (major milestone for program)

  - Enabled domain QAEs (they were already testing in English so this was few hours net new ask for them) to generate contextual screenshots without additional headcount

Outcome:

- Successfully established a cross-functional localization strategy and resolved QA testing gaps to meet tech SOP scope, demonstrating scalability and efficiency in resource utilization.
- Met scope for first launch and created mechanism for teams to self-manage localization process

Teams Involved:

1. Quattro domain teams (10+)

2. AGQ (Automated Global Quality)

3. DS2 localization team

4. Quattro QA team

5. 3P vendor (for post-tech SOP support)

Leadership Principles Demonstrated:

1. Ownership

2. Bias for Action

3. Invent and Simplify

4. Think Big

5. Deliver Results

6. Frugality

7. Learn and Be Curious

8. Dive Deep

STORY 4: Lack of automated Testing

Innovation Project: Automated System Testing Implementation for Transmission Control Systems

Situation:

As a software development manager working on a next-generation double-clutch transmission controller program, I identified a critical bottleneck in our development cycle. The system testing phase required 30+ days of manual testing at the end of each development cycle, creating delays in bug detection and resolution. This led to quality issues and customer dissatisfaction during integration phases. First, I started off with creating criteria a success criterion which is listed below:

Success Criteria: Develop and implement an automated testing solution to:

- Reduce system testing time from 30+ days to under a week

- Allow developers to perform system-level testing during development

Action:

Recognizing the need for automated testing, I developed a problem statement and presented it to my manager. After thoughtful discussions, my manager suggested submitting the idea to our Customer Innovation (CI) board- an internal body tasked with evaluating and funding innovative employee proposals.

I approached this opportunity methodically, preparing a detailed business case that included not only the justification but also precise estimates of required resources and hours needed for an automation prototype. My proposal was strengthened by thorough research into various automation test equipment options, demonstrating to decision-makers that I had a well-conceived plan ready for immediate implementation.

The CI board presentation proved successful- my vision for automated testing resonated strongly with the board members, and they granted approval and funding to proceed with the initiative.

To ensure long-term success, I strategically partnered with a member of the test organization, anticipating the eventual transition of the prototype to their team. For the technical foundation, I selected the Dspace Hardware in the Loop (HIL) module, a choice driven by its robust input/output capabilities and Python programming support. This decision was particularly forward-thinking, as Python's gentle learning curve would facilitate quick onboarding of new team members. Looking ahead to summer, I collaborated with test managers to recruit three interns, building a dedicated team to accelerate the project's development.

With the groundwork laid, I began by procuring the necessary hardware while simultaneously mapping out the testing requirements. I conducted a thorough review of existing manual tests and customer interfaces, ensuring our automation framework would meet requirements. A key insight came from consulting our resident engineer stationed at the customer site. Their first-hand experience with integration challenges proved invaluable, helping me prioritize areas like the problematic piston misfire detection system.

Once the hardware arrived, my team and I quickly established a build environment for test scripts. we focused initially on developing Python scripts for processing cam and crankshaft signals- crucial components in determining fuel injection timing- and automated the misfire detection testing. This particular test had been a bottleneck, requiring two days of manual setup and execution by specialized system test team members. Software engineers had previously been unable to run these tests due to the complex system knowledge required.

The initial script implementation wasn't without challenges. During our first target run, we discovered incorrect signal readings, which we traced back to improper routing through the breakout box (BoB). However, once resolved, the impact was dramatic- the misfire detection test that previously took two days could now be completed in just 20 minutes with a single button press.

With the framework proven, I directed my intern team to expand our automation efforts to additional interfaces. Throughout the summer, we maintained close collaboration with subject matter experts to address questions and challenges. Our efforts yielded impressive results- achieving 15% requirements coverage through automation.

To maximize efficiency, I implemented nightly test runs, automating the process of testing each day's final build. I developed Python post-processing scripts to generate comprehensive reports detailing which requirements were met, failed, or not executed. This automated reporting system provided valuable insights each morning, enabling quick decisions on necessary adjustments and next steps.

