

90-Day AI Workflow Pilot

Scope, economics, controls and rollout



HOW TO USE THIS RESOURCE

90-Day AI Workflow Pilot

Marketing operations leaders and cross-functional pilot owners

Your practical outputs

- Signed pilot charter and weekly plan
- Comparable baseline and time ledger
- Quality gates and incident procedure
- Evidence-based day-90 release decision

Read, choose, apply

Start with the overview and decision rules. Choose a concrete constraint in your business. Work through the relevant templates using your own evidence, record open assumptions and agree a review with the accountable people. The complete online edition provides linked sources and contextual navigation.

Editorial context

Explicitly fictional cases and example figures are working aids, not client results or universal benchmarks. Frameworks structure decisions; they are not scientifically validated scores. Platform capabilities may change. Sources and review context are disclosed.

Read online: [90-Day AI Workflow Pilot](#)

CONTENTS

1. Use ninety days as a learning container	4
2. Days 1–15: write a falsifiable charter	4
3. Build a baseline that survives scrutiny	5
4. Select the smallest informative experiment	6
5. Build the pilot team and its decision rights	6
6. Calculate pilot economics without selling certainty	7
7. Days 16–35: prototype the whole work path	8
8. Days 36–65: supervise and protect the comparison	8
9. Create a pilot-specific evaluation pack	9
10. Install controls at the pilot boundary	10
11. Days 66–90: price uncertainty and write the decision	10
12. Hand the pilot over as an operable service	11
13. Investigate pilot failures with a causal log	12
14. Use explicit gates instead of calendar momentum	12
Turn this resource into a team brief	15
Sources & further documentation	16

1. Use ninety days as a learning container

A ninety-day pilot is a management container, not evidence that a workflow needs exactly ninety days to prove value. The purpose is to resolve a bounded set of uncertainties while limiting permissions and investment. Start with a decision question: can a specific team produce a specific accepted output with acceptable quality and total effort under a controlled operating process? The answer may be “continue learning,” not simply yes or no.

The original framework here is SCOPE: Specify the work, Compare fairly, Observe failure, Price total effort, Expand conditionally. It links the pilot plan to an actual business decision. Each stage produces an artifact that the next owner can inspect. Avoid using the pilot as a disguised purchase justification after the technology choice has already been made.

Period	Main uncertainty	Required artifact	Permission
Days 1-15	What work belongs?	Charter and baseline plan	Offline only
Days 16-35	Can the process work?	Prototype and failure log	Shadow operation
Days 36-65	What does accepted work cost?	Supervised observations	Bounded pilot
Days 66-90	Is expansion justified?	Release packet	Decision required

These dates are planning suggestions, not guarantees. If source permissions remain unresolved, delay the relevant stage. If a hard stop occurs, the calendar does not override it. Define what the team will learn even if the workflow is rejected. Better source governance, a reliable manual baseline, and a clearer acceptance rubric can remain valuable outcomes without turning a failed automation into a success story.

2. Days 1-15: write a falsifiable charter

A useful charter states what would make the team abandon or redesign the pilot. “Explore AI productivity” cannot be falsified. “Compare accepted internal campaign briefs with the current manual route while preserving claim support and review ownership” can produce a clear decision. Include excluded work so participants do not demonstrate value by drifting into easier tasks.

Use the FICTIONAL B2B SaaS company NorthstarDesk as a worked example. The pilot transforms an approved product narrative into an internal campaign brief. It excludes live customer data, outbound sending, public publishing, price changes, and budget allocation. Every number in this guide is an assumption for planning, never a client result or promised outcome. The team plans observations across normal, incomplete, and contradictory source situations rather than selecting only polished inputs.

Charter worksheet

State the business question, eligible work item, required quality, current route, proposed assistance, source classes, sensitive-data exclusions, review owner, expected observation volume, and maximum investment. Name the person who can stop the pilot. Record unresolved assumptions separately from approved facts. Write the day-90 decision options now: stop, redesign, extend narrowly, or expand one dimension.

