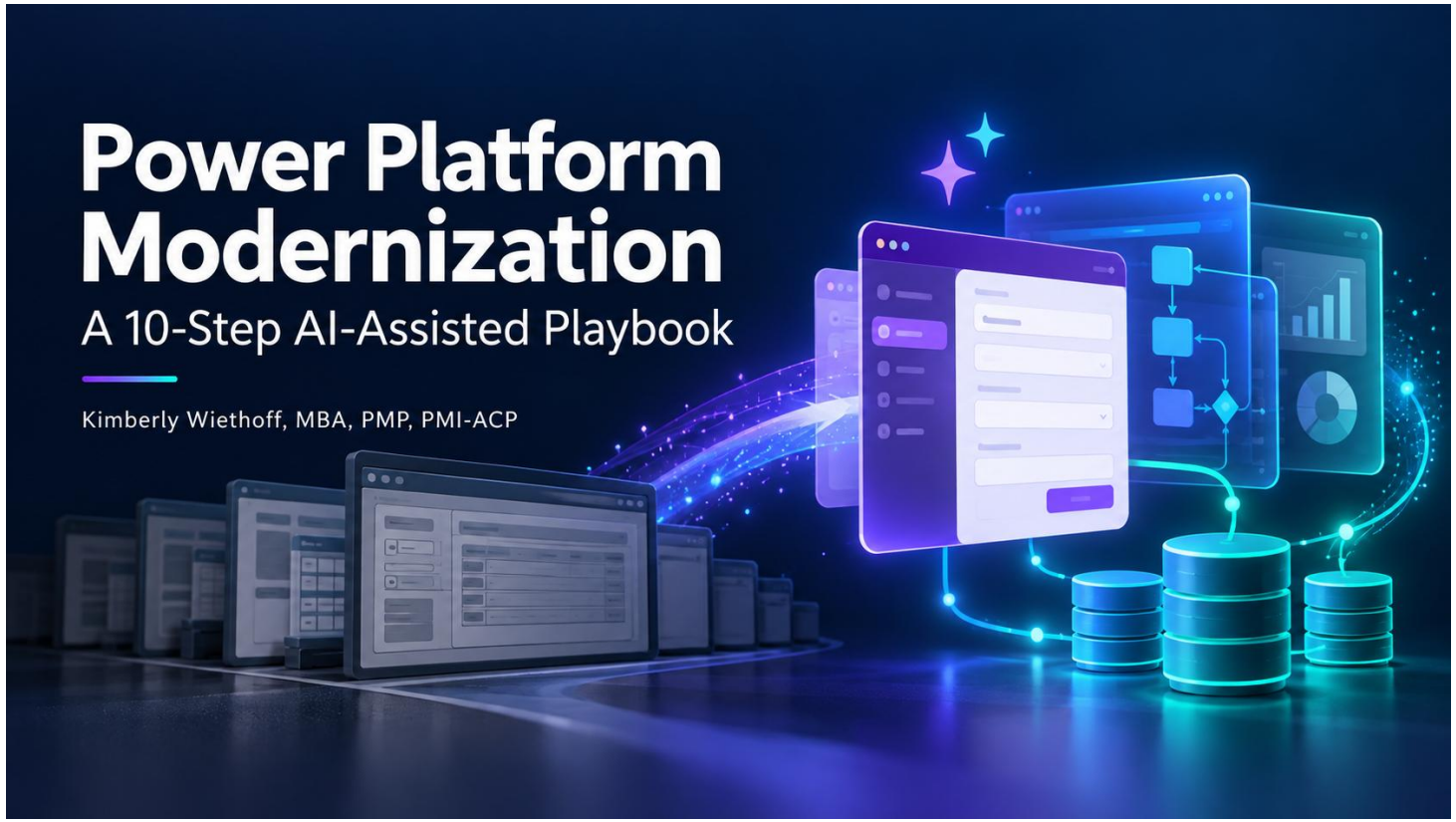
Power Platform Modernization: A 10-Step AI-Assisted Playbook

A structured, end-to-end guide for enterprise teams migrating legacy applications to Microsoft Power Platform — with clear deliverables, acceptance criteria, and practical AI integration points at every phase.

