CONFIDENTIAL WORK SAMPLE

Product Management Case Study

Prepared for: [redacted]

Prepared by: Jason R Griffin

Date: November 22, 2025

Contact: [redacted]

INTELLECTUAL PROPERTY NOTICE

This document is my original work and intellectual property. I'm sharing it with you as an example of my product management approach and thinking as part of my application.

A few important notes:

- I created this to demonstrate how I approach product problems and/or challenges and the build strategy to address those challenges
- Everything in here—the analysis, frameworks, and recommendations—is my own work
- This is confidential and meant only for your review during the hiring process
- Please don't share, copy, or use any part of this outside of evaluating my candidacy
- By reading this, you're agreeing to keep it confidential and use it only to assess my fit for the role

---

DISCLAIMER

If I reference actual companies, products, or market examples in this case study, the strategies and analysis are entirely my own perspective. Nothing here reflects internal information or views from any employer I've worked with.

Part 1: Defining the Strategy

Discovery

The path to re-platforming [redacted] PLM/catalog systems and repositories starts with understanding the current architecture and pain points. This discovery exercise will consist of three main activities: system architecture audit, stakeholder interviews and pain point analysis.

System architecture audit. As part of this audit, we would map all the current PLM/catalog systems and repositories, including the hidden tech such as user-created macros, offline automation processes, etc. A review of existing documentation would contribute to this audit. We would generate data flow diagrams showing how items move from inception to delivery through the various platforms and identifying integration points and mechanisms, as well as any batch or offline processes.

Stakeholder Interviews. We would arrange for and conduct sessions with the various stakeholders. These stakeholders are teams including, but not limited to, product development (designers), go-to-market (marketing, merchandising), operations (supply chain, manufacturing, quality control), IT (system owners, data architects) and of course the customer service teams dealing with customer quality issues. These interviews will focus on their "day in the life", gathering pain points, and seeking out their "hacks" (creative solutions?) for the existing systems. One very important objective of these sessions would be determining what, if any, areas are "untouchable". These are cultural foundations in the company (not organization or team) that cannot be changed, like Costco's $1.50 hot dog or Nordstrom's return policy.

Pain Point Analysis. After conducting the first two exercises, we should have a comprehensive list of what the pain points are with the current system and processes. We should be able to quantify impacts, document error rates, identify bottlenecks, and measure data inconsistencies across the different systems and processes.

Following these sessions, we should be able to answer these three key questions:

- From where does the data originate, and how many sources of "truth" are there today
- What are the most frequent complaints in the process and data quality
- Which step or category causes the most issues or pain

Design

Having concluded the discovery phase and gained alignment to move forward, we then move to design the go-forward plan. For this, I would recommend a federated domain model. The following assumptions are being made:

- Independent parts of the business want to maintain their domain autonomy
- Cross-functional governance model is feasible given organizational structures and culture
    - Validated through stakeholder interviews

- Current systems can support API-based integrations
    - Validated through technical feasibility reviews with engineering teams

The platform would have the following domains

1. **Product Identity Domain** (Centralized)
    - SKUs, style numbers, GTINs
    - Taxonomy and categorization
    - Master hierarchy
2. **Product Design Domain** (Federated by category)
    - Design specs and construction
    - Materials, fabrics, trims
    - Color and print details
    - Sizing and fit specifications
    - Sustainability and compliance
3. **Merchandising & Planning Domain**
    - Collections and seasons
    - Assortment planning
    - Channel strategy
    - Pricing and margin
4. **Digital & Content Domain**
    - Product imagery
    - Marketing copy
    - Digital assets
    - Size guides
5. **Lifecycle Management Domain**
    - Product status and versions
    - Change history
    - Variant relationships
6. **Supply Chain Domain**
    - Sourcing and manufacturing
    - Vendor management
    - Packaging and logistics
7. **Customer Experience Domain**
    - Fit and sizing data
    - Style recommendations
    - Reviews and returns
    - Personalization

This structure allows specialized teams to own domain-specific attributes while maintaining centralized control over cross-cutting concerns like identity and lifecycle.

Strategy

Once the structure is aligned, I would assess each system/domain against:

- Data quality issues and manual workaround costs
- Integration complexity and maintenance burden
- Scalability limitations for future growth
- Strategic importance to customer experience
- Technical debt and modernization urgency

This enables a phased deployment approach.

Phase 1: Foundation - Product Identity Layer

- Implement a Master Data Management (MDM) hub for golden records
- Establish universal identifiers and taxonomy
- Create API layer for product identity services
- **Why first:** Everything depends on consistent product identification

Phase 2: High-Impact Domain - Commercial/GTM

- Re-platform merchandising and marketing content management
- Implement Product Information Management (PIM) for customer-facing data
- **Why second:** Direct impact on revenue and customer experience

Phase 3: Technical Specifications

- Modernize engineering/technical attribute management
- Integrate with PLM systems
- **Why third:** Complex domain requiring deep product expertise

Phase 4: Supply Chain Integration

- Connect re-platformed core to supply chain systems
- Implement automated data flows
- **Why last:** Dependent on stable upstream data foundations

Handling Duplicate Records & Conflicting Attributes

There will be duplicate records across the different systems/domains so it will be imperative that there be a triage strategy.