Interview the requester, producer, reviewer, and final user. Ask each to describe a good accepted output and a costly failure. Reconcile differences before collecting measurements. If the requester values speed but the reviewer values traceable claims, the pilot must measure both. Sponsor signature alone is insufficient when practitioners disagree about completion. The charter becomes usable when the people doing the work can identify an eligible case and explain where a doubtful case should go.

3. Build a baseline that survives scrutiny

Measure the existing process before showing participants the preferred prototype. Otherwise the comparison may mix changes in tools, instructions, source quality, and reviewer behaviour. Use the same task definition, acceptance rubric, source packet, and output destination wherever feasible. Document unavoidable differences rather than pretending the comparison is perfectly controlled.

A practical design pairs comparable cases and balances which route is attempted first. Reusing the exact same case may create learning effects: the second attempt benefits from knowledge gained during the first. Different cases create difficulty differences. Neither design is automatically superior. Describe the choice, inspect the case mix, and use the comparison as directional operating evidence unless the design supports stronger inference.

Time ledger

Record case ID, task segment, route, source-preparation minutes, production minutes, review minutes, correction minutes, coordination minutes, total active time, waiting time, acceptance decision, and reasons for rejection. Separate training and initial configuration from recurring work, but do not omit either from the economic assessment. Log interruptions consistently across both routes.

Ask reviewers to evaluate without knowing the route when practical. Perfect blinding may be impossible because style or formatting reveals the method; state that limitation. Keep failed and abandoned items in the ledger, including the manual work needed to rescue them. Compare medians, ranges, and the long tail rather than a single average. A workflow that helps ordinary cases but creates expensive exceptions needs a clear exception route. The baseline is complete when another colleague can reproduce the calculation and identify what was excluded. If that colleague cannot, fix measurement before interpreting the result.

4. Select the smallest informative experiment

Select a task that is frequent enough to observe and narrow enough to inspect. The first candidate should have accessible, approved inputs and an identifiable acceptance owner. If the business question depends on confidential data that is not yet authorized, choose a different learning task rather than quietly substituting a public demo and claiming it answers the original question.

Write a candidate comparison using the same criteria for each workflow. Include volume, variation, source readiness, review expertise, consequence of error, and manual reversibility. Do not maximize an abstract opportunity score. Look for an experiment whose result could change a concrete decision. A task may be commercially important yet unsuitable for a short pilot because its outcome appears only after a long sales cycle.

Selection conversation

Ask the sponsor what decision will change if the experiment succeeds. Ask the practitioner which work will enter the queue next week. Ask the reviewer whether a good output can be recognized without lengthy investigation. Ask the technical owner whether the route can be isolated from external actions. If one answer is missing, turn it into a prerequisite rather than an optimistic footnote.

Set a maximum number of concurrent work types. Mixing campaign briefs, account research, reporting and outreach makes a small pilot difficult to interpret. Keep rejected candidates in the record with reasons. This prevents the team from reviving an unsuitable candidate halfway through the pilot merely because another vendor demonstration looks impressive. The smallest informative experiment is one that resolves a decision-relevant uncertainty while leaving a clear account of what was never tested.

5. Build the pilot team and its decision rights

Give the pilot four explicit responsibilities: business sponsorship, everyday operation, domain review, and technical control. Add source and specialist owners when needed. The sponsor approves the learning question and investment, but does not override a domain reviewer simply because the results look promising. The operator records actual work rather than editing the ledger to support the business case.

Pilot decision	Owner	Required input	Recorded outcome
Admit a case	Operator	Eligibility rules	Included or excluded
Accept quality	Domain reviewer	Draft and evidence	Pass, edit, return
Change a version	Technical owner	Regression comparison	Version decision

Pilot decision	Owner	Required input	Recorded outcome
Fund next stage	Sponsor	Quality and cost packet	Conditional permission

At the start of each stage, confirm capacity and deputies. A pilot can produce more drafts than reviewers can responsibly inspect. Limit work in progress when the review queue grows, rather than encouraging superficial approval. Set aside time for failed cases, measurement, and retrospective analysis; these are part of the experiment.

