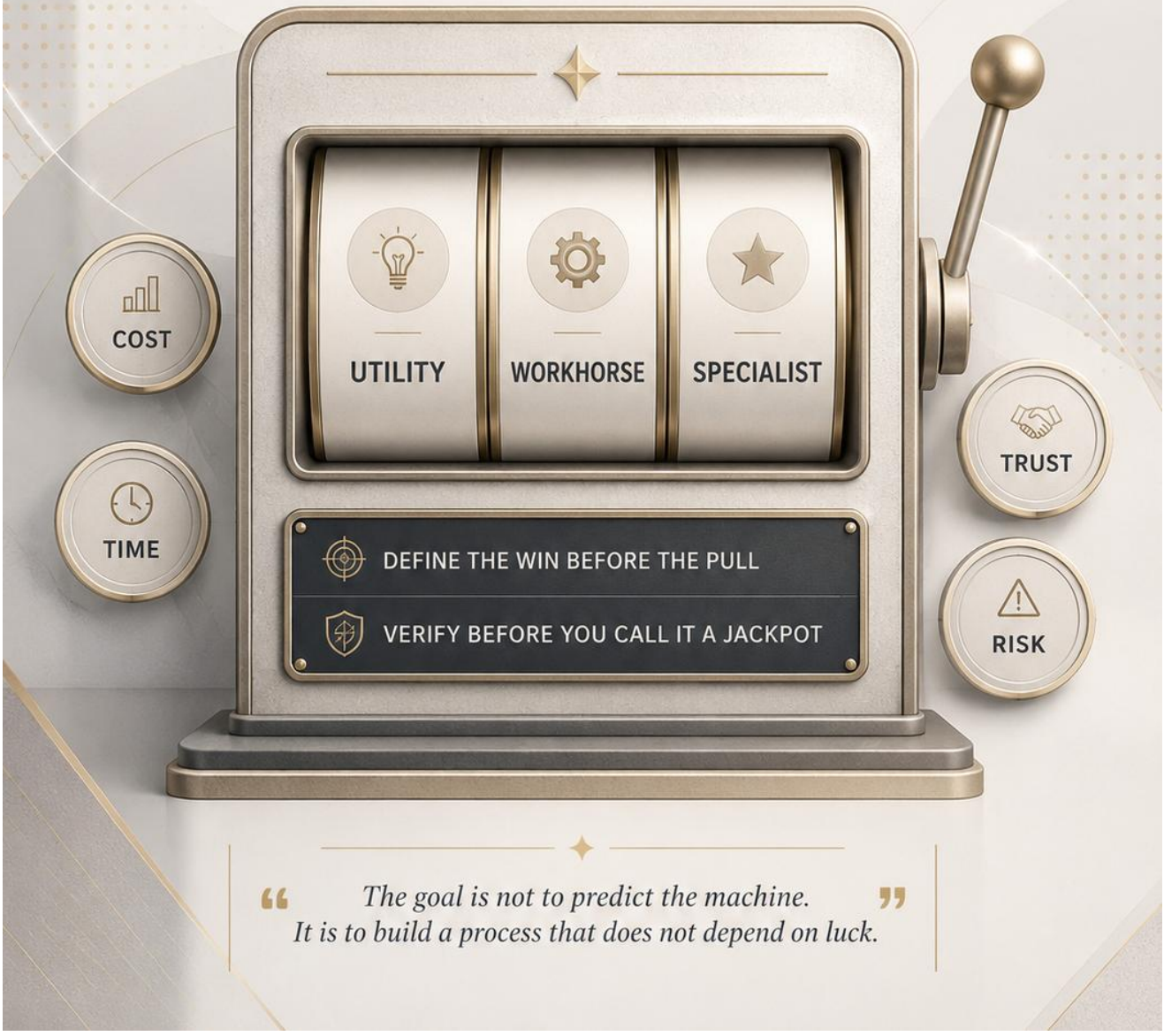
The Knowledge Casino Framework

A practical method for choosing, testing and governing AI models without endlessly pulling the handle.



> *The goal is not to predict the machine.*
> *It is to build a process that does not depend on luck.*

Framework version 1.0 | 17 August 2026

Companion framework to The Knowledge Casino research paper

1. Purpose of the framework

The Knowledge Casino Framework converts a playful analogy into a practical discipline for selecting and using generative AI models. Its purpose is simple: reduce accidental model hopping, reduce needless retries, make cost visible, improve repeatability and force the user to decide what a winning answer looks like before asking the machine to produce one.

