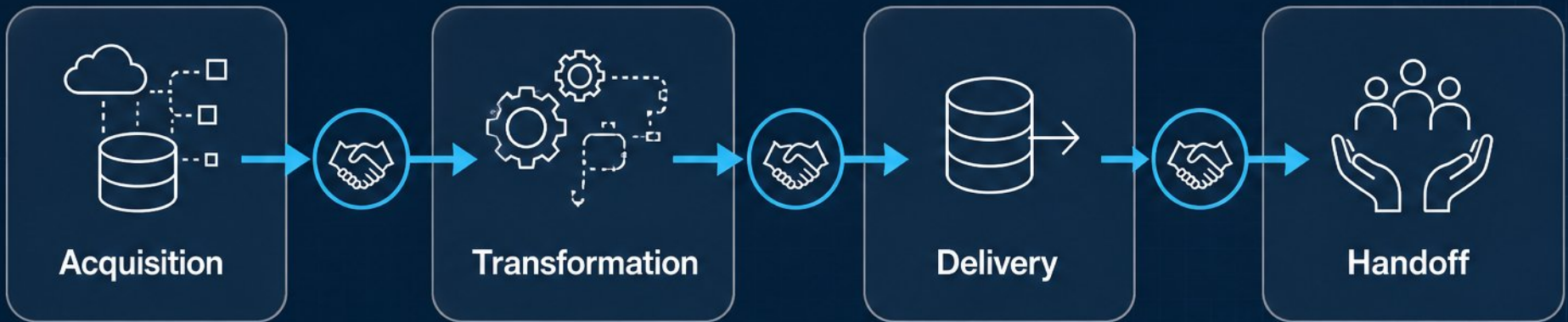


Data Engineering Workflow: Steps, Handoffs, and Improvement



- Structured workflows build trust in data products
- Core stages: acquisition, transformation, delivery, handoff
- Clear handoffs define ownership and responsibilities
- Continuous improvement is a core practice, not an afterthought



Why Undefined Workflows Fail



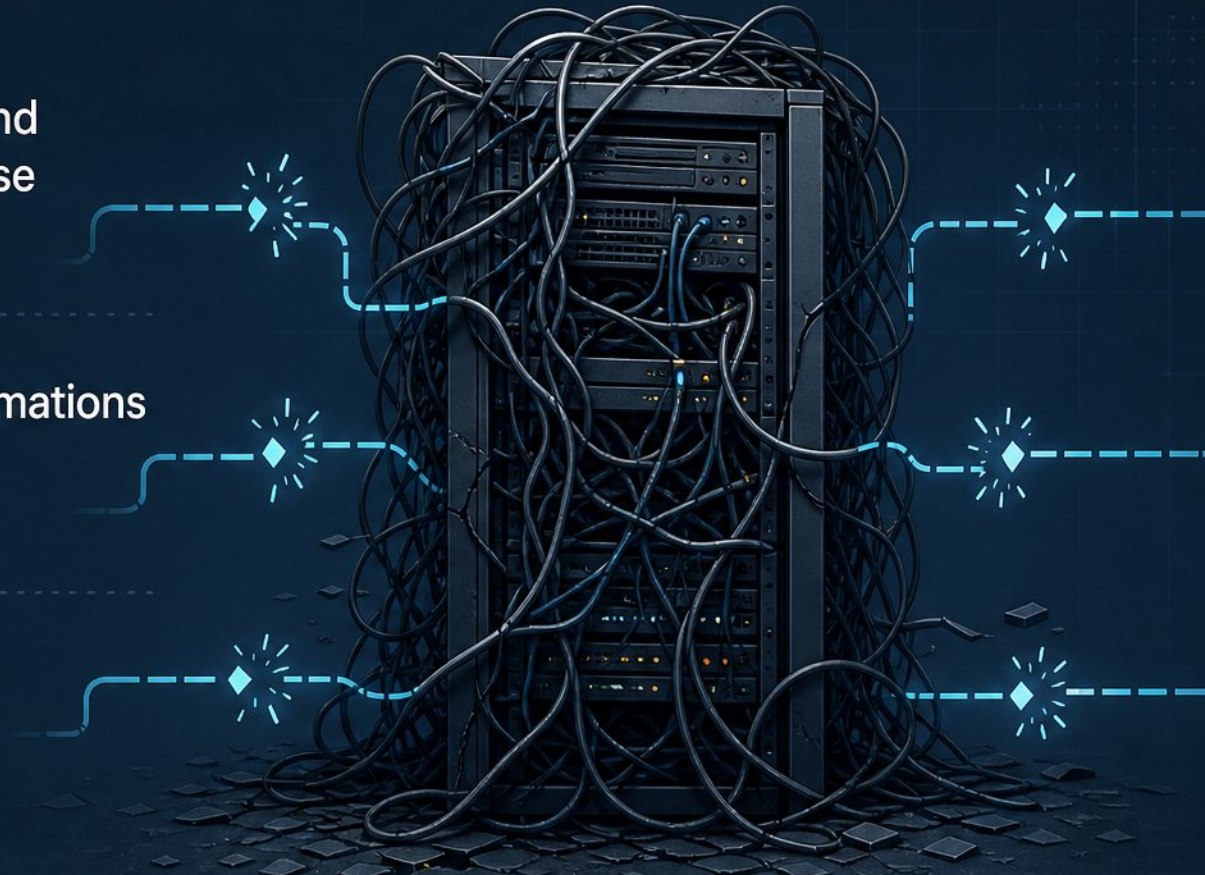
- Unclear ownership and missed handoffs cause pipeline breaks