Decision-right rehearsal

Ask what happens if the sponsor wants an external demo before the review process is ready. Ask who can refuse an ineligible source. Ask whether the operator can pause the system without waiting for a steering meeting. Write the answers as permissions in the charter. If authority depends on a friendly relationship between two individuals, the design is fragile. The pilot should reveal whether the organization can operate the workflow, not only whether one enthusiastic employee can make a demonstration work.

6. Calculate pilot economics without selling certainty

Use a ledger that separates setup, recurring operation, and learning overhead. Setup includes workflow design, source preparation, configuration, access checks, and training. Recurring operation includes production, review, correction, maintenance, tooling, and incident handling. Learning overhead may fall after the pilot, but do not remove it without a reasoned estimate of what remains.

FICTIONAL NorthstarDesk arithmetic: assume 80 accepted briefs in a comparison period. Manual work averages 75 minutes; assisted work averages 52 minutes including review and correction. At an assumed loaded rate of €55 per hour, the difference is approximately €1,687 of capacity value. Assume €350 tools and €500 recurring maintenance for the same period. Net recurring capacity value is approximately €837 before recovering an assumed €3,600 setup investment. This is not cash saved, revenue generated, or a forecast.

Planning sensitivity	Net recurring capacity value	Decision implication
Assisted time rises to 65 minutes	About –€117	Redesign review burden
Assisted time remains 52 minutes	About €837	Observe repeatability
Volume falls to 40 briefs	About –€7	Fixed cost dominates

Use actual invoices or measured usage when available and disclose estimates elsewhere. Calculate cost per accepted brief, including rejected attempts and manual rescues. If quality falls below the agreed floor, the cheap result is not comparable. Report uncertainty in volume, labour value, review time, and maintenance rather than displaying a precise return percentage. Ask the business owner where released time will go. Unused theoretical capacity is not automatically realized economic value.

7. Days 16-35: prototype the whole work path

Prototype the complete route from request to acceptance, not only the generation step. Begin with a manually assembled source packet and a read-only draft destination. This keeps the first experiment focused on whether the transformation and review process are useful. Introduce retrieval or additional tools only when an observed bottleneck justifies them.

Run ordinary cases first to establish a working path, then deliberately test missing sources, contradictory versions, an ambiguous audience, an out-of-scope request, and hostile instructions embedded in a document. A refusal or request for clarification is sometimes the desired result. Keep the failure log next to the prototype so the team can see the gap between an attractive demonstration and an operable service.

Prototype review questions

Can the reviewer trace each material claim? Can the requester tell what remains unresolved? Does the draft retain the product scope? Can the operator identify which configuration produced it? Does an unauthorized action remain blocked even when requested forcefully? Can the manual route continue when the tool is unavailable?

Use [OpenAI's agents overview](#) as a product reference if implementing that stack, but inspect the actual permissions in your own system. Architecture labels do not prove controls are present. Avoid switching models, prompts, retrieval settings, and output schemas simultaneously. A simple change log helps distinguish improvement from random variation or a new case mix. At the end of this stage, the deliverable is a repeatable route with known defects and a supervised observation plan. It is not a polished demo promoted directly into production.

8. Days 36-65: supervise and protect the comparison

Run the supervised pilot with the charter still visible. Admit only eligible work and retain a record of work that was excluded. A source outage, a reviewer absence, or a campaign deadline can tempt the team to make unofficial exceptions. Those exceptions change the meaning of the pilot. Pause, route manually, or amend the charter through a recorded decision.