Core proposition: AI use becomes less casino-like when model choice is based on task fit, the acceptance criteria are defined in advance, the conditions of the test are held steady and verification rises with consequence.

The framework does not assume that AI output is pure randomness. Generative models are steerable probabilistic systems. Good prompts, useful context, appropriate tools, fixed versions and evidence checks can narrow the range considerably. The casino metaphor is used to expose uncertainty, novelty chasing and repeated regeneration, not to claim that AI use is psychologically equivalent to gambling.

Who should use it

- Individuals who regularly switch between several AI models and want a reliable default rather than a favourite-of-the-week.
- Teams that need to explain why different models are approved for different types of work.
- People comparing model quality against price, speed, checking effort and consistency.
- Anyone who has found themselves pressing Regenerate several times because each answer is almost right.
- Organisations moving from experimentation into repeatable production workflows.

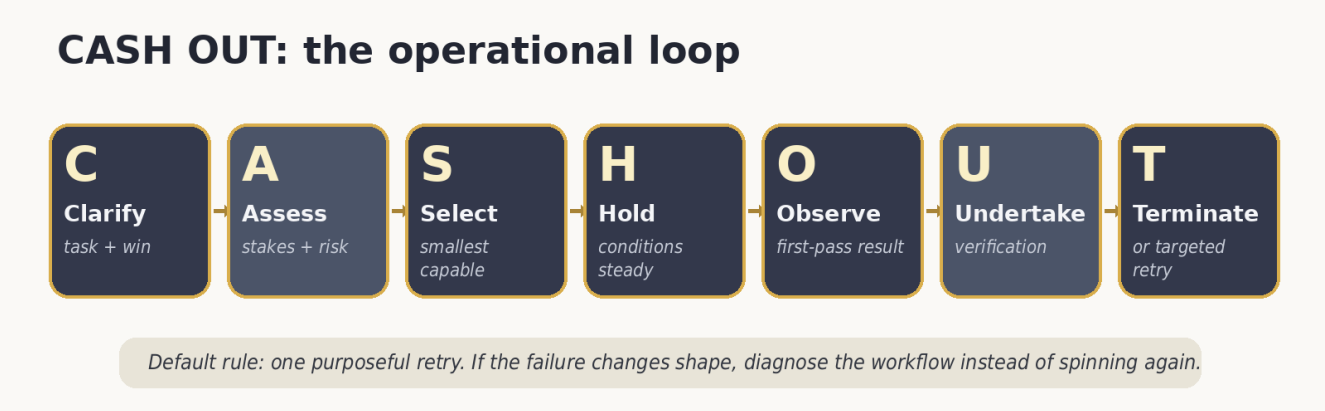
The outcome

At the end of the process you should be able to state: which model is the default for this class of task, what makes an output acceptable, what must be verified, how many retries are permitted, what the effective task cost is and what would justify testing a new model.

2. The framework at a glance

The framework has three layers. The first is behavioural, the second is operational and the third is governance. Together they turn the model picker from a wall of flashing lights into a small, explicit decision system.

Layer	Question	Practical output
House Rules	What behaviours keep us out of the regeneration loop?	Eight simple rules for model use.
CASH OUT loop	What should happen on each task?	A seven-step operational decision.
Control system	How do we keep a good workflow good?	Metrics, model portfolio, version control and review.



The three decisions that matter most

1. Define the win. Decide what acceptable means before generation, not after seeing several alternatives.
2. Choose the smallest capable machine. Select the least costly model that reliably clears the quality and risk threshold for the task.
3. Cash out. Once the acceptance criteria are met and required verification is complete, stop generating alternatives.

The anti-jackpot principle: Optimise for reliable first-pass acceptance, not for the most impressive answer you have ever seen from a model.

3. The eight House Rules

These are the behavioural guardrails. They are deliberately memorable because most model-selection problems do not begin with a benchmark. They begin with a human being staring at a menu.

1	**Define a win before you play. Write down the minimum acceptable outcome before the model sees the task.**
2	**Do not confuse fluency with a payout. A polished answer is not automatically a correct, complete or useful answer.**
3	**New is not a task requirement. A new model earns production use by beating the current workflow on agreed measures.**
4	**Use the smallest capable model. Cheap can be wasteful, but maximum capability can be wasteful too.**
5	**One purposeful retry. Fix a diagnosed problem once. If the second failure changes shape, stop spinning and redesign the workflow.**
6	**Free spins are not free. Subscription access can hide the cost of time, attention, comparison and verification.**
7	**Separate exploration from production. Exploration can chase surprise. Production should chase repeatability.**
8	**Pin what works and re-test what changes. A moving alias or new model version is a change event, not a cosmetic update.**

4. CASH OUT: the operating loop

CASH OUT is the core working method. It can be used for a single prompt, a recurring personal workflow or a team process. The language is intentionally casino-flavoured; the decisions are deliberately sober.