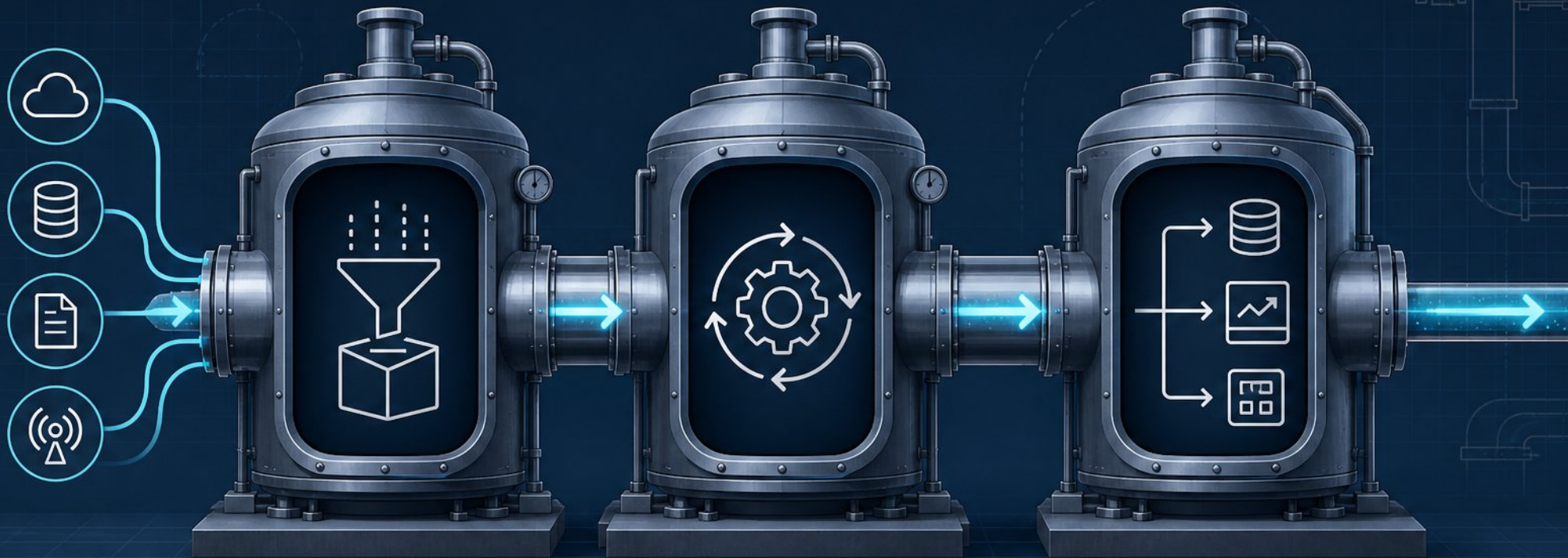
- Inconsistent transformations erode data trust and decisions



