TECHNICAL PRODUCT LEADERSHIP CASE STUDY

Modernizing a Complex Customer Journey Platform

An anonymized case study in product judgment, platform strategy, user experience, and cross-functional execution.

Prepared by Mandie Morgan | Technical Product Leader - Enterprise Platforms and Customer Experience

Executive overview. I led an early product proof for a new enterprise transaction platform, then carried the direction into implementation. I aligned strategy, architecture, design, requirements, engineering, vendor, and executive partners around product boundaries, sequencing, and delivery readiness. The work cut design-brief turnaround from about six weeks to 2.5-3 weeks, reduced related configuration defects from 2-5 per week to 2-5 total over the next three months, and created a reusable cross-product planning model. I also built Rovia, a site that connects requirements to code, and an approval-gated automation pipeline that updates affected artifacts when changes occur and creates a merge request only after human approval.

Business and technical context

The company was replacing a legacy core platform that supported high-volume customer acquisition and post-purchase servicing. This was not a simple system swap. Teams needed shared API ownership, messaging patterns, platform boundaries, and product direction across internal systems and vendor technology.

Frontline employees also encountered inconsistent workflows across customer journeys. Similar eligibility-review and follow-up-data tasks were designed separately, increasing cognitive load and duplicating product and engineering effort. The work required one product view that connected user feedback to architecture and delivery.

Role progression

Phase	My mandate	Evidence and outcome
Aug-Dec 2025 Proof of concept	Led early product validation; partnered with enterprise strategy, served as a delegate for product leadership, and translated direction for stakeholders.	Set scope and roadmap, partnered on messaging and platform-boundary decisions, and presented progress to executives.
Jan 2026 onward Implementation	Product lead for the customer-acquisition experience within an enterprise platform modernization.	Sequence work, make decisions and dependencies visible, guide reusable workflow concepts, and raise design and delivery readiness.

Product decisions and technical judgment

Two decisions show how I connect user needs to platform ownership, technical boundaries, and buildable product direction.

Decision 1 | Define platform boundaries before scaling delivery

Situation. The new platform needed early messaging and API decisions before teams could scale implementation. Ownership across the core transaction platform and customer-facing applications was not yet clear.

Technical judgment. I partnered with a Staff Engineer to evaluate messaging frameworks and trace user, application, and system responsibilities across the integration.

Direction. I proposed API ownership and platform boundaries that separated responsibilities without fragmenting the end-to-end customer journey.

Outcome. The work gave later delivery a clearer technical foundation and a shared reference point for scope, sequencing, and tradeoff decisions.

Boundary principle. Define which product owns each decision and data contract before asking delivery teams to optimize locally.

Decision 2 | Design one reusable workflow instead of duplicating experiences

Gap identified. Two customer-facing product areas handled comparable eligibility reviews and follow-up data differently. Rebuilding each workflow separately would preserve inconsistency and technical debt.

Product concept. I proposed a shared decision-support workflow across product areas and a configurable follow-up-question pattern that let frontline users correct responses when new information triggered additional review.

Product direction. Frontline-user feedback shaped both concepts, connecting customer and operational needs to a more coherent cross-product direction and clearer future scope.

Execution mechanisms I owned

Sequence	Order work by priority, prerequisites, dependencies, and complexity, with current risks and decisions visible.
Model	Map end-to-end user scenarios into Mermaid flows and design briefs that connect product intent to experience and implementation.
Trace	Built Rovia to connect requirements with implementation code, plus an approval-gated pipeline that updates affected artifacts and creates a merge request after human approval.
Communicate	Maintain a shared cross-product planning hub for milestones, blockers, schedule decisions, weekly changes, and executive context.

Improving delivery readiness and quality

Two recurring delays had the same underlying issue: teams were being asked to act before scenarios, configuration expectations, and affected artifacts were visible enough. I treated readiness as a product problem and strengthened the operating path from business intent to design and delivery.

System view. Faster execution came from making prerequisites and evidence visible earlier, not from asking teams to work faster.

Readiness and quality mechanisms

Scenario mapping	Converted business-use scenarios into Mermaid flows to expose state-transition paths, decision points, and missing assumptions.
Design briefs	Created concise briefs that aligned product, design, and engineering expectations before build decisions.
Definition of Done	Added configuration and evidence checks so vendor work arrived with clearer requirements and acceptance expectations.

Turn a recurring traceability problem into a product

I built a shared planning hub across two customer-facing product areas, bringing milestones, blockers, schedule decisions, and weekly changes into one view. The deeper issue was traceability between intent and implementation. I built Rovia, a site that connects requirements to code and helps teams investigate whether drift comes from a stale requirement or incorrectly implemented code. I also built an approval-gated automation pipeline that updates affected artifacts when changes occur and creates a merge request only after the proposed updates are approved.

Measured change and attribution

Signal	Before	After the changes
Design-brief turnaround	About 6 weeks	2.5-3 weeks
Related configuration defects	2-5 weekly	2-5 total over the following three months
Decision visibility	Cross-product updates spread across sources	Shared hub for milestones, blockers, schedule decisions, and weekly changes

The defect comparison reflects observed issue volume, not a normalized defect rate. It followed strengthened scenario, readiness, and Definition of Done practices and is not attributed to one action alone.

Scale forward

With one product area operating on a clearer technical and planning foundation, I shared the method for potential reuse across adjacent teams. I am also partnering on a way to identify requirement and implementation gaps across planned end-to-end customer scenarios and the initial product proof, extending the same emphasis on traceability, human approval, and reusable product direction.

Leadership pattern. See the whole system, make ownership and tradeoffs visible, connect user feedback to platform decisions, and turn direction into reliable delivery.