Step	Meaning	Decision question	Output
C	Clarify the task	What exactly is the job, and what would count as a win?	Task Passport + acceptance criteria
A	Assess the stakes	How costly would a wrong, incomplete or inconsistent answer be?	Knowledge Risk Score
S	Select the machine	Which approved model is the smallest one likely to clear the bar?	Model Portfolio + Fit Score
H	Hold conditions steady	Are prompt, context, tools and version controlled enough for a fair comparison?	Controlled Pull Record
O	Observe the result	Did the first output clear the bar without rescue work?	First-Pass Acceptance + repair minutes
U	Undertake verification	What must be checked before the answer is used?	Verification Gate
T	Terminate or retry	Should we accept, make one targeted retry, switch approach or stop?	Stop Rule + decision log

The rule behind the loop

CASH OUT is not designed to make every prompt slow. Low-stakes creative work may take seconds. High-consequence work may need evidence checks, fixed versions and a documented decision. The framework scales the control to the consequence.

Fast path: For low-risk work, run C, A and S mentally, then use H, O, U and T as a quick discipline. The framework should remove friction caused by indecision, not add bureaucracy for its own sake.

5. C: Clarify the task and define the win

The fastest way to turn a model picker into a slot machine is to begin without knowing what success looks like. The Task Passport forces the user to define the job before evaluating the answer.

Task Passport field	Prompt for the user	Example
Task	What is the concrete job?	Summarise a 40-page report into an 800-word briefing.
Audience	Who will use the output?	Senior UK managers with limited reading time.
Acceptance criteria	What must be present for the answer to pass?	Five key findings, three implications, no invented facts, plain English.
Evidence	What evidence must support it?	Claims traceable to the supplied report.
Variability tolerance	Would alternative wording be useful or harmful?	Low. Structure and findings should remain stable.
Format	What should the output look like?	Headings, short paragraphs, final action list.
Time limit	How much human time is reasonable?	Ten minutes including review.
Finish line	When do we stop?	When criteria are met and factual claims are checked.

Acceptance criteria test

A criterion is useful only if you can judge it. “Make it excellent” is not an acceptance criterion. “Include all five findings, keep each under 120 words and cite the relevant section” is.

- Observable: another person can tell whether it passed.
- Relevant: it measures the actual job, not general cleverness.
- Proportionate: a low-stakes brainstorm does not need an audit trail.
- Finite: it gives the user permission to stop.

Casino translation: The Task Passport writes the payout table before the reels start moving.

6. A: Assess the knowledge stakes

The required model and verification level should depend on consequence, not on how impressive the model name sounds. Use the Knowledge Risk Score to calibrate the workflow.

Factor	1: low	3: medium	5: high
Consequence	Minor inconvenience	Material rework or poor decision	Serious financial, legal, safety or customer impact
Verification difficulty	Easy to check immediately	Requires some expertise or source reading	Hard to independently confirm
Scale / reach	Private, one-off use	Shared with a team or repeated	Published, automated or used at scale
Reuse / persistence	Discarded after use	May inform later work	Becomes a template, policy, dataset or system input

Knowledge Risk Score: Average the four factor scores. 1.0 to 1.9 is Low, 2.0 to 2.9 is Guarded, 3.0 to 3.9 is High, 4.0 to 5.0 is Critical. The score does not replace judgement. It creates a consistent starting point.

Risk band	Model approach	Verification approach	Retry freedom
Low	Utility or workhorse model is usually enough.	User judgement or light checks.	Creative variation is acceptable.
Guarded	Workhorse by default. Escalate if evidence is weak.	Check key claims and calculations.	One targeted retry.
High	Workhorse or specialist chosen from evaluation evidence.	Source-level checks, tests or domain review.	One targeted retry, then diagnose.
Critical	Specialist only if workflow is approved and controllable.	Independent validation, human approval and documented evidence.	No open-ended regeneration.

7. S: Select the machine

Most people do not need to master every model. A small portfolio reduces decision fatigue and makes performance learnable over time.