- Explicit stages and handoff protocols prevent failures



The Three Core Pipeline Stages



ACQUISITION

- Acquisition: sources, connectors, and landing zones



TRANSFORMATION

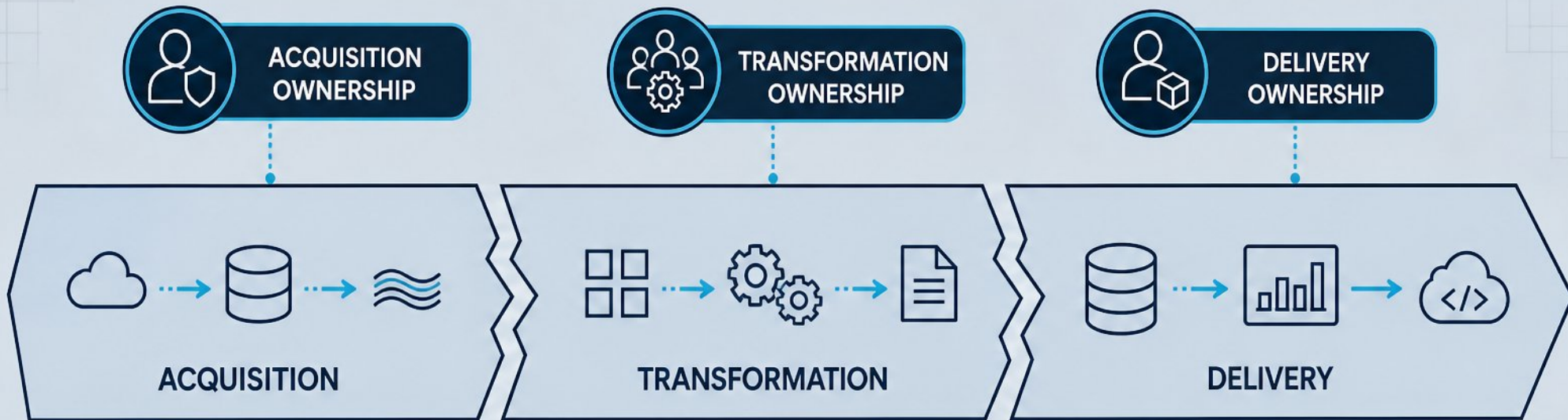
- Transformation: cleaning, modeling, enrichment, business logic



DELIVERY

- Delivery: serving datasets to warehouses, dashboards, apps

Why Stage Boundaries Matter for Ownership



- Stage boundaries define explicit pipeline ownership
- Acquisition tools: connectors, landing zones, ingestion services
- Transformation tools: dbt, Spark, SQL engines
- Delivery tools: warehouses, dashboards, APIs
- Boundaries reduce ambiguity and cross-team conflict



