PDFS OF REVOLUTION

State-of-art edition generated for 28 May 2026

AI RISK GOVERNANCE PLAYBOOK: THE TRUST AND RISK SYSTEM REVOLUTION OS

TRUST AND RISK SYSTEM

Repurposed from: 100_PAGE_PDFS\22_ai-risk-governance-playbook_100_pages.pdf

Theory first. Production logic second. No-code UI first. Knowledge base always.

Revolution Premise

> **Security production is the conversion of uncertainty into visible assets, controls, roles, evidence, and rehearsed response.**

This edition repurposes the source PDF into a deeper production manual. It keeps the source as the seed, but rewrites the mission around theory first, explicit definitions, practical execution, no-code UI-first implementation, and a knowledge-base operating system. Date stamp: 28 May 2026.

The reader should not merely consume ideas. The reader should leave with a theory of how the system works, a simple interface they can build for free or nearly free, a database schema, a feedback loop, and a way to evaluate whether the thing is true in the world.

Renamed Promise

Old signal: AI Risk Governance Playbook. New signal: AI Risk Governance Playbook: The Trust and Risk System Revolution OS. The new title names the underlying machine: not just a topic, but a production system with memory, interface, workflow, and evidence.

Core Terms From This Source

- Practical: track this as a recurring object in the knowledge base, not as a vague idea.
- Chapter: track this as a recurring object in the knowledge base, not as a vague idea.
- Checklist: track this as a recurring object in the knowledge base, not as a vague idea.
- Buyer: track this as a recurring object in the knowledge base, not as a vague idea.
- No-code: track this as a recurring object in the knowledge base, not as a vague idea.
- Ebook: track this as a recurring object in the knowledge base, not as a vague idea.
- Reader: track this as a recurring object in the knowledge base, not as a vague idea.
- Asset: track this as a recurring object in the knowledge base, not as a vague idea.
- Risk: track this as a recurring object in the knowledge base, not as a vague idea.
- Ui-first: track this as a recurring object in the knowledge base, not as a vague idea.

Definitions Needed Before Production

Definitions come before steps because undefined words become bad systems. If a term is not operationalized, the UI cannot capture it, the database cannot store it, and the automation cannot improve it.

Term	Operational definition
Production	The repeatable transformation of intention, knowledge, materials, tools, and feedback into a shipped artifact or outcome.
Knowledge base	A structured external memory that stores sources, decisions, examples, procedures, prompts, assets, metrics, and version history.
No-code UI-first solution	A solution that gives users an interface before custom software exists, usually by combining forms, tables, dashboards, automations, and simple pages.
RAG	Retrieval-augmented generation: connecting a model to external documents or records so answers can be grounded in current, domain-specific evidence.
Feedback loop	A cycle where outputs become inputs for the next decision through evidence, metrics, user response, or error signals.
Quality gate	A test that must pass before moving to the next stage, such as source citation, human review, conversion proof, security review, or usability check.
Asset	Anything valuable enough to protect: account, device, data set, system, process, reputation, or access path.
Control	A policy, process, technical safeguard, or behavior that reduces likelihood or impact of a risk.
Residual risk	The risk that remains after controls are applied and reviewed.

Production Philosophy

Production is not hustle. Production is a chain of transformations. Raw attention becomes notes. Notes become distinctions. Distinctions become procedures. Procedures become interfaces. Interfaces create data. Data creates feedback. Feedback improves the procedure. A product is mature only when this loop can repeat without heroic effort.

The hidden layer behind every strong ebook, course, template, service, agent, or app is a knowledge base. The visible product is the front stage. The knowledge base is the backstage: source evidence, examples, buyer language, prompts, decisions, metrics, and quality gates.

Scientific and Operational Principles Applied

The edition uses nine principles repeatedly. They appear in the modules as lenses for making decisions, not decorative citations.

Principle	How it changes production
Systems thinking [Meadows, 1999]	Look for stocks, flows, feedback loops, delays, incentives, rules, and goals. The deeper move is often changing the system rule, not adding more effort.
Theory of Constraints [Goldratt, Theory of Constraints]	Find the one constraint that limits throughput, exploit it, subordinate other work to it, then elevate it only after measurement proves it is still the bottleneck.
Process approach and PDCA [ISO 9001 process approach]	Treat every promise as a process: plan the work, do the smallest useful version, check evidence, then act on findings in the next cycle.
Validated learning [Ries, 2011]	Do not ask whether an idea feels good. Ask what behavior changed, what was learned, and what decision the evidence now supports.
Cognitive load [Sweller, 1988]	Design the next action so working memory is not overloaded. Reduce extraneous load, sequence intrinsic load, and use examples before open-ended tasks.
Retrieval practice [Roediger and Karpicke, 2006]	Turn knowledge into prompts, questions, drills, and decisions. If the learner never retrieves, the product is only a library, not a learning machine.
Deliberate practice [Ericsson, Krampe, and Tesch-Romer, 1993]	Skill improves through focused tasks, immediate feedback, error correction, and repetition near the edge of ability, not passive consumption.
Retrieval augmented generation [Lewis et al., 2020]	AI output is more useful when grounded in an external memory. Build retrieval, citations, and source records before scaling generation.
Trustworthy AI risk management [NIST AI RMF 1.0, 2023]	Govern, map, measure, and manage AI risks. Define who owns data, decisions, review, errors, and customer impact before automation touches production.