This systematic approach to automation transformed our testing capabilities, enabling developers to conduct system-level testing early in the development cycle and improving our ability to detect and address issues before they reached the customer.

Results:

- Reduced system testing time from 30 days to under 1 week

- Achieved 15% requirements coverage via automation within first 3 months

- Enabled developers to perform system-level testing during development

- Received leadership commitment to form dedicated automation team

- Improved customer satisfaction through enhanced product quality

- Established framework became standard practice for future projects

Leadership Principles Demonstrated:

1. Invent and Simplify

- Created an innovative automated testing solution

- Simplified complex testing processes into automated scripts

- Reduced testing time from 30+ days to under a week

2. Think Big

- Developed a solution that became standard practice for future projects

- Envisioned long-term benefits beyond immediate problem

- Created scalable framework for future expansion

3. Bias for Action

- Proactively identified bottleneck and developed solution

- Initiated proposal to CI board without waiting

- Quickly established build environment once hardware arrived

4. Learn and Be Curious

- Researched various automation test equipment options

- Selected Python for its learning curve advantages

- Conducted thorough review of existing manual tests

5. Deliver Results

- Reduced testing time by 85%

- Achieved 15% requirements coverage in 3 months

- Improved customer satisfaction through enhanced quality

6. Customer Obsession

- Consulted with resident engineer at customer site

- Prioritized known customer pain points (misfire detection)

- Improved product quality for customer satisfaction

Teams Involved:

1. Development Team

- Software developers

- System test team

2. Test Organization

- Test managers

- Test automation partner

- System test specialists

3. Customer Innovation (CI) Board

- Decision-makers for funding approval

4. Customer Site Team

- Resident engineer

5. Intern Team

- Three summer interns

6. Subject Matter Experts (SMEs)

- For technical consultation and validation

Story 5: CI/CD pipeline development

Problem Statement (Lack of CI/CD)

Our customers (internal development teams) were experiencing friction with OS integration:

- OS changes deployed only every 8 weeks
- 60% regression rate in API functionality
- 3-5 day uplevel process across 10+ domains
- 1+ week turnaround time for bug fixes

Customer-Focused Solution

Working backwards from our customers' needs, I led the initiative to establish a CI/CD pipeline infra that would reduce integration friction and improve code quality.

Data-Driven Approach

Analyzed last 3 OS releases

- Documented 60% regression rate
- Quantified uplevel effort: 3-5 days × 10+ domains

Built cross-functional tiger team

- Partnered with Sr. SDM, Sr. SDE, SCM team, and QAM
- Aligned on customer pain points and success metrics
- Leveraged diverse perspectives for comprehensive solution

Success Criteria

- Increase deployment frequency: 8 weeks → 2x/week
- Eliminate manual uplevel process
- Reduce regressions: 60% → <10% within 6 months

Technical Leadership

Drove end-to-end solution development (see diagram below):