Defining Handoffs and Responsibilities



- Handoffs mark ownership changes across teams and systems



- Define acceptance criteria at each boundary



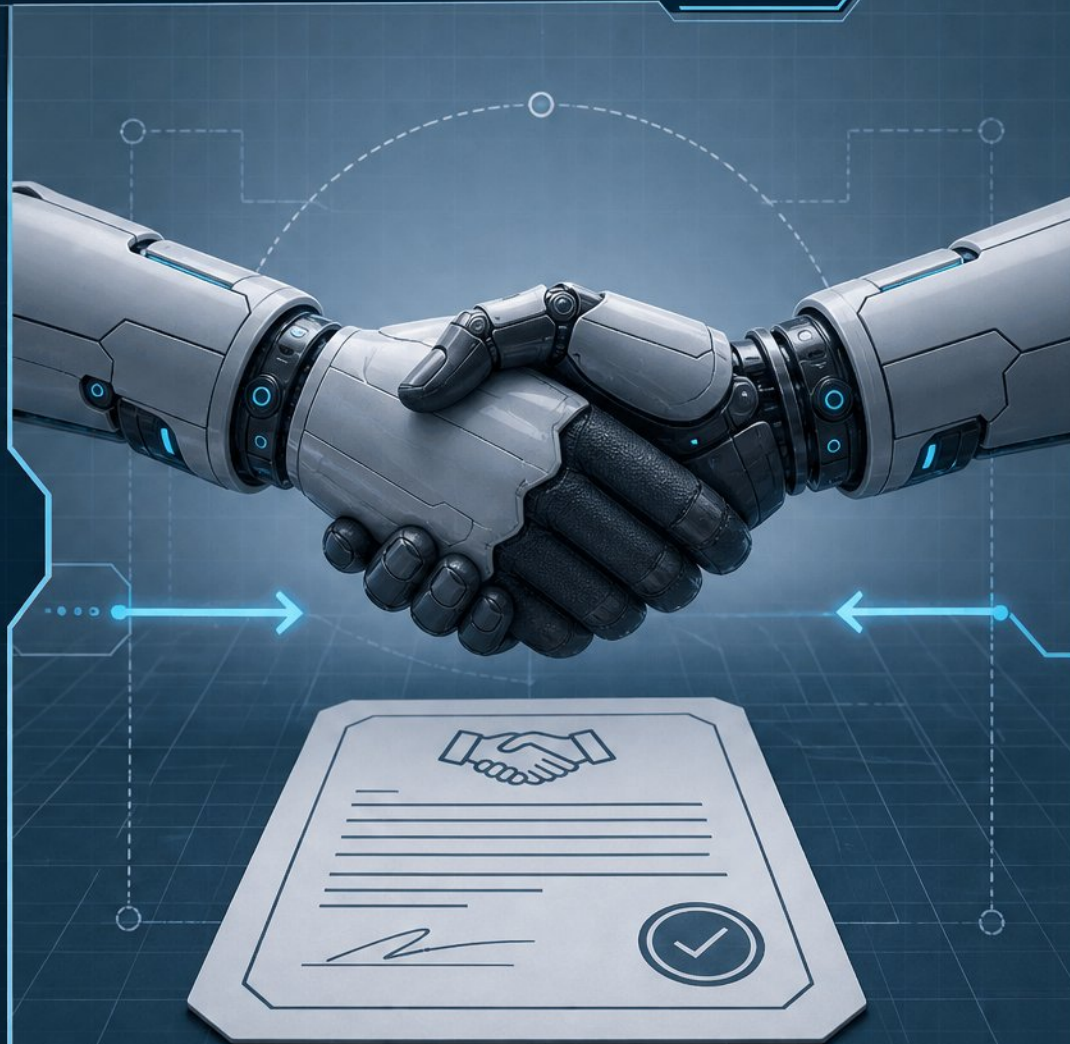
- Clear responsibilities prevent gaps and overlaps



- Document handoffs for auditability and review



- Treat handoffs as formal, observable contract points



HANDSHAKE PROTOCOL

Handoff Anti-Patterns and RACI in Data Contexts



- Common anti-patterns: ownership gaps, vague handoffs, rework loops



- RACI clarifies Responsible, Accountable, Consulted, Informed roles



- Prevents ambiguity at each pipeline stage boundary



- Map RACI to both datasets and transformation logic



- Review assignments during incident retrospectives



Pipeline Stage	RACI			
	R Responsible	A Accountable	C Consulted	I Informed
Ingest	Responsible	Accountable	Consulted	Informed
Transform	Responsible	Accountable	Consulted	Informed
Publish	Responsible	Accountable	Consulted	Informed
Consume	Responsible	Accountable	Consulted	Informed



HANDSHAKE PROTOCOL

Clear expectations. Shared context.
Reliable handoffs.