Use a weekly case review to inspect accepted outputs and failures together. Reviewers should record correction reasons, not just a pass mark. "Wrong audience emphasis," "unsupported

feature,” “missing qualification,” and “unusable structure” require different repairs. Capture near misses where the reviewer prevented a bad output from reaching its destination. Near misses demonstrate the value and workload of the review layer; they are not evidence that the model alone was reliable.

Protect reviewer independence

Do not reward reviewers for higher pass rates. Give them enough time to inspect evidence and a safe way to challenge the pilot sponsor. Where possible, rotate a second qualified reviewer through difficult cases. Record disagreements before resolving them so the team can distinguish an unclear rubric from inconsistent model behaviour.

Keep a weekly view of arrivals, accepted items, manual fallbacks, queue age, correction effort, and changes to configuration. Look for rising review minutes even if generation time falls. Review the manual route as well: better instructions or source hygiene may improve it, reducing the apparent benefit of AI. That is useful learning. The pilot should choose the best operating process, not protect a predetermined technology story. Freeze the final candidate configuration before the end-of-pilot comparison, and reserve genuinely unseen cases for the final check.

9. Create a pilot-specific evaluation pack

The test pack should reflect the charter, not a generic language-model benchmark. For an internal campaign-brief pilot, test product scope, claim support, audience fit, open-question handling, and correct refusal. Add cases that intentionally lack a required source or contain a superseded narrative. Include language variants if both editions belong in the pilot boundary.

Freeze a compact holdout before choosing the preferred configuration. Keep training examples and tuning cases separate from that holdout. Repeatedly optimizing against the final test turns it into a development set. Record each case's source packet, expected disposition, critical claims, reviewer rationale, and permitted destination. A test without an expected decision is difficult to interpret.

Recommended scoring card

Use hard-failure flags for invented evidence, unauthorized information, or forbidden actions. Then score claim support, task completion, and usefulness independently. Add review minutes and material correction count. If an automated judge assists scoring, compare it with qualified human decisions and inspect disagreement. The judge is another measured component, not a source of unquestionable ground truth.

A sample pass rate describes this selected pack, version, and conditions. It does not estimate all future work automatically. Keep results segmented by language, source condition, and task difficulty. A strong average can conceal a consistently weak segment. For pilot decisions, combine the observed pattern with consequence and reversibility. No severe failure in a small sample is reassuring evidence about those cases, not proof that the failure is impossible. Release only the population and permissions actually supported by the experiment.

10. Install controls at the pilot boundary

A pilot is still a real information-processing activity. Restrict the environment to approved participants, approved source classes, and a draft destination. Keep outbound communication, public publishing, CRM updates, and spending disabled unless separately authorized and tested. Assistance in creating a brief does not imply authority to execute the campaign.

Use access tests with both privileged and restricted participants. Confirm that source restrictions are enforced by the system, not merely written in a prompt. Keep operational logs useful for reconstruction while minimizing sensitive content. Ask the technical owner to demonstrate what happens when a source contains instructions to reveal information or use an unapproved tool. [OWASP's prompt-injection discussion](#) is relevant background for this threat class.

Pilot control checklist

- Source admission includes owner, version, permitted use and review date.
- Participants know which inputs must never enter the pilot.
- Drafts are visibly provisional and cannot publish automatically.
- Review decisions and configuration changes are traceable.
- The manual fallback is available without reconstructing lost context.
- The stop channel reaches someone with authority during operating hours.

This checklist is operational guidance, not legal advice or a compliance claim. Resolve data-processing, contractual, and jurisdiction-specific questions with qualified owners. If a necessary approval is absent, exclude that data or pause the affected experiment. Replacing sensitive inputs with synthetic examples can support a narrower technical test, but disclose that it changes what the result proves. A clean synthetic demonstration does not validate performance or permissions on real customer information.

11. Days 66–90: price uncertainty and write the decision

Prepare the final packet before the last meeting. Include observed case mix, exclusions, quality by segment, total active time, waiting time, reviewer capacity, setup effort, recurring costs, incidents, unresolved defects, and the permission requested. Attach representative failures as well as successful examples. A deck of only good outputs cannot support an operating decision.