Practical rule: every major idea must become a record, a step, a screen, a metric, and a review question. If it cannot do that, it is still only inspiration.

State-of-Art Tool Stack, Free First

This stack is written for 28 May 2026. Specific pricing, model names, and product features change quickly, so the operating principle is more important than loyalty to a tool. Start free, prove the loop manually, then upgrade only where friction or scale justifies it.

Free-first route: Sheets/Airtable asset register, Notion policy library, Forms incident intake, AppSheet risk dashboard, and Make/Zapier/n8n notifications with least-privilege accounts.

Layer	Free-first / low-cost build	State-of-art logic
Knowledge base	Notion, Google Docs, Google Sheets, Airtable free tier	Keep sources, decisions, prompts, procedures, and metrics permission-aware and versioned.
UI first	Tally, Google Forms, AppSheet, Softr, Glide, Carrd, Google Sites	A visible interface makes the system testable before custom code exists.
Design and packaging	Canva, Google Slides, Docs export, existing cover PNGs	A cover, preview, and sample reduce buyer uncertainty faster than long claims.
Automation	Make, Zapier, n8n, Apps Script	Automate handoffs only after the manual path is understood and the failure modes are known.
AI layer	ChatGPT/OpenAI, Claude, Gemini, local/open models where appropriate	Use models for bounded tasks with source grounding, review, and evals.
Agent/tool layer	OpenAI Agents SDK, MCP-compatible connectors, Copilot Studio, Zapier Agents, Make AI Agents	Agents need tools, permissions, traces, and rollback paths, not just prompts.
Measurement	Sheets dashboards, Airtable interfaces, platform analytics, UTM links	The loop improves only when behavior, quality, and risk are measured.

Knowledge Base Rule

Knowledge base focus: assets, threats, controls, evidence, incidents, vendors, owners, residual risk, and review dates.

Knowledge Base Architecture

Use a database, not a pile of documents. The minimum viable knowledge base has linked records. Each record should answer what it is, why it matters, where the evidence lives, how it is used, and how quality is checked.

Table	Core fields	Why it exists
Source Library	Source, URL/file, date, claim, reliability, quote/snippet, tags	Prevents hallucination, vague claims, and forgotten context.
Buyer/Reader Jobs	Segment, pain, desired outcome, language, objection, proof needed	Connects production to demand and usefulness.
Concept Map	Definition, examples, non-examples, principle, dependency, misuse	Turns theory into teachable structure.
Procedure/SOP	Trigger, inputs, steps, owner, tool, output, quality gate, failure mode	Makes repeated work reliable.
Prompt/Agent Task	Goal, context, sources, constraints, tool access, expected output, eval	Keeps AI work bounded and testable.
Experiment Log	Hypothesis, build, metric, result, decision, next iteration	Prevents random activity and creates validated learning.
Asset Register	PDF, template, UI, landing page, automation, owner, version, status	Makes the product portfolio visible.

If the knowledge base is weak, the ebook becomes a one-time document. If the knowledge base is strong, the ebook becomes an operating system that can produce templates, tools, coaching, agents, dashboards, and future editions.

Source-to-Revolution Expansion Modules

Every module below repurposes source pages into a deeper production unit. The source page anchors keep the original content represented; the added layers supply theory, definitions, no-code implementation, knowledge-base structure, metrics, and practical steps.

Module 01: Source Pages 1, 2, 3, 4 - AI Risk Governance | Strategy | Page 002 | Strategy | Page 003 | Tech Stack | Page 004

Original seed: AI Risk Governance Playbook 100-page no-code UI-first implementation blueprint Category: Cybersecurity / AI Risk / Online Protection Buyer: Agencies, consultants, SMBs using AI tools High-ticket angle: Avoid privacy, legal, and operational disasters from unmanaged AI use. Product format: Ebook + AI policy + vendor checklist + approval workflow Price range: $97-$299 Low-budget stack. No-code workflows. UI-first build path. Practical execution. Strategy | Page 002 Positioning Snapshot AI Risk Governance Playbook sells to Agencies, consultants, SMBs using AI tools. The pain is simple: Teams are using AI without privacy, legal, quality, or operational controls. The high-ticket...