- ✓ Define scope and deliverables
- ✓ Confirm RACI assignments
- ✓ Acknowledge and proceed

Quality Gates at Handoff Points



- Quality gates catch errors early, protecting downstream consumers



- Key checks: schema validation, null thresholds, freshness, volume anomalies



- Block on critical failures; use warnings for minor deviations



- Adjust gate strictness by stage, team, and data criticality



Balancing Strict Gates with Delivery Speed



- Strict gates reduce risk but slow data delivery



- Progressive enforcement: validate critical fields first



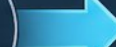
- Warnings for non-blocking issues, blocks for severe ones



- Stage-specific thresholds prevent unnecessary bottlenecks



- Align gate strictness with downstream business impact



Data Contracts and Lightweight Documentation

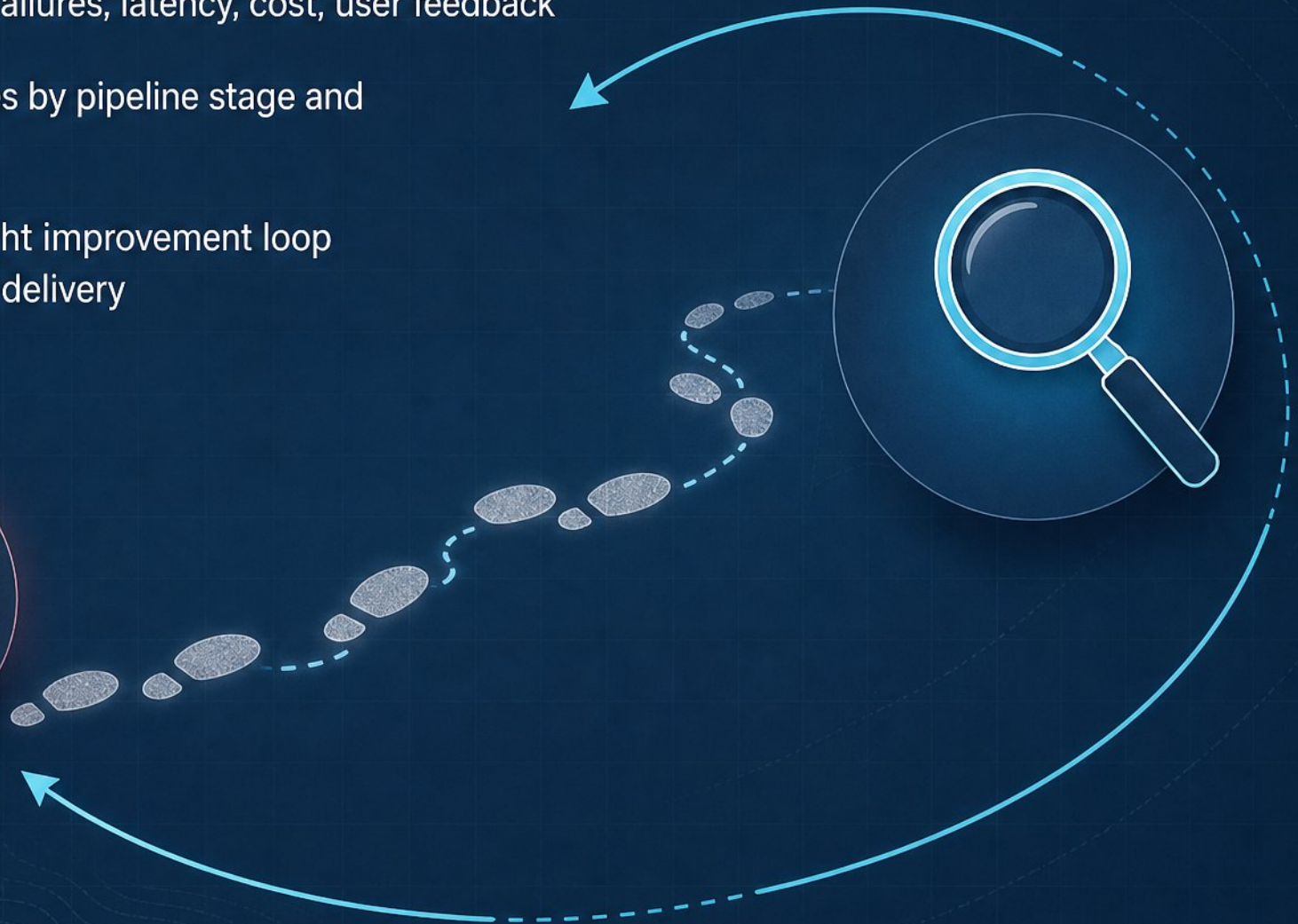


- Data contracts define schema, ownership, and expectations at each handoff