Portfolio role	Use it for	What good looks like	Do not use merely because
Utility	Extraction, classification, rewriting, simple transformations, bulk tasks.	Fast, cheap, predictable, low repair effort.	It is the cheapest token price.
Workhorse	Most daily drafting, analysis, summarisation and mixed knowledge work.	Strong first-pass fit at sensible latency and cost.	It is the provider default.
Specialist	Complex reasoning, difficult coding, long-context synthesis, high-consequence work where tests show a gain.	Measurable quality or reliability improvement.	It is the flagship or newest model.
Second opinion	Deliberate challenge on selected high-risk tasks.	Independent critique or alternative reasoning path.	You simply want to see another answer.

The smallest capable model principle

Select the least expensive model that clears the acceptance and risk threshold with acceptable consistency. This is not “always choose cheap”. It is “do not pay for capability that the task cannot use”. The mirror image also matters: do not use a cheap model when repair, retries and verification erase the saving.

Model Fit Score

Score candidate models from 1 to 5 against your representative work. Convert each score to its weighted contribution. A perfect total is 100.

Criterion	Weight	What to measure
Task quality	30	Accuracy, completeness and reasoning against the defined acceptance criteria.
First-pass fit	20	How often the first response is usable without regeneration.
Verification burden	15	Human minutes and difficulty required before safe use.
Effective cost	15	Generation, retries, review and repair, not token price alone.
Consistency	10	Stability across repeated representative tasks.
Latency	10	Time to an acceptable result, not merely model response time.

Decision rule: A higher Fit Score does not automatically win. The model must also satisfy the risk controls. After that, prefer the lowest effective cost among models that clear the required threshold.

8. H: Hold conditions steady

A model comparison is meaningless if every pull changes the prompt, context, files or tools. The Controlled Pull makes comparisons fair enough to learn from.

What actually happens when you “pull the handle”?

Variability can enter at several layers, not just through a visible “temperature” setting. Newer model families may also change or remove those controls.

For a controlled comparison, record the factors below. You do not need laboratory-grade reproducibility for every task, but you do need enough control to know what changed.

Control	Record
Model	Provider, model name and fixed version where available.
Prompt	Same instructions and examples for each candidate.
Context	Same files, conversation state and source material.
Tools	Same search, code, retrieval and browsing permissions.
Output constraint	Same length, format and structure requirements.
Date	Useful because models, routing and provider defaults can change.
Evaluation	Same acceptance criteria and scoring method.

No cherry-picking: Do not compare the best result from Model A with the first result from Model B. Compare first passes, or use the same fixed number of trials for both.

9. O: Observe the result

The first response is operationally important. A model that occasionally produces a spectacular answer but regularly needs rescue can be worse than a quieter model that lands inside the acceptance zone almost every time.

Core measures

Measure	Formula / method	Why it matters
First-Pass Acceptance Rate	Accepted first attempts / representative tasks x 100	Measures reliability before retry behaviour distorts the picture.
Repair Minutes	Minutes spent editing an output into an acceptable state	Captures hidden human cost.
Regeneration Ratio	Total generations / accepted outputs	Shows whether the workflow is becoming a spin loop.
Verification Minutes	Minutes spent checking claims, sources, code or calculations	Makes trust cost visible.
Failure Severity	1 to 5 judgement of the worst plausible or observed failure	Prevents average quality from hiding dangerous edge cases.

A simple first-pass target

For recurring production work, set a target First-Pass Acceptance Rate. The exact number depends on the task, but the act of setting one changes the conversation. Instead of “Model X feels clever”, you can say “Model X passed 17 of 20 representative tasks without regeneration and required a median four minutes of checking”.

Jackpot warning: Do not score the most memorable output. Score the distribution of ordinary outcomes.

10. U: Undertake verification

A slot machine displays the payout. Generative AI often does not. Verification is therefore part of the product, not an optional afterthought.

Output type	Low-risk check	Higher-risk check
Factual prose	Sense-check and inspect obvious claims.	Trace important claims to primary or authoritative sources.
Summaries	Compare with source structure and key points.	Sample claims against the original material and check omissions.
Calculations	Recalculate simple figures.	Use deterministic calculation or an independently checked spreadsheet.
Code	Read and run basic tests.	Use automated tests, edge cases, review and controlled deployment.
Recommendations	Check assumptions and constraints.	Require evidence, alternatives, limitations and accountable human judgement.
Creative work	Check fit with brief.	Usually no factual verification unless claims are embedded in the work.