1. Authored one-pager proposing daily integration model

2. Led white-boarding sessions with tiger team

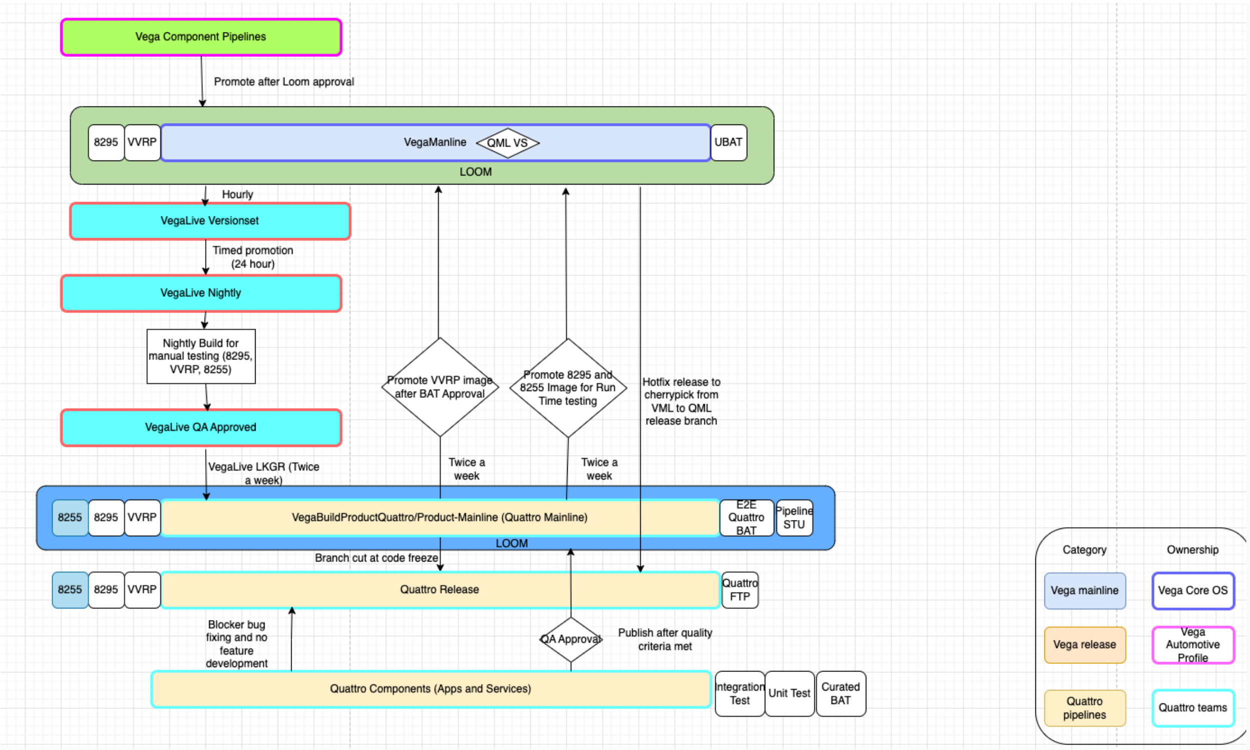
3. Developed HLD through 10+ iterations

4. Achieved alignment across 5+ teams spanning 2 L10 organizations

5. Created and executed workback plan with clear Go-live dates

6. Implemented automated testing and validation gates

7. Established clear escalation paths and SLAs



Vega Component Pipelines
Promote after Loom approval
8295
VVRP
VegaManline
QML VS
UBAT
LOOM
Hourly
VegaLive Versionset
Timed promotion (24 hour)
VegaLive Nightly
Nightly Build for manual testing (8295, VVRP, 8255)
VegaLive QA Approved
VegaLive LKGR (Twice a week)
Promote VVRP image after BAT Approval
Promote 8295 and 8255 Image for Run Time testing
Hotfix release to cherrypick from VML to QML release branch
Twice a week
Twice a week
8255
8295
VVRP
VegaBuildProductQuattro/Product-Mainline (Quattro Mainline)
E2E Quattro BAT
Pipeline STU
LOOM
Branch cut at code freeze
8255
8295
VVRP
Quattro Release
Quattro FTP
Blocker bug fixing and no feature development
QA Approval
Publish after quality criteria met
Quattro Components (Apps and Services)
Integration Test
Unit Test
Curated BAT
Category
Ownership
Vega mainline
Vega Core OS
Vega release
Vega Automotive Profile
Quattro pipelines
Quattro teams

Mechanisms

Created scalable processes:

- Daily triage mechanism for pipeline failures
- Clear ownership and accountability model
- SLA tracking and enforcement
- Automated regression detection
- Streamlined revert/fix decision framework

Results and Customer Impact

- Speed: Successfully transitioned from 8-week to 2x/week OS integration
- Quality: Reduced regression rate from 60% to <30% within 2 months
- Efficiency: Eliminated manual uplevel process
- Scale: Established foundation for automated testing expansion

Next Steps (In progress today)

Currently working on:

- Expanding automated test coverage
- Further reducing regression rate to meet <10% target
- Scaling triage mechanism

This initiative demonstrates strong ownership, customer obsession, and bias for action, resulting in significant improvements to our development velocity and code quality.

Story 6: Tradeoff example

Situation (Example of Trade off):

Product unit exceeded critical weight and power specifications, with initial design waiver request denied by customer. This threatened program timeline and deliverables while potentially impacting future feature capabilities.