- Contracts reduce ambiguity and make pipeline changes safer
- Lightweight docs: templates, READMEs, and inline pipeline comments
- Documentation accelerates onboarding and speeds up troubleshooting
- Keep contracts living artifacts, reviewed and updated as pipelines evolve



Finding Improvement Opportunities

- Signal sources: failures, latency, cost, user feedback
- Categorize issues by pipeline stage and business priority
- Build a lightweight improvement loop without slowing delivery



Pipeline Health Metrics and Improvement Loops



- Track freshness, volume, schema, and data quality metrics
- Monitor stage-specific latency, failure rates, and error trends
- Alert on anomalies to catch issues before consumers notice
- Improvement loops: blameless reviews, root-cause fixes, and automated checks

Mapping the Framework to Your Team



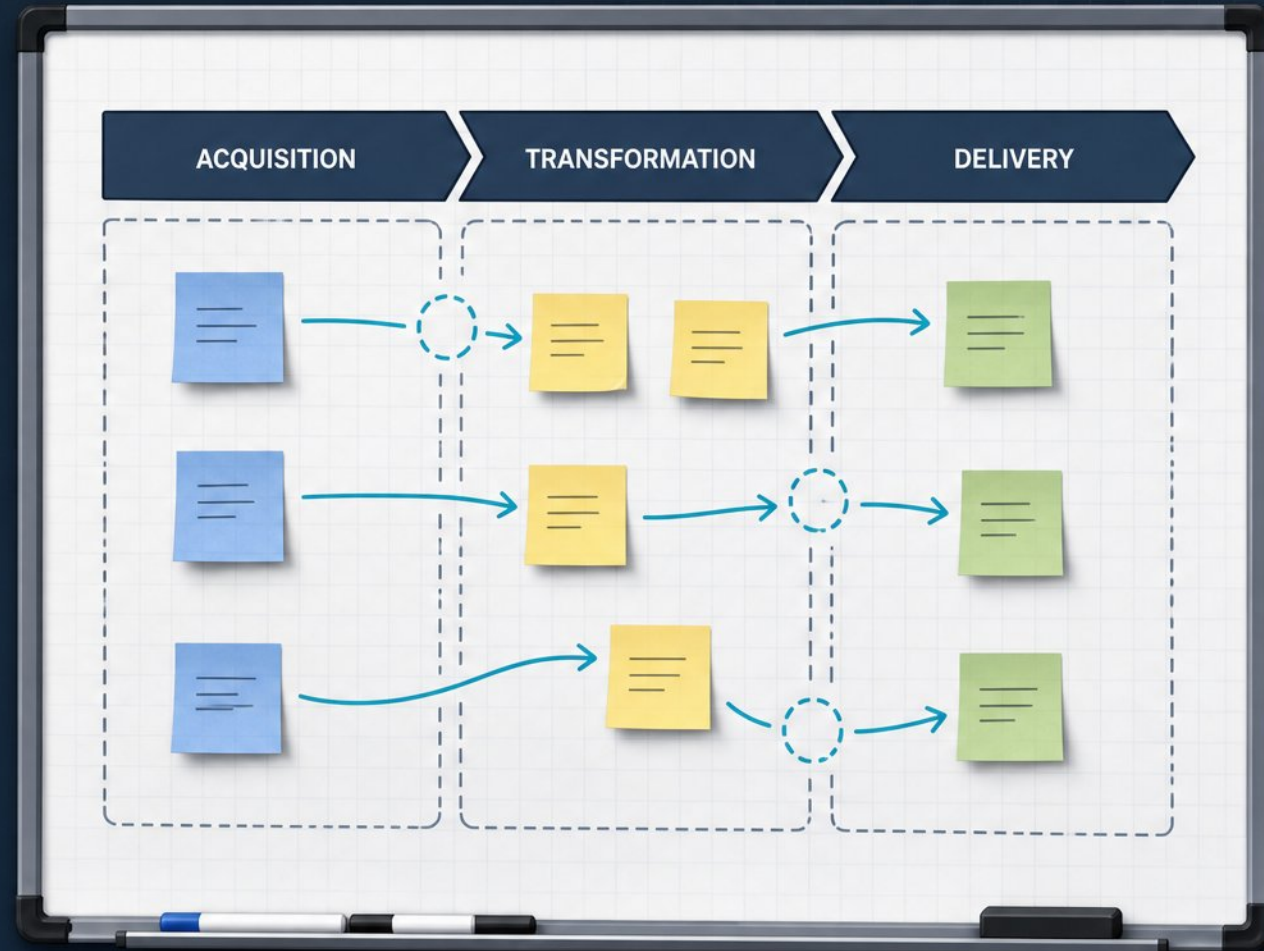
- Map each current step to acquisition, transformation, delivery