By Kimberly Wiethoff, MBA, PMP, PMI-ACP

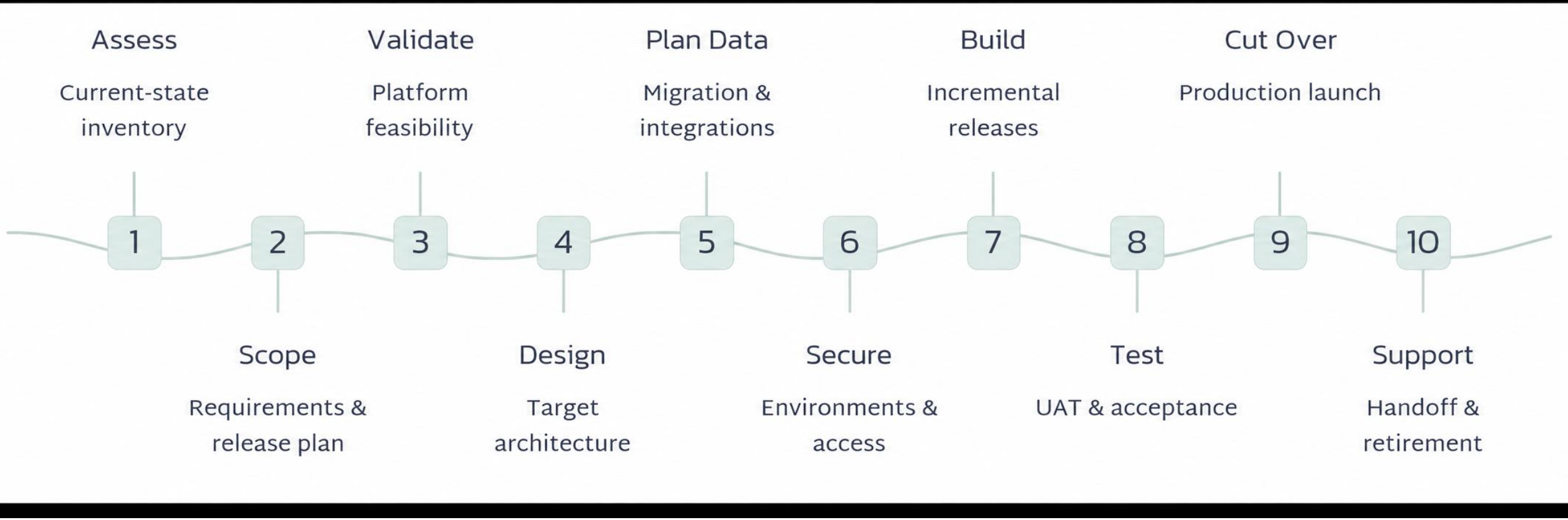
ENTERPRISE IT PLAYBOOK

AI-ASSISTED DELIVERY



The Modernization Journey at a Glance

Ten structured phases carry your team from initial discovery through production launch and legacy retirement. Each phase produces a concrete deliverable and carries measurable acceptance criteria — so progress is always visible and handoffs are unambiguous.



PHASE 1

Assess the Current Application

What Needs to Happen

Inventory every screen, workflow, business rule, report, integration, data source, and user role. Identify known pain points, technical debt, and the business and technical owners accountable for each area.

Key Deliverable

Current-state assessment document, validated by business and technical owners.

Where AI Can Help

- Summarize existing documentation and legacy code at speed
- Draft process maps from narrative descriptions
- Extract embedded business rules from screen logic or specs
- Surface gaps and generate targeted stakeholder questions

Acceptance Criteria

- All in-scope screens, workflows, rules, reports, integrations, and data sources documented
- User roles and application dependencies identified
- Business and technical owners have validated the assessment

PHASE 2

Define Scope and Success Measures

What Needs to Happen

Decide which capabilities to retain, improve, or retire. Define the first release scope and agree on measurable outcomes with baselines and targets so success is objectively verifiable at go-live.

Where AI Can Help

- Convert stakeholder notes into structured user stories and acceptance criteria
- Flag duplicate, conflicting, or ambiguous requirements before they cause rework
- Identify out-of-scope items and suggest explicit exclusions

Key Deliverable

Approved requirements backlog and first-release scope document.

Acceptance Criteria

- Every in-scope requirement has a unique ID, priority, owner, and testable acceptance criterion
- First-release scope and exclusions are approved by stakeholders
- Success measures have documented baselines and quantified targets

Use the **Given / When / Then** format for individual requirements — it forces precision and makes test-case generation straightforward.

PHASE 2 · DEEP DIVE

Writing Requirements in Given / When / Then

A consistent requirement format eliminates ambiguity, speeds review cycles, and makes AI-assisted test-case generation far more reliable. Every testable condition is explicit from the start.

> **Given** an authorized approver has a request awaiting review,
> **When** they approve the request,
> **Then** its status changes to "Approved," the approver and timestamp are recorded, and the requester receives a notification within [X minutes].