Replace planning assumptions with measured ranges wherever possible. If the sample is too small to estimate recurring review effort credibly, say so and request a narrow extension with a specific learning question. Do not convert uncertain time savings into revenue. A revenue claim requires a separate causal design and observation period; a workflow pilot usually cannot isolate it from campaign, market, sales, and product effects.

Decision memo template

State the decision in the first sentence. Then explain the eligible population, evidence, important uncertainty, accepted residual risks, named operational owner, funding, and next review. For a redesign, identify the causal defect and the limited experiment that could resolve it. For a stop, record why the manual route is preferable and what reusable artifacts remain.

A conditional release might allow one team to use internal drafts while keeping external actions disabled. An extension might focus on German source ambiguity rather than rerunning easy English tasks. An expansion request should name exactly one new dimension and the tests needed for it. At day ninety, the strongest result is a decision that matches the evidence. Continuing indefinitely without a decision is not caution; it is an unowned service with accumulating cost and unclear permissions.

12. Hand the pilot over as an operable service

A successful pilot does not automatically create an operating team. Before handover, identify who pays recurring costs, maintains sources, updates configurations, reviews outputs, handles incidents, and retires the workflow. A sponsor's enthusiasm is not a maintenance plan. The receiver should accept each responsibility explicitly and confirm capacity.

Create a compact runbook with the workflow contract, source register, configuration location, approved user group, review rubric, known defects, monitoring schedule, incident channel, and manual fallback. Include screenshots or examples only when they help the receiver make a real decision. Avoid a large archive that hides the current instructions among obsolete experiments.

Handover rehearsal

The receiving operator processes an eligible case from request to acceptance. The deputy then processes a missing-source case and demonstrates the pause procedure. The technical owner restores a previous approved configuration in a safe environment. The business owner identifies the next review date and explains what would cause shutdown. Record any gaps as handover blockers, with a person and due date.

After transfer, schedule an early operating review focused on actual work rather than the pilot's final score. Examine whether users are bypassing controls, whether review effort is sustainable, and whether source updates are reaching the system. A release can remain intentionally small. If the service loses its owner or quality floor, return to the manual route. The pilot's purpose was to make this decision possible, not to create an obligation to keep the technology alive. Preserve the measurement method and failure taxonomy even if the implementation is retired.

13. Investigate pilot failures with a causal log

Maintain a defect record that explains where the work first went wrong. Distinguish task-definition errors, source defects, retrieval failures, generation errors, review misses, and handover failures. A bad brief is an outcome, not a diagnosis. Inspect the source packet and the intermediate evidence before changing the prompt.

Failure class	Diagnostic question	Pilot response
Task mismatch	Was this case eligible?	Repair admission rule
Source defect	Was approved evidence adequate?	Repair source or decline
Generation defect	Did output contradict evidence?	Change and retest configuration
Review failure	Was a material defect accepted?	Repair rubric and capacity
Handover failure	Did an unaccepted draft travel onward?	Repair status and permissions

Capture the cost of investigation and rework. It belongs in the pilot economics even when the final output is good. Preserve a reproduction with case and version identifiers while restricting sensitive material. If a suspected disclosure or forbidden action occurs, stop the affected path and involve the designated incident owner. A better regenerated answer does not close the incident.

At the weekly review, choose repairs by consequence and recurrence rather than how easy they are to demonstrate. Verify the failed case, then a broader regression set. Keep unknown-cause defects open with a bounded permission. Do not claim that a prompt change fixed a source-access problem. The pilot's failure log should teach the receiving team where to look first and what symptoms require immediate escalation. It is part of the final service handover, not an embarrassing appendix to hide.