Verification Gate

4. Identify the claims or actions that matter if wrong.
5. Choose a check that does not depend on the same unverified output.
6. Record material failures, not just successful checks.
7. If verification is too expensive for the task, redesign the task or lower the model's authority.
8. Only treat the output as accepted after the required checks are complete.

House edge: The hidden cost in knowledge work is uncertainty about whether you have actually won. Verification is how you reveal the payout.

11. T: Terminate, retry or change the game

The stop rule is the part most likely to save time immediately. It breaks the habit of treating a near-miss as evidence that one more regeneration will certainly fix everything.

First result	Action
Passes acceptance criteria and verification	Accept. Cash out. Stop generating alternatives.
Fails for one clear, promptable reason	Make one targeted correction and retry once.
Fails because evidence or source material is missing	Add evidence or change the task. Do not simply regenerate.
Second attempt fails in a materially different way	Stop. Diagnose model fit, prompt design, context or task structure.
Output is high quality but high-risk verification fails	Reject or escalate. Fluency does not override evidence.
You are comparing alternatives only because another answer might be nicer	Return to the acceptance criteria. If it already passes, stop.

The one-retry rule

One retry is not a universal law. It is a useful default for routine knowledge work because it distinguishes correction from wandering. Creative exploration can permit multiple generations because diversity is the objective. Production workflows should not.

Near-miss trap: When every answer is 80 per cent right in a different way, the problem may no longer be the answer. The problem may be the task definition, evidence, prompt, model choice or workflow.

12. Cost: count the whole stake

Provider price is only the visible coin slot. The Knowledge Casino treats cost as the price of reaching a verified acceptable outcome.

Effective Task Cost: generation cost + retry cost + human review time + verification effort + repair work + expected failure cost

Cost component	Question to ask
Generation	What did the model invocation cost directly, if anything?
Retries	How many extra generations were needed to reach acceptance?
Review	How long did a person spend reading and comparing outputs?
Verification	How long did checking sources, calculations, tests or claims take?
Repair	How much editing or rework was needed?
Failure risk	What would a wrong answer cost if it escaped review?

A simple comparison

Model	Visible generation cost	Retries	Review + verification	Outcome
Model A: cheap	Very low	3	18 minutes	Accepted after repair
Model B: workhorse	Low	0	6 minutes	Accepted first pass
Model C: flagship	High	0	5 minutes	Accepted first pass

In this example, Model B may be the rational choice even though Model A is cheaper per token and Model C is more capable. The framework asks which route reaches a safe, acceptable answer at the lowest total cost.

13. The Shiny Machine Gate

New models deserve curiosity, not automatic promotion. The Shiny Machine Gate gives novelty a proper audition.

A new model should enter evaluation only when at least one trigger is present

- Capability trigger: the current portfolio cannot perform an important task well enough.
- Reliability trigger: first-pass acceptance or failure severity is below target.
- Cost trigger: a new model may materially reduce effective cost or latency.
- Change trigger: the provider is forcing migration or retiring the current version.
- Strategic trigger: a new tool, context capability or modality creates a genuinely new use case.

The audition

9. Run the same representative evaluation set used for the incumbent.
10. Score first-pass quality, consistency, verification burden, latency and effective cost.
11. Ignore launch-day excitement and isolated spectacular examples.
12. Promote the model only if it clears a pre-set improvement threshold or solves a missing capability.
13. Keep the previous workflow available until the replacement has passed regression checks.

Recommended promotion rule: Do not switch because the new model wins one task. Switch when it improves the portfolio on the measures that matter, without creating unacceptable new failure modes.

14. Model change control and drift

A workflow can change even when the prompt does not. Model aliases can move, providers can alter routing, tools can change and new versions can behave differently. Treat model change as a controlled event.

Event	Required action
New model version	Run representative regression tasks before production use.
Moving alias changes	Record the date and recheck critical workflows. Prefer fixed versions where available.
Provider deprecation notice	Plan migration, compare candidate replacements and preserve evidence from the old workflow.
Prompt template change	Re-run at least the tasks most sensitive to the altered instruction.
Tool or retrieval change	Test both answer quality and evidence traceability.
Unexpected performance drift	Pause open-ended retries and investigate what changed.

Change Drift Score

For a recurring evaluation set, record the percentage-point change in First-Pass Acceptance Rate after a model or workflow change. Pair it with the change in Verification Minutes and Failure Severity. A model that raises apparent quality but doubles checking time has not necessarily improved the workflow.