Theory first: use Systems thinking [Meadows, 1999]. Look for stocks, flows, feedback loops, delays, incentives, rules, and goals. The deeper move is often changing the system rule, not adding more effort.

Definition needed: Asset means Anything valuable enough to protect: account, device, data set, system, process, reputation, or access path.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, No-code, Ui-first, Ebook, Low-budget into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 02: Source Pages 5, 6, 7, 8 - Strategy | Page 005 | Framework | Page 006 | Monetization | Page 007 | Execution | Page 008

Original seed: Strategy | Page 005 Buyer Journey The buyer starts with this pain: Teams are using AI without privacy, legal, quality, or operational controls. The ebook should move them through awareness, setup, first result, repeatable workflow, proof, and paid offer. Practical Build Moves Make the first win possible in one day. Make the full system possible in seven to thirty days. Use the UI/dashboard as proof of progress. End with a service, product, or career asset. Before state After state Practical low-budget no-code UI-first ebook blueprint. 5 Framework | Page 006 Core Operating Mechanism The mechanism...

Theory first: use Theory of Constraints [Goldratt, Theory of Constraints]. Find the one constraint that limits throughput, exploit it, subordinate other work to it, then elevate it only after measurement proves it is still the bottleneck.

Definition needed: Control means A policy, process, technical safeguard, or behavior that reduces likelihood or impact of a risk.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Ebook, Practical, Asset, Week, Blueprint into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 03: Source Pages 9, 10, 11, 12 - Quality | Page 009 | Monetization | Page 010 | Chapter 1 | Page 011 | Chapter 1 | Page 012

Original seed: Quality | Page 009 Risk And Claims Safety Avoid guarantees. Teach process, education, templates, and implementation. High-ticket topics can still be honest without promising income, rankings, health outcomes, security certainty, or employment. Use disclaimers for wellness, cybersecurity, finance, career, and income topics. Practical Build Moves Replace guaranteed outcomes with implementation outcomes. Add human review before publishing. Flag legal, financial, medical, cybersecurity, employment, and income claims. Use examples as education, not proof of guaranteed results. Risky claim Safer wording Practical low-budget no-code UI-first ebook blueprint. 9 Monetization | Page 010 Sales Page Spine The sales page...

Theory first: use Process approach and PDCA [ISO 9001 process approach]. Treat every promise as a process: plan the work, do the smallest useful version, check evidence, then act on findings in the next cycle.

Definition needed: Residual risk means The risk that remains after controls are applied and reviewed.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Buyer, Pain, Practical, Chapter, No-code into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 04: Source Pages 13, 14, 15, 16 - Chapter 1 | Page 013 | Chapter 1 | Page 014 | Chapter 1 | Page 015 | Chapter 1 | Page 016

Original seed: Chapter 1 | Page 013 Chapter 1: Market Pain And Buyer Psychology - Tool Setup Research the urgent pain, buyer language, buying triggers, and objections. List the low-budget tools and exact setup sequence. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Airtable or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Risk checklist. End with a practical output the reader can inspect...

Theory first: use Validated learning [Ries, 2011]. Do not ask whether an idea feels good. Ask what behavior changed, what was learned, and what decision the evidence now supports.

Definition needed: Asset means Anything valuable enough to protect: account, device, data set, system, process, reputation, or access path.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Buyer, Pain, Practical, Chapter, Checklist into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 05: Source Pages 17, 18, 19, 20 - Chapter 1 | Page 017 | Chapter 1 | Page 018 | Chapter 2 | Page 019 | Chapter 2 | Page 020

Original seed: Chapter 1 | Page 017 Chapter 1: Market Pain And Buyer Psychology - Worksheet Page Research the urgent pain, buyer language, buying triggers, and objections. Give fields the reader can fill in to produce a usable asset. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Bitwarden or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Risk checklist. End with a practical output...

Theory first: use Cognitive load [Sweller, 1988]. Design the next action so working memory is not overloaded. Reduce extraneous load, sequence intrinsic load, and use examples before open-ended tasks.

Definition needed: Control means A policy, process, technical safeguard, or behavior that reduces likelihood or impact of a risk.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Buyer, Chapter, Pain, Checklist into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 06: Source Pages 21, 22, 23, 24 - Chapter 2 | Page 021 | Chapter 2 | Page 022 | Chapter 2 | Page 023 | Chapter 2 | Page 024

Original seed: Chapter 2 | Page 021 Chapter 2: UI-First System Design - Tool Setup Design the dashboard, form, landing page, workflow board, or checklist interface first. List the low-budget tools and exact setup sequence. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Airtable or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Policy template. End with a practical output the reader can inspect...

Theory first: use Retrieval practice [Roediger and Karpicke, 2006]. Turn knowledge into prompts, questions, drills, and decisions. If the learner never retrieves, the product is only a library, not a learning machine.