Task:

Develop and implement solution to bring unit within specified weight and power requirements while preserving future expandability.

Action:

1. Cross-Functional Team Assembly

- Formed tiger team comprising systems, hardware, software, program, and business experts

- Ensured representation from all critical technical domains

- Established clear decision-making framework

2. Root Cause Analysis

- Conducted deep dive into power and weight constraints

- Identified Circuit Card Assembly (CCA) designated for future growth as primary contributor

- Mapped dependencies and impacts of potential CCA removal

3. Solution Development

- Engaged SMEs for comprehensive impact assessment across domains

- Developed design modification proposal removing targeted CCA

- Validated new design met power and weight requirements

- Created technical specification for modified configuration

4. Future-Proofing Implementation

- Maintained empty slot in design for future expansion

- Preserved capability to add CCA for customers without strict power/weight constraints

- Documented design decisions and rationale for future reference

Result:

1. Immediate Wins:
   - Achieved compliance with weight and power specifications
   - Eliminated need for design waiver
   - Maintained program timeline
2. Strategic Benefits:
   - Preserved future growth capability through clever design
   - Created flexible solution adaptable to varying customer requirements
   - Balanced immediate needs with long-term product strategy

Leadership Principles Demonstrated:

- Invent and Simplify
- Bias for Action
- Think Big
- Deliver Results

STORY 7: Trade off Example

Situation:

Clinical team required a blinded patient solution, presenting opportunity to either develop quick tactical solution or strategic long-term implementation through upcoming Simplera platform.

Task:

Evaluate requirements and determine optimal implementation strategy balancing immediate clinical needs with long-term platform scalability.

Action:

1. Requirements Analysis

- Conducted deep dive with clinical team to understand specific blinding needs
- Mapped current study timeline against product roadmap
- Identified dependencies with upcoming sensor app launch

2. Strategic Assessment

- Developed detailed implementation plans for both tactical and strategic approaches
- Analyzed resource requirements and impact on existing projects
- Quantified maintenance costs for multiple solutions vs. unified platform

3. Trade-off Analysis

- Presented recommendation to VP of Engineering to integrate blinding feature into Simplera
- Key benefits identified:
  * Single codebase maintenance
  * Future study readiness
  * Long-term resource optimization
- Acknowledged short-term impact:
  * Temporary market share decrease
  * Need to pause similar-priority project
  * Two-month gap in solution availability

4. Risk Mitigation

  - Developed transition strategy to manage period until Simplera launch

  - Created communication plan for stakeholders

  - Established metrics to track impact of temporary solution gap

Result:

1. Strategic Wins:

  - Secured executive buy-in for long-term solution

  - Positioned product for future scalability

  - Reduced technical debt through unified platform approach

2. Investment in Future:

  - Accepted calculated short-term market share impact

  - Prioritized long-term platform sustainability

  - Created foundation for efficient future study support

Leadership Principles Demonstrated:

- Think Big

- Bias for Action

- Are Right, A Lot

- Deliver Results

- Frugality (through long-term resource optimization)

STORY 8: Opportunity to re-risk

Situation:

Program facing escalating costs and increasing technical complexity, threatening overall project viability and long-term sustainability.

Task:

Identify and implement strategic cost reduction measures while simplifying program complexity to enhance efficiency and scalability.

Action:

1. Analysis
   - Conducted review of program components
   - Identified GHS RTOS implementation and certification as high-impact area
2. Technical Deep Dive
   - Analyzed current processor architecture:
     * Two processors from same family
     * Third processor from different family
   - Recognized inefficiencies in RTOS development and certification process
3. Cost Impact Assessment
   - Quantified potential savings:
     * GHS licensing fees
     * Internal labor costs
     * Cumulative impact: $2M+

4. Cross-Functional Collaboration
   - Convened key stakeholders:
     * Hardware designers
     * Software engineers
     * System architects

- Facilitated focused discussion on design communization opportunity

5. Data-Driven Proposal Development
   - Created concise, impactful one-chart summary
   - Highlighted technical benefits and cost savings
   - Secured buy-in from technical stakeholders