14. Use explicit gates instead of calendar momentum

Stage gates should define what may happen next, not merely celebrate completion. An offline prototype can proceed to shadow observation when the work path is reproducible and sources are approved. A supervised pilot requires a functioning reviewer, traceable decisions, access tests, and a manual fallback. Wider use requires acceptable quality and effort in the relevant segments plus an owner who accepts operation.

Gate decision worksheet

Record the requested permission, evidence examined, cases not covered, unresolved defects, economic range, operational owner, rollback trigger, and next review. The permitted outcomes are stop, redesign, extend the narrow experiment, or release one bounded dimension. Avoid an ambiguous “approved” that leaves users guessing whether external actions are now allowed.

Hard STOP conditions for this pilot design include unauthorized disclosure or action, absence of a review owner, inability to reconstruct material decisions, or loss of the manual route. A cost or quality miss can justify redesign, but the workflow should not expand while that miss remains unresolved. Record accepted residual uncertainty with a named owner and expiry date.

After day ninety, the pilot label should disappear. Either close the experiment or move it into a service with explicit ownership and budget. A continuing experiment needs a new, bounded question and a decision date. Do not use extension to avoid an uncomfortable result. Preserve the charter, measurement method, case mix, and failure taxonomy so a future team can understand why the decision was reasonable at the time. That record remains useful even when tools and models later change.

Weekly uncertainty ledger

Keep a short ledger that names what remains unknown rather than filling the weekly review with confident status colours. For each uncertainty, record why it matters to the day-90 decision, the next observation that could reduce it, the owner, and the cost of learning. Distinguish “not yet observed” from “observed and acceptable.” If no difficult German brief has entered the pilot, its review burden is unknown, not zero.

At NorthstarDesk, a fictional weekly entry might note that ordinary briefs are usable but source-conflict cases still need specialist review. The next experiment should target that conflict class, not produce more easy examples. Keep its results separate from routine throughput. If the learning cost exceeds the plausible benefit of extending coverage, the sponsor can deliberately exclude the difficult class and retain a manual route. Record that narrowed population in the final decision. This is a valid scope decision when the exclusion is visible, operationally enforceable, and acceptable to the people receiving the work.

Primary references and limits

Sources checked 24 October 2026. Technical documentation changes; verify implementation details before release. Frameworks, worksheets and fictional assumptions are original operating proposals, not vendor guarantees.

- [NIST AI Risk Management Framework](#)
- [NIST Generative AI Profile](#)
- [OpenAI: Working with evals](#)
- [OpenAI: Agents](#)
- [Anthropic: Building effective agents](#)

- OWASP: Prompt injection
- Lewis et al.: Retrieval-Augmented Generation
- OpenAI: File search

YOUR TRANSFER CANVAS

Turn this resource into a team brief

Use this page for a joint review. Agree one practical output, observable evidence and an accountable owner. Add the specific methods and templates from the relevant chapters.

Working field	Your agreement
Priority output	Signed pilot charter and weekly plan
Your starting point	Finding / source / date: _____
Smallest useful test	_____ _____
Evidence & counter-evidence	_____ _____
Ownership & review	Owner / reviewer / date: _____
Stop or change when ...	_____ _____
Decision & next action	_____ _____

Keep unresolved assumptions explicitly open. A completed worksheet is not a substitute for evidence or accountable judgement.

Sources & further documentation

Primary sources checked 24 October 2026. Sources support the linked technical or factual statements; the working models and fictional cases are original editorial decision aids. Recheck changing platform details before implementation.

1. [NIST AI Risk Management Framework](#)
2. [NIST Generative AI Profile](#)
3. [OpenAI: Working with evals](#)
4. [OpenAI: Agents](#)
5. [Anthropic: Building effective agents](#)
6. [OWASP: Prompt injection](#)
7. [Lewis et al.: Retrieval-Augmented Generation](#)
8. [OpenAI: File search](#)

Next step

If you want help adapting the working plan to your product, evidence and team capacity, request a non-binding strategy conversation.

patrickwings.com