Definition needed: Residual risk means The risk that remains after controls are applied and reviewed.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Checklist, Chapter, Dashboard, Design into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 07: Source Pages 25, 26, 27, 28 - Chapter 2 | Page 025 | Chapter 2 | Page 026 | Chapter 3 | Page 027 | Chapter 3 | Page 028

Original seed: Chapter 2 | Page 025 Chapter 2: UI-First System Design - Worksheet Page Design the dashboard, form, landing page, workflow board, or checklist interface first. Give fields the reader can fill in to produce a usable asset. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Bitwarden or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Policy template. End with a practical output...

Theory first: use Deliberate practice [Ericsson, Krampe, and Tesch-Romer, 1993]. Skill improves through focused tasks, immediate feedback, error correction, and repetition near the edge of ability, not passive consumption.

Definition needed: Asset means Anything valuable enough to protect: account, device, data set, system, process, reputation, or access path.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, No-code, Ui-first, Checklist into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 08: Source Pages 29, 30, 31, 32 - Chapter 3 | Page 029 | Chapter 3 | Page 030 | Chapter 3 | Page 031 | Chapter 3 | Page 032

Original seed: Chapter 3 | Page 029 Chapter 3: No-Code Stack Setup - Tool Setup Set up the affordable tools, accounts, data structure, permissions, and workspace. List the low-budget tools and exact setup sequence. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Airtable or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Training script. End with a practical output the reader can inspect. Practical...

Theory first: use Retrieval augmented generation [Lewis et al., 2020]. AI output is more useful when grounded in an external memory. Build retrieval, citations, and source records before scaling generation.

Definition needed: Control means A policy, process, technical safeguard, or behavior that reduces likelihood or impact of a risk.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, No-code, Checklist, Script into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 09: Source Pages 33, 34, 35, 36 - Chapter 3 | Page 033 | Chapter 3 | Page 034 | Chapter 4 | Page 035 | Chapter 4 | Page 036

Original seed: Chapter 3 | Page 033 Chapter 3: No-Code Stack Setup - Worksheet Page Set up the affordable tools, accounts, data structure, permissions, and workspace. Give fields the reader can fill in to produce a usable asset. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Bitwarden or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Training script. End with a practical output the...

Theory first: use Trustworthy AI risk management [NIST AI RMF 1.0, 2023]. Govern, map, measure, and manage AI risks. Define who owns data, decisions, review, errors, and customer impact before automation touches production.

Definition needed: Residual risk means The risk that remains after controls are applied and reviewed.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, No-code, Output, Reader into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 10: Source Pages 37, 38, 39, 40 - Chapter 4 | Page 037 | Chapter 4 | Page 038 | Chapter 4 | Page 039 | Chapter 4 | Page 040

Original seed: Chapter 4 | Page 037 Chapter 4: Workflow And Automation Map - Tool Setup Map the practical workflow from input to output, including manual checkpoints. List the low-budget tools and exact setup sequence. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Airtable or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Incident plan. End with a practical output the reader can inspect...

Theory first: use Systems thinking [Meadows, 1999]. Look for stocks, flows, feedback loops, delays, incentives, rules, and goals. The deeper move is often changing the system rule, not adding more effort.

Definition needed: Asset means Anything valuable enough to protect: account, device, data set, system, process, reputation, or access path.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Workflow, Output, Chapter, Checklist into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 11: Source Pages 41, 42, 43, 44 - Chapter 4 | Page 041 | Chapter 4 | Page 042 | Chapter 5 | Page 043 | Chapter 5 | Page 044

Original seed: Chapter 4 | Page 041 Chapter 4: Workflow And Automation Map - Worksheet Page Map the practical workflow from input to output, including manual checkpoints. Give fields the reader can fill in to produce a usable asset. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Bitwarden or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Incident plan. End with a practical output...

Theory first: use Theory of Constraints [Goldratt, Theory of Constraints]. Find the one constraint that limits throughput, exploit it, subordinate other work to it, then elevate it only after measurement proves it is still the bottleneck.

Definition needed: Control means A policy, process, technical safeguard, or behavior that reduces likelihood or impact of a risk.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, Ebook, Output, Reader into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 12: Source Pages 45, 46, 47, 48 - Chapter 5 | Page 045 | Chapter 5 | Page 046 | Chapter 5 | Page 047 | Chapter 5 | Page 048

Original seed: Chapter 5 | Page 045 Chapter 5: Templates, Prompts, And Scripts - Tool Setup Build reusable assets that make the ebook feel immediately useful. List the low-budget tools and exact setup sequence. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Airtable or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Risk checklist. End with a practical output the reader can inspect. Practical...

Theory first: use Process approach and PDCA [ISO 9001 process approach]. Treat every promise as a process: plan the work, do the smallest useful version, check evidence, then act on findings in the next cycle.