6. Executive Alignment
   - Presented cost-saving initiative to business team
   - Leveraged technical stakeholder support to strengthen proposal

Result:

1. Financial Impact:
   - Secured approval for cost-saving measures
   - Achieved $2M+ reduction in program expenses
2. Technical Simplification:
   - Standardized processor architecture across all units
   - Reduced software complexity through unified driver development
3. Long-Term Efficiency Gains:
   - Enabled design communization for future iterations
   - Streamlined RTOS implementation and certification process
   - Improved procurement leverage through higher volume of standardized components
4. Strategic Positioning:
   - Enhanced program competitiveness through reduced costs
   - Improved scalability for future product iterations
   - Demonstrated commitment to continuous improvement and cost optimization

Leadership Principles Demonstrated:

- Frugality
- Invent and Simplify
- Dive Deep
- Deliver Results
- Think Big

STORY 9: Left shift schedule and maintain high quality bar

Situation:

Program experiencing significant schedule delays and deteriorating customer satisfaction, requiring immediate intervention to maintain business relationship and program viability.

Task:

Accelerate program timeline and enhance customer engagement while maintaining quality standards and deliverables.

Action:

1. Technical Assessment
   - Conducted deep dive into program execution bottlenecks
   - Identified qualification testing as critical path:
     * Factory Qualification Testing (FQT)
     * Electromagnetic Interference (EMI)
     * Environmental Qualification Testing (EQT)
     * Highly Accelerated Life Testing (HALT)
   - Mapped sequential testing dependencies and parallelization opportunities
2. Resource Optimization
   - Escalated to leadership for additional test units
   - Built business case for parallel testing approach
   - Secured necessary resources through stakeholder alignment
3. Risk Management
   - Developed risk assessment for parallel testing if there are failures
   - Created customer-facing justification for modified test approach
4. Customer Engagement Enhancement
   - Implemented informal review cycles for early feedback
   - Reduced formal review cycles through proactive engagement
   - Increased transparency in decision-making process

5. Cross-Program Collaboration

  - Identified shared laboratory assets across programs

  - Negotiated access to additional test facilities

  - Optimized resource utilization across multiple programs

Result:

1. Schedule Impact:

  - Achieved 5-month acceleration in program timeline

  - Eliminated sequential testing bottlenecks

  - Optimized resource utilization across test phases

2. Customer Relationship:

  - Improved customer satisfaction through increased engagement

  - Reduced review cycle times

  - Enhanced transparency and communication

3. Operational Efficiency:

  - Established parallel testing capabilities

  - Maximized equipment utilization

  - Created repeatable process for future programs

Leadership Principles Demonstrated:

- Customer Obsession

- Bias for Action

- Deliver Results

- Earn Trust

- Think Big

STORY 10: Strategic Pipeline Infrastructure Pivot

Situation:

While in the final stages of implementing a planned pipeline design, our team discovered a new tool that offered advantages- faster build times and simplified maintenance. This presented a critical decision point between continuing with our nearly-completed implementation or pivoting to the new solution.

Action:

- Conducted thorough analysis of the new tooling to understand its benefits and implementation requirements

- Initially advocated for completing current implementation and incorporating new tool in next update to avoid delays

- After team discussion and data analysis, recognized the long-term benefits outweighed short-term timeline impact. I disagree and committed as this would delay our POR by 2 months and teams would need to work with current pain points for another 2 months.

- After disagree and commit, presented unified recommendation to leadership highlighting:

  * Expected 2-month delay in delivery

  * Long-term operational benefits

  * Maintenance cost reduction

  * Improved build performance

- Secured leadership buy-in for the strategic pivot

Results:

- Developed accelerated implementation plan to minimize delay

- Created focused onboarding campaign with dedicated working sessions with domain teams

- Reduced expected delay from 8 weeks to 6 weeks through optimized execution

- Successfully deployed new infrastructure tooling that delivered:

  * 90-minute reduction in build times

  * Simplified SCM maintenance processes

  * More sustainable long-term solution

Leadership Principles Demonstrated: disagree and commit