Given

Establishes the precondition — who is involved and what state the system is in before the action.

When

Describes the specific action or event that the user or system performs.

Then

Defines the expected, verifiable outcomes — including data changes, notifications, and timestamps.

PHASE 3

Confirm Power Apps Suitability

Before committing to Power Platform, validate it against your real-world constraints. Surface risks early — when they are cheap to resolve — rather than mid-build, when rework is costly.

Performance & Volume

Confirm that expected data volumes, concurrent users, and response-time targets are achievable within platform limits.

Security & Licensing

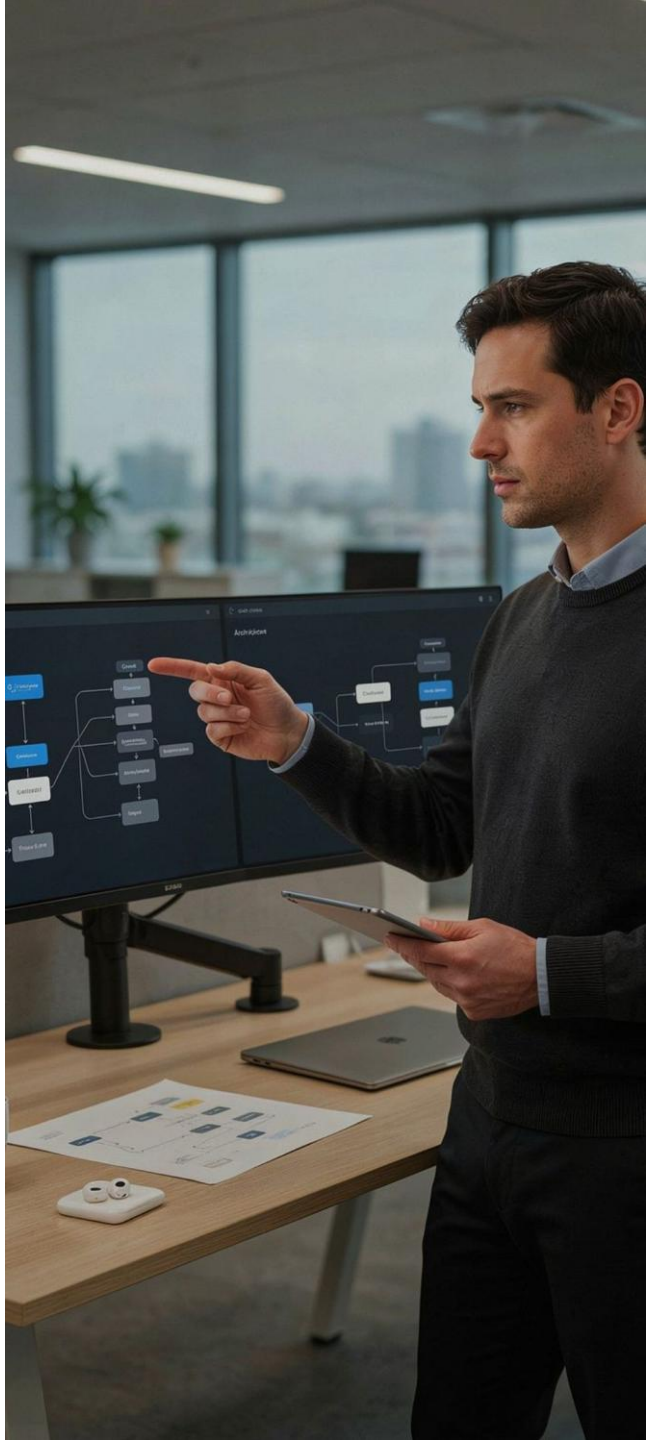
Validate connector policies, data-loss prevention rules, and per-user licensing costs against current Microsoft guidance.

Integrations & Offline

Assess connectivity requirements, offline scenarios, and API availability for all required external systems.

Proof of Concept

Build a targeted prototype covering the highest-risk business scenarios using representative data volumes.



PHASE 4

Select the Application Architecture

What Needs to Happen

Choose canvas app, model-driven app, or a hybrid combination. Determine whether Power Pages is needed for external users and define the Power Automate automation strategy for approvals, notifications, and scheduled processes.