Definition needed: Residual risk means The risk that remains after controls are applied and reviewed.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, Ebook, Checklist, Script into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 13: Source Pages 49, 50, 51, 52 - Chapter 5 | Page 049 | Chapter 5 | Page 050 | Chapter 6 | Page 051 | Chapter 6 | Page 052

Original seed: Chapter 5 | Page 049 Chapter 5: Templates, Prompts, And Scripts - Worksheet Page Build reusable assets that make the ebook feel immediately useful. Give fields the reader can fill in to produce a usable asset. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Bitwarden or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Risk checklist. End with a practical output the...

Theory first: use Validated learning [Ries, 2011]. Do not ask whether an idea feels good. Ask what behavior changed, what was learned, and what decision the evidence now supports.

Definition needed: Asset means Anything valuable enough to protect: account, device, data set, system, process, reputation, or access path.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, Reader, Checklist, Create into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 14: Source Pages 53, 54, 55, 56 - Chapter 6 | Page 053 | Chapter 6 | Page 054 | Chapter 6 | Page 055 | Chapter 6 | Page 056

Original seed: Chapter 6 | Page 053 Chapter 6: First Result In 24 Hours - Tool Setup Create a fast, visible win the reader can complete quickly. List the low-budget tools and exact setup sequence. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Airtable or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Policy template. End with a practical output the reader can inspect...

Theory first: use Cognitive load [Sweller, 1988]. Design the next action so working memory is not overloaded. Reduce extraneous load, sequence intrinsic load, and use examples before open-ended tasks.

Definition needed: Control means A policy, process, technical safeguard, or behavior that reduces likelihood or impact of a risk.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Reader, Chapter, Create, Visible into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 15: Source Pages 57, 58, 59, 60 - Chapter 6 | Page 057 | Chapter 6 | Page 058 | Chapter 7 | Page 059 | Chapter 7 | Page 060

Original seed: Chapter 6 | Page 057 Chapter 6: First Result In 24 Hours - Worksheet Page Create a fast, visible win the reader can complete quickly. Give fields the reader can fill in to produce a usable asset. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Bitwarden or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Policy template. End with a practical output...

Theory first: use Retrieval practice [Roediger and Karpicke, 2006]. Turn knowledge into prompts, questions, drills, and decisions. If the learner never retrieves, the product is only a library, not a learning machine.

Definition needed: Residual risk means The risk that remains after controls are applied and reviewed.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, Reader, Create, Visible into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 16: Source Pages 61, 62, 63, 64 - Chapter 7 | Page 061 | Chapter 7 | Page 062 | Chapter 7 | Page 063 | Chapter 7 | Page 064

Original seed: Chapter 7 | Page 061 Chapter 7: Seven-Day Implementation Sprint - Tool Setup Turn the first win into a working system with daily tasks. List the low-budget tools and exact setup sequence. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Airtable or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Training script. End with a practical output the reader can inspect. Practical...

Theory first: use Deliberate practice [Ericsson, Krampe, and Tesch-Romer, 1993]. Skill improves through focused tasks, immediate feedback, error correction, and repetition near the edge of ability, not passive consumption.

Definition needed: Asset means Anything valuable enough to protect: account, device, data set, system, process, reputation, or access path.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, Checklist, Script, Buyer into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 17: Source Pages 65, 66, 67, 68 - Chapter 7 | Page 065 | Chapter 7 | Page 066 | Chapter 8 | Page 067 | Chapter 8 | Page 068

Original seed: Chapter 7 | Page 065 Chapter 7: Seven-Day Implementation Sprint - Worksheet Page Turn the first win into a working system with daily tasks. Give fields the reader can fill in to produce a usable asset. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Bitwarden or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Training script. End with a practical output the...

Theory first: use Retrieval augmented generation [Lewis et al., 2020]. AI output is more useful when grounded in an external memory. Build retrieval, citations, and source records before scaling generation.

Definition needed: Control means A policy, process, technical safeguard, or behavior that reduces likelihood or impact of a risk.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, Reader, Script, Asset into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 18: Source Pages 69, 70, 71, 72 - Chapter 8 | Page 069 | Chapter 8 | Page 070 | Chapter 8 | Page 071 | Chapter 8 | Page 072

Original seed: Chapter 8 | Page 069 Chapter 8: Client, Customer, Or User Delivery - Tool Setup Package the system so someone else can use it or buy it. List the low-budget tools and exact setup sequence. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Airtable or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Incident plan. End with a practical output the reader...

Theory first: use Trustworthy AI risk management [NIST AI RMF 1.0, 2023]. Govern, map, measure, and manage AI risks. Define who owns data, decisions, review, errors, and customer impact before automation touches production.