1. **Automated Detection & Flagging**

- Implement fuzzy matching algorithms for duplicate detection
- Create data quality dashboards showing conflicts by severity
- Establish automated alerts for critical discrepancies

2. **Resolution Framework:**

   For Duplicates:

   - **Survivorship Rules:** Define which system is authoritative for each attribute type
     - Example: Marketing copy → PIM system wins
   - Create merge workflows with human-in-the-loop for edge cases
   - Establish "golden record" concept in MDM hub

   For Conflicting Attributes:

   - **Governance Model:**
     - Ownership assignment: Each attribute has a designated owner/steward
     - Data lineage tracking: System of record designation
     - Change control process for critical attributes

   For Lifecycle Rules:

   - Implement state mechanism for product lifecycle
   - Define valid transitions (e.g., Development → Active → Discontinued/Archived)
   - Establish versioning strategy (major/minor versions for different change types)
   - Create master/variant relationships with inheritance rules
   - Document approval workflows for lifecycle transitions

3. **Data Cleansing Sprints**
   - Prioritize by business impact (customer-facing first)
   - Assign cross-functional teams to resolve domain-specific conflicts
   - Build institutional knowledge base of resolution patterns

OKRs

When defining the OKRs for this effort I will use a value vs effort matrix combined with a dependency map. For these OKRs, I am making the assumption that data quality issues cause measurable business impact via delays, errors or inflated costs. Percentages supplied are not based on real data.

1. Establish Single Source of Truth for Product Identity

- **Why prioritize:** Foundation for everything else; prevents duplicate work
- **Key Results:**
    - 100% of active SKUs have unique, validated identifiers in MDM hub
    - Reduce duplicate product records by 90%
    - Achieve 99% accuracy in product taxonomy classification
- **Success validation:** Audit sample of 500 products across categories; measure downstream system synchronization accuracy

2. Reduce Time-to-Market for New Products by 25%

- **Why prioritize:** Direct business value; addresses competitive pressure
- **Key Results:**
    - Cut product data creation cycle from 6 weeks to 3.5 weeks
    - Automate 80% of data propagation across systems
    - Reduce data-related product launch delays by 75%
- **Success validation:** Track actual launch timelines; measure % of launches meeting target dates; stakeholder satisfaction surveys

3. Improve Customer-Facing Data Quality to 98% Accuracy

- **Why prioritize:** Direct customer impact; revenue protection
- **Key Results:**
    - Reduce product information-related returns by 50%
    - Achieve 98% accuracy in critical attributes (dimensions, materials, compatibility)
    - Eliminate customer-reported product data errors by 85%
- **Success validation:** Mystery shopping audits; customer complaint analysis; A/B testing impact on conversion rates

Deprioritized OKRs (and rational)

- Harmonization OKRs: Important but dependent on establishing identity first
- Lifecycle rules: Valuable but can be iterative enhancement after foundation
- Stewardship models: Necessary but can be developed in parallel with technical work

Part 2: Prioritizing Initiatives

System	Annual Maintenance Cost	Avg Time to launch new product (days)	% of Product data errors	Estimated replatforming cost	Estimated Annual savings post-replatform	Strategic Importance (1= least important - 5 = most important)
A	$1.2M	45	12%	$3.5M	$1.8M	5
B	$800K	30	5%	$2.2M	$900K	3
C	$1.5M	60	18%	$4.0M	$2.5M	4
D	$600K	25	3%	$1.5M	$500K	2

Using the above data points, I have made the following assumptions:

- Average Time to launch new products refers to customer facing products, not the platform itself
- The company would prioritize ROI over speed to market of the new systems
- Other factors like user or customer sentiment are not material in this prioritization.
- The budget for re-platforming is sufficient to staff the team(s) so that delivery of the systems are relatively similar

System	Break Even period (years)	Products per year	3 year RIO (in M)	5 year ROI (in M)
A	1.9	8.1	5.4	9.0
B	2.4	12.2	2.7	4.5
C	1.6	6.1	7.5	12.5
D	3.0	14.6	1.5	2.5

Given the above breakdown of systems and costs, I would prioritize the system re-platforming in the following order:

1. C
2. A
3. B
4. D

While system C is the most expensive to re-platform, it is also the one that gives the quickest breakeven point, greatest ROI in 3 and 5 years, as well as accounting for the

largest percentage of errors. Even though the other systems arguably could produce more products per year, without understanding more about those products and the overall benefit to the company's bottom line, the prioritization above offers the best balance of system cleansing and responsible budgeting.