- Spot highest-risk handoffs: unclear ownership or missing checks



- Plan documentation, ownership, and quality gates for each stage

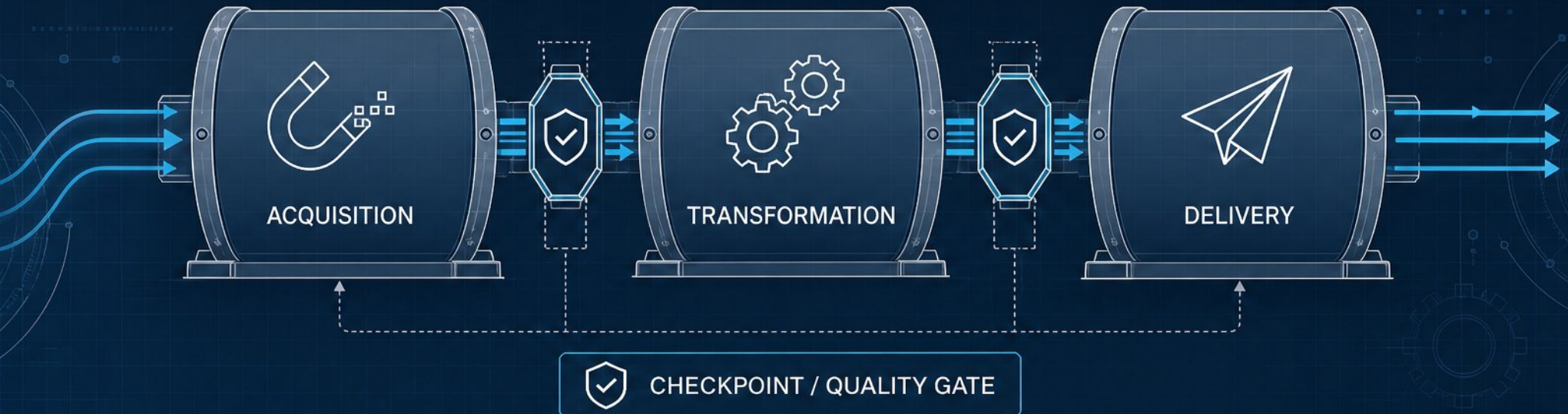


Adoption Challenges and Practical Adaptations



- - Avoid over-engineering process and documentation
- - Adapt ownership models to team size and maturity
- - Start with highest-risk handoffs, not all pipelines
- - Keep quality gates flexible until teams stabilize
- - Sustain improvements through lightweight rituals

Summary and Immediate Action Plan



- ✓ Recap three stages: acquisition, transformation, delivery
- ✓ Handoffs need clear ownership, acceptance criteria, documentation
- ✓ Apply RACI to prevent gaps; enforce quality gates at handoffs
- ✓ Start with highest-risk handoff points in your pipeline
- ✓ Track pipeline health metrics and build improvement loops

PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/accff7ba/public