Definition needed: Residual risk means The risk that remains after controls are applied and reviewed.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, Checklist, Script, Buyer into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 19: Source Pages 73, 74, 75, 76 - Chapter 8 | Page 073 | Chapter 8 | Page 074 | Chapter 9 | Page 075 | Chapter 9 | Page 076

Original seed: Chapter 8 | Page 073 Chapter 8: Client, Customer, Or User Delivery - Worksheet Page Package the system so someone else can use it or buy it. Give fields the reader can fill in to produce a usable asset. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Bitwarden or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Incident plan. End with a...

Theory first: use Systems thinking [Meadows, 1999]. Look for stocks, flows, feedback loops, delays, incentives, rules, and goals. The deeper move is often changing the system rule, not adding more effort.

Definition needed: Asset means Anything valuable enough to protect: account, device, data set, system, process, reputation, or access path.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, Create, Outreach, Reader into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 20: Source Pages 77, 78, 79, 80 - Chapter 9 | Page 077 | Chapter 9 | Page 078 | Chapter 9 | Page 079 | Chapter 9 | Page 080

Original seed: Chapter 9 | Page 077 Chapter 9: Traffic, Outreach, And Sales - Tool Setup Create the landing page, outreach scripts, content hooks, and follow-up system. List the low-budget tools and exact setup sequence. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Airtable or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Risk checklist. End with a practical output the reader can inspect...

Theory first: use Theory of Constraints [Goldratt, Theory of Constraints]. Find the one constraint that limits throughput, exploit it, subordinate other work to it, then elevate it only after measurement proves it is still the bottleneck.

Definition needed: Control means A policy, process, technical safeguard, or behavior that reduces likelihood or impact of a risk.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, Create, Outreach, Checklist into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 21: Source Pages 81, 82, 83, 84 - Chapter 9 | Page 081 | Chapter 9 | Page 082 | Chapter 10 | Page 083 | Chapter 10 | Page 084

Original seed: Chapter 9 | Page 081 Chapter 9: Traffic, Outreach, And Sales - Worksheet Page Create the landing page, outreach scripts, content hooks, and follow-up system. Give fields the reader can fill in to produce a usable asset. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Bitwarden or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Risk checklist. End with a practical output...

Theory first: use Process approach and PDCA [ISO 9001 process approach]. Treat every promise as a process: plan the work, do the smallest useful version, check evidence, then act on findings in the next cycle.

Definition needed: Residual risk means The risk that remains after controls are applied and reviewed.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, Quality, Checklist, Create into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 22: Source Pages 85, 86, 87, 88 - Chapter 10 | Page 085 | Chapter 10 | Page 086 | Chapter 10 | Page 087 | Chapter 10 | Page 088

Original seed: Chapter 10 | Page 085 Chapter 10: Scale, Maintenance, And Quality - Tool Setup Add review loops, metrics, SOPs, safety checks, and upgrade paths. List the low-budget tools and exact setup sequence. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Airtable or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Policy template. End with a practical output the reader can inspect. Practical...

Theory first: use Validated learning [Ries, 2011]. Do not ask whether an idea feels good. Ask what behavior changed, what was learned, and what decision the evidence now supports.

Definition needed: Asset means Anything valuable enough to protect: account, device, data set, system, process, reputation, or access path.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Chapter, Quality, Checklist, Script into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 23: Source Pages 89, 90, 91, 92 - Chapter 10 | Page 089 | Chapter 10 | Page 090 | Assets | Page 091 | Assets | Page 092

Original seed: Chapter 10 | Page 089 Chapter 10: Scale, Maintenance, And Quality - Worksheet Page Add review loops, metrics, SOPs, safety checks, and upgrade paths. Give fields the reader can fill in to produce a usable asset. For AI Risk Governance Playbook, connect this page directly to the buyer pain: Teams are using AI without privacy, legal, quality, or operational controls. Practical Build Moves Use Bitwarden or a cheaper equivalent. Include one visible UI element: dashboard, table, form, page, checklist, or script. Create or reference this core asset: Policy template. End with a practical output the...

Theory first: use Cognitive load [Sweller, 1988]. Design the next action so working memory is not overloaded. Reduce extraneous load, sequence intrinsic load, and use examples before open-ended tasks.

Definition needed: Control means A policy, process, technical safeguard, or behavior that reduces likelihood or impact of a risk.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Asset, Quality, Ebook, No-code into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 24: Source Pages 93, 94, 95, 96 - Assets | Page 093 | Sales | Page 094 | Sales | Page 095 | UI | Page 096

Original seed: Assets | Page 093 Template Vault Plan Create 25 templates the buyer can duplicate and use immediately. Apply it to AI Risk Governance Playbook with a practical no-code stack: Bitwarden, Google Workspace, Notion, Airtable, Tally. Practical Build Moves Buyer: Agencies, consultants, SMBs using AI tools. Product format: Ebook + AI policy + vendor checklist + approval workflow. High-ticket angle: Avoid privacy, legal, and operational disasters from unmanaged AI use.. Make the asset duplicate-ready and screenshot-friendly. Asset notes Launch notes Practical low-budget no-code UI-first ebook blueprint. 93 Sales | Page 094 Outreach Scripts Create 20 outreach...