Key Deliverable

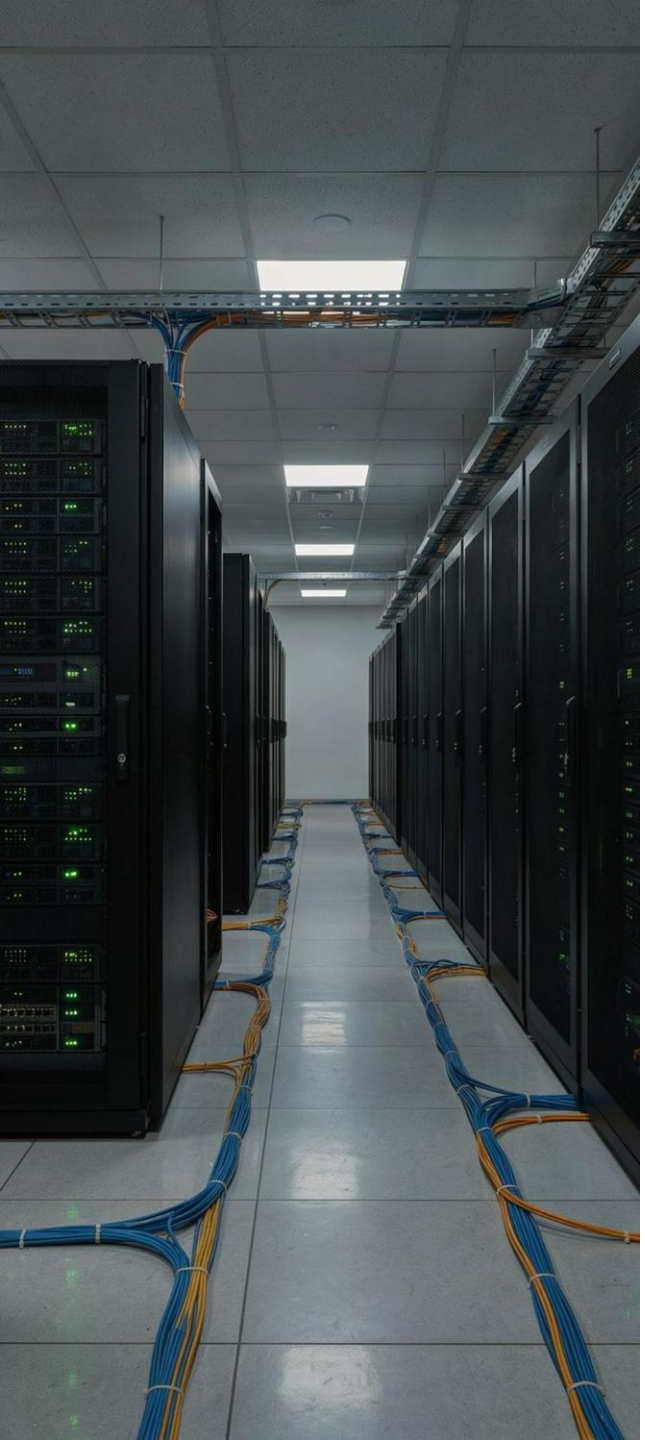
Target solution design — documenting application type, data platform, integrations, and automation approach, all reviewed and approved by the solution architect and technical lead.

Where AI Can Help

- Draft architecture options with tradeoff comparisons
- Propose initial Dataverse data models and table relationships
- Generate workflow design drafts for technical review

Acceptance Criteria

- Application type, data platform, integrations, and automation approach are documented and approved
- Design addresses required user journeys and access needs
- Error handling and supportability considerations are covered



PHASE 5

Plan the Data and Integrations

Data migration is consistently one of the highest-risk phases of any modernization project. Resolve mapping, quality, and reconciliation questions before a single record moves to production.

1 Map Fields and Relationships

Document every in-scope table, field, relationship, attachment, and history record. Define how legacy identifiers translate to the new data model.

2 Define Transformation and Reconciliation Rules

AI can suggest field mappings, flag data-quality issues, draft transformation logic, and generate reconciliation queries for trial migration runs.

3 Validate Integration Interfaces

Each integration must pass normal, failure, and recovery scenarios without unexplained data loss or duplication before production cutover.

PHASE 6

Establish Environments and Security

What Needs to Happen

Provision development, test, and production environments with consistent configuration. Set up security roles, data-loss prevention connector policies, and automated deployment controls before any build work begins.

Where AI Can Help

- Draft role-access matrices from requirements and user-role inventory
- Generate environment configuration checklists
- Flag potential permission gaps for security review before deployment
- Draft deployment pipeline documentation and runbooks

Acceptance Criteria

- Development, test, and production environments are fully configured
- Tests confirm users can perform authorized actions and cannot access restricted data
- Connector policies are reviewed and approved
- A deployment rehearsal succeeds end-to-end using environment-specific configuration

Never share production credentials or sensitive personal data with AI tools during environment setup. Use synthetic or anonymized data throughout.

