

John Whitman

Mortgage product leadership

Build mortgage products Moder can sell and deploy repeatedly

I built two product organizations. I bring consumer-lending experience and hands-on Foundry work.

I'd lead product decisions from field problem to working product: buyer, scope, operator adoption, packaging, pricing and whether to stop or scale. **Product owns what; Architecture and Build determine how.**

Get the agreed repayment plan recorded correctly

PROPOSED DEMONSTRATION / FICTIONAL CASE / NO LIVE CONNECTION

Buyer to validate: Head of default servicing • **User:** Mortgage servicing specialist

A homeowner has agreed to an approved catch-up plan. [2] The proposed product helps the specialist record it correctly, resolve exceptions and verify the result.

AI assembles evidence; the specialist authorizes an exact first plan-payment date correction. Fresh readback must verify that field on the intended record; otherwise work stays open with an owner. This simulated field result does not establish borrower or business outcomes. Specialists retain assistance decisions.

Choose what is worth building

Validate frequency, current cost and willingness to pay. Test existing software and simple rules first. Measure actual operator adoption, correction quality and all-in cost across every assigned case, including failures and fallback. Stop if simpler tools already solve the problem economically.

Test a packaged servicing offer

Test the offer with the head of default servicing: approved-plan recording, human-authorized correction and verified results. Reuse supported connectors, configuration, QA and training across customers. Measure whether this reduces deployment and support effort, leaves buyer value and covers Moder's full costs. Demand and improvement remain hypotheses.

Build with the existing team

Moder's Palantir partnership is established; this workflow is proposed. [1] Foundry/Ontology and AIP could support records and operator work. AI FDE/Pilot could accelerate reviewed builds where available. Architecture verifies integrations and entitlements. [3-6]

Prove the product with a second customer

NOT YET TESTED

BUY

A second customer pays for the bounded product.

USE

Its specialists use it in their daily work.

DEPLOY

A different team deploys it using configuration.

Same product version. Documented configuration. Same outcome, quality and economics tests.

Let's discuss the first product and the role.

Sources: [1] Partnership [2] Repayment-plan context [3] Ontology [4] AIP [5] AI FDE [6] Pilot

Experience is candidate-reported. Demand, deployment, savings and repeatability are not established.