Theory first: use Retrieval practice [Roediger and Karpicke, 2006]. Turn knowledge into prompts, questions, drills, and decisions. If the learner never retrieves, the product is only a library, not a learning machine.

Definition needed: Residual risk means The risk that remains after controls are applied and reviewed.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Asset, Ebook, No-code, Notes into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

Module 25: Source Pages 97, 98, 99, 100 - UI | Page 097 | Execution | Page 098 | Monetization | Page 099 | Quality | Page 100

Original seed: UI | Page 097 Google Sheets Calculator Create a calculator for pricing, budget, ROI, content volume, or risk scoring. Apply it to AI Risk Governance Playbook with a practical no-code stack: Bitwarden, Google Workspace, Notion, Airtable, Tally. Practical Build Moves Buyer: Agencies, consultants, SMBs using AI tools. Product format: Ebook + AI policy + vendor checklist + approval workflow. High-ticket angle: Avoid privacy, legal, and operational disasters from unmanaged AI use.. Make the asset duplicate-ready and screenshot-friendly. Asset notes Launch notes Practical low-budget no-code UI-first ebook blueprint. 97 Execution | Page 098 First Client In...

Theory first: use Deliberate practice [Ericsson, Krampe, and Tesch-Romer, 1993]. Skill improves through focused tasks, immediate feedback, error correction, and repetition near the edge of ability, not passive consumption.

Definition needed: Asset means Anything valuable enough to protect: account, device, data set, system, process, reputation, or access path.

Production logic: the source material should become an observable loop: input, interpretation, artifact, interface, feedback, decision, and next version.

- 1. Define the production object: turn Practical, Asset, Ebook, No-code, Notes into one named record, one user problem, and one promised outcome.
- 2. Build the free UI first: use a form, table, or simple page so the process can be tested by a real user before custom software.
- 3. Add the knowledge base: store source snippets, decisions, examples, prompts, and quality gates beside the workflow.
- 4. Run one manual cycle: produce the asset, collect evidence, log friction, and keep the human review step visible.
- 5. Automate only the stable handoff: trigger notifications, generate drafts, update records, or deliver files after the manual path works.
- 6. Check the metric: compare expected behavior to actual behavior, then decide whether to keep, change, or remove the step.

No-code UI-first solution:

- Interface: create a trust and risk system dashboard in Notion, Airtable, AppSheet, Softr, or Google Sheets before writing code.
- Input: collect user, buyer, learner, or client data through Tally, Google Forms, Typeform-style forms, or a simple intake page.
- Automation: connect records with Make, Zapier, n8n, or Apps Script only after each field and decision has a clear owner.
- AI: ask the model to draft, classify, score, or transform records only with source context and a quality rubric.

Knowledge base record to create: Record name, Definition, Source evidence, User problem, Procedure, UI field, Automation trigger, Review metric, Risk/failure mode, Next experiment.

> **Quality gate: a page is not complete when it is written. It is complete when a user can act on it, the system can store what happened, and the next iteration is obvious.**

90-Day Practical Build Roadmap

Phase	Theory	Build	Evidence
Days 1-10: Map	Define the system boundary, user job, constraint, and promise.	Create KB tables, collect source evidence, rewrite title/promise, and make first UI mockup.	A reader/buyer can explain what the product does and why it matters.
Days 11-30: Prototype	Reduce cognitive load and expose the smallest useful workflow.	Build form, dashboard, sample asset, checkout or delivery route, and manual fulfillment SOP.	One real test user completes the flow without private explanation.
Days 31-60: Produce	Use deliberate practice and PDCA to improve repeated output.	Create templates, prompt tasks, automations, examples, and quality gates.	The system produces consistent outputs with fewer corrections each cycle.
Days 61-90: Scale carefully	Use constraints and risk management before adding complexity.	Automate stable handoffs, add analytics, publish proof, and document failures.	Metrics show what to keep, stop, and improve.

Practical Rule Set

- Start with a manual workflow because manual work reveals the fields the database really needs.
- Build the UI before the automation because interfaces expose missing logic.
- Automate only after three successful manual runs or after one repeated error is clearly understood.
- Every AI output must have a source, a rubric, a reviewer, and a failure log.
- Every promise must have a proof asset before it becomes marketing copy.

Reusable Prompt and Agent Tasks

Use these as structured tasks inside ChatGPT, Claude, Gemini, Copilot Studio, Zapier Agents, Make AI Agents, n8n, or an OpenAI Agents SDK workflow. The important part is not the brand. The important part is grounding, constraints, and evaluation.