PHASE 7

Build the Application Incrementally

Deliver working software in short, demonstrable increments. Regular reviews with product owners catch misalignments early and keep the backlog aligned with real business needs.

AI Accelerates Development

AI can assist with Power Fx formulas, workflow logic, screen layout suggestions, and technical documentation — reducing time on boilerplate and freeing developers for complex logic.

Errors Become Learning Moments

Developers can paste error messages into AI tools for plain-language explanations and candidate fixes — but every suggested fix must be tested and understood before it is committed.

Traceability Is Non-Negotiable

Every implemented requirement must trace to working, accepted functionality. Business rules, approvals, notifications, and reports must behave exactly as specified in acceptance criteria.

PHASE 8

Test and Obtain Business Acceptance

Scope of Testing

- Functional correctness across all business scenarios
- Permission and access-control verification
- Integration data accuracy and error recovery
- Performance under representative load
- Accessibility to agreed standards
- User Acceptance Testing conducted by business representatives

Where AI Can Help

- Generate test cases directly from approved requirements
- Suggest edge cases and negative scenarios teams often miss
- Create synthetic test data sets at scale
- Summarize defect reports and coverage gaps for stakeholder reviews

Acceptance Criteria

- All critical business scenarios pass
- No open critical or high-severity defects — exceptions require written authorization
- Performance and accessibility targets met
- Business representatives complete UAT and sign off on release readiness

PHASE 9

Prepare and Execute Cutover

1

Train Users

Deliver role-based training using AI-drafted materials. Confirm support coverage is in place before go-live.

2

Rehearse Migration

Run a full cutover rehearsal. AI analyzes rehearsal results to surface missed dependencies and timing risks.

3

Final Data Load

Load final changes, run reconciliation checks, and validate production data before releasing access to users.

4

Go Live & Smoke Test

Execute production smoke tests. Rollback triggers, owners, and procedures must be documented and rehearsed.

PHASE 10

Transition to Support and Retire the Legacy System

What Needs to Happen

Assign a named support owner, transfer documentation and runbooks, monitor adoption metrics and error rates through hypercare, and retire the legacy application only after all archival and acceptance conditions are satisfied.

Where AI Can Help

- Draft support guides, FAQs, and escalation procedures
- Summarize recurring incidents into actionable patterns
- Analyze adoption feedback and suggest backlog improvements

Acceptance Criteria

- Support owner accepts runbook, access, monitoring, and escalation responsibilities
- Support team demonstrates independent resolution of representative incidents
- Application meets agreed stability targets for [X days]
- Required legacy records are archived and retrievable before retirement approval

Strongest Starting Opportunities for AI Assistance

Not all phases benefit equally from AI involvement. These four areas deliver tangible, reviewable outputs immediately — making them the ideal entry points for teams new to AI-assisted delivery.

Requirements Drafting

Convert raw stakeholder notes into structured user stories with Given/When/Then acceptance criteria. Reviewable within hours rather than days.

Documentation Analysis

Summarize legacy specs, extract business rules, and identify gaps — dramatically reducing the time spent reading and interpreting existing material.

Development Assistance

Accelerate Power Fx, workflow logic, and screen design. AI explains errors in plain language and proposes candidate fixes for developers to evaluate.

Test-Case Generation

Generate comprehensive test cases from approved requirements, including edge cases and negative paths that manual review often misses.

AI Governance: Acceptance Criteria Across All Phases

AI accelerates delivery — but accountability remains with named human owners. Apply these criteria consistently across every phase to maintain quality, security, and audit readiness.

Area	Acceptance Criteria
Requirements & Documentation	A designated owner reviews AI-generated content against source material. Unsupported assumptions are resolved or explicitly recorded before approval.
Formulas, Code & Workflows	AI-generated components pass the same review, security checks, and functional tests as all other components. No untested generated logic is deployed.
Migration & Data Mapping	Proposed mappings and transformations are validated using representative records and reconciliation results before any production use.
Testing	Test cases trace to approved requirements. Expected results are independently verified, and actual execution evidence supports every pass/fail decision.
Data Handling	Information supplied to AI tools complies with the organization's approved data-handling rules. No sensitive production data is shared without authorization.
Approval & Accountability	Named business or technical owners approve all deliverables and release decisions — including those prepared with AI assistance.