Task	Prompt
Source distiller	Given the source records for AI Risk Governance Playbook: The Trust and Risk System Revolution OS, extract claims, definitions, steps, examples, missing evidence, and risks. Return a table with source page, claim, confidence, needed citation, and action.
Knowledge-base builder	Convert this chapter into records for Source Library, Concept Map, SOP, Prompt Task, Experiment Log, and Asset Register. Do not invent sources. Mark unknowns explicitly.
No-code UI planner	Design a free-first UI for this trust and risk system system using forms, tables, dashboards, and simple automations. Include fields, screens, user actions, and quality gates.
Scientific reviewer	Review this section using cognitive load, retrieval practice, deliberate practice, systems thinking, constraints, PDCA, RAG, and NIST AI RMF. Identify what is missing, vague, risky, or untestable.
Conversion editor	Rewrite the offer without hype. Keep the promise strong but honest. Add proof requirements, objections, guarantees that are ethical, and a clear next action.

Citations and 2026 Reference Base

These sources informed the state-of-art framing as of 28 May 2026. Tool features, model names, pricing, and availability should be checked again immediately before publishing or selling.

Source	Use in this edition
OpenAI API model documentation https://platform.openai.com/docs/models	Model availability and API capabilities checked for the 2026 state-of-art tool stack.
OpenAI Agents SDK guide https://platform.openai.com/docs/guides/agents-sdk/	Agentic applications with tools, handoffs, tracing, and additional context.
NIST AI Risk Management Framework 1.0 https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-ai-rmf-10	Govern, map, measure, manage framing for AI risk.
NIST Generative AI Profile https://www.nist.gov/itl/ai-risk-management-framework	Generative AI risk profile released as NIST AI 600-1 in 2024.
OECD AI Principles https://www.oecd.org/en/topics/ai-principles.html	Human-centered, trustworthy AI principles.
Stanford AI Index 2026 https://hai.stanford.edu/ai-index	AI adoption and responsible AI macro context.
Anthropic Model Context Protocol docs https://docs.anthropic.com/en/docs/mcp	Open protocol pattern for connecting AI assistants to external systems.
Anthropic MCP announcement https://www.anthropic.com/news/model-context-protocol	MCP origin and purpose.
Lewis et al. 2020 RAG paper https://arxiv.org/abs/2005.11401	Retrieval-augmented generation for knowledge-intensive tasks.
Microsoft Copilot Studio https://www.microsoft.com/en-us/microsoft-copilot/microsoft-copilot-studio/	Low-code agent creation, governance, and ROI tracking.
Microsoft Copilot Studio 2026 release wave https://learn.microsoft.com/en-us/power-platform/release-plan/2026wave1/microsoft-copilot-studio/planned-features	2026 planned features including SharePoint/code interpreter and connector updates.
Google AppSheet Gemini help https://support.google.com/appsheet/answer/16106353?hl=en	Natural-language app creation through Gemini in AppSheet.
Notion Enterprise Search https://www.notion.com/help/enterprise-search	Permission-aware search across workspace, connected tools, and web.
Airtable AI automation action https://support.airtable.com/docs/airtable-automation-actions-generate-with-ai	AI generation in automations, updated 2026.
Airtable AI fields https://support.airtable.com/docs/using-airtable-ai-in-fields	AI fields and field-agent style use cases.
Make AI Agents https://www.make.com/en/blog/announcing-next-generation-make-ai-agents	Visual automation and shareable AI agents across apps.
Zapier Agents help https://help.zapier.com/hc/en-us/articles/24393442652557-Build-an-agent-in-Zapier-Agents	Creating AI agents across Zapier integrations.
n8n AI workflow tutorial https://docs.n8n.io/advanced-ai/intro-tutorial/	AI workflow nodes, chat model, and agent patterns in n8n.
ISO quality management https://www.iso.org/quality-management	Process approach and continual improvement.
Theory of Constraints Institute https://www.tocinstitute.org/theory-of-constraints	Constraint-based process improvement.

Source	Use in this edition
Donella Meadows Project leverage points https://donellameadows.org/a-visual-approach-to-leverage-points/	Systems leverage point framing.
Roediger and Karpicke testing effect https://journals.sagepub.com/doi/10.1111/j.1745-6916.2006.00012.x	Retrieval practice and testing effect.
Sweller cognitive load theory https://www.instructionaldesign.org/theories/cognitive-load/	Cognitive load principles for instructional design.
Ericsson deliberate practice citation https://www.scirp.org/reference/referencespapers?referenceid=1602879	Deliberate practice reference and DOI.

Ethical publishing note: strong claims should be tied to observable mechanisms, real examples, transparent limitations, and user outcomes. Never sell guaranteed income, guaranteed health outcomes, or guaranteed AI accuracy.