Production principle: Freeze the things that work until there is evidence that changing them is worthwhile.

15. Team and organisational governance

At team scale, the framework becomes a lightweight control system. The goal is not one universally approved model. It is an approved portfolio linked to task classes and risk.

Control	Minimum team practice
Model register	Approved model, role, fixed version where possible, owner and review date.
Task classes	Common use cases with a defined risk band and acceptance criteria.
Evaluation set	Representative tasks that can be rerun when models change.
Metrics	First-pass acceptance, verification minutes, effective cost and failure severity.
Change log	Record model, prompt, tool and version changes that affect production work.
Escalation	Define what happens when verification fails or risk is higher than expected.
Exploration sandbox	Give new models somewhere to be tested without silently replacing production behaviour.

What not to govern by

- Provider prestige alone.
- Benchmark headlines without representative internal tasks.
- The loudest internal advocate for a particular model.
- Token price without human time and verification cost.
- One spectacular demonstration.
- A single global model for every task merely because it simplifies procurement.

16. Worked scenarios

The same framework should produce different answers for different jobs. That is the point.

Scenario A: creative naming workshop

Task: generate 30 names for a fictional project. Risk is Low, variability is desirable and factual verification is irrelevant. Select the utility or workhorse model that gives useful diversity quickly. Multiple generations are acceptable because exploration is the objective. Cash out when the idea pool is large enough to curate.

Scenario B: weekly executive briefing

Task: summarise a fixed set of source documents into a repeatable briefing. Risk is Guarded to High. Variability tolerance is low. Use the workhorse that has the strongest first-pass acceptance on the internal evaluation set. Hold the template, sources and output structure steady. Verify key claims. Allow one targeted retry, then diagnose.

Scenario C: complex technical diagnosis

Task: investigate an intermittent technical fault. Risk is High and the cost of missing an edge case is material. The specialist may justify its higher price if tests show a measurable gain. Require explicit assumptions, run tests where possible and use a second opinion only as a deliberate challenge, not as an endless vote among models.

Scenario D: shiny new model launch

A new flagship arrives with impressive benchmarks. Nothing is currently broken. The Shiny Machine Gate says: explore it in the sandbox if curious, but do not change production. If it later shows a material improvement on the same evaluation set, promote it through change control.

Pattern: The framework does not ask “Which model is best?” It asks “Which model gives this task the best odds of a verified acceptable result at this level of cost and risk?”

17. One-page decision card

Use this when you do not need the full framework. If you can answer these ten questions, you are probably making a controlled model decision rather than pulling a random handle.

#	Question	Yes / No / Note
1	Can I state the task in one sentence?	
2	Have I defined the minimum acceptable output?	
3	Do I know the consequence if the answer is wrong?	
4	Have I chosen the smallest capable model from my portfolio?	
5	Am I keeping prompt, context and tools stable enough to judge the result?	
6	Did the first pass meet the acceptance criteria?	
7	Have I completed the verification appropriate to the risk?	
8	If I retried, was it one targeted correction rather than another spin?	
9	Am I counting review and verification time as part of cost?	
10	If the answer now passes, am I willing to stop?	

If question 10 is “no”: You may no longer be solving the task. You may be browsing possible answers.

18. Reusable worksheet: Task Passport

Copy this page for recurring task types. A completed passport should be short enough to use, but precise enough to judge the result.

Field	Your entry
Task	
Audience	
Acceptance criteria	
Required evidence	
Variability tolerance	
Required format	
Time budget	
Knowledge Risk Score	
Finish line	

Pre-pull note

Write this sentence: "I will accept the output when _____."

19. Reusable worksheet: Model bake-off

Use five to twenty representative tasks depending on the importance and frequency of the workflow. Keep conditions as steady as practical.

Candidate	Task quality /5	First-pass /5	Verify burden /5	Effective cost /5	Consistency /5	Latency /5	Weighted /100
Model A							
Model B							
Model C							
Model D							

Weights: Task quality 30, First-pass fit 20, Verification burden 15, Effective cost 15, Consistency 10, Latency 10.

Decision

Question	Record
Which models clear the minimum quality and risk threshold?	
Which of those has the lowest effective task cost?	
What failure mode would prevent promotion?	
What model becomes the default, specialist or utility choice?	
When will this decision be reviewed?	

20. Reusable worksheet: Controlled Pull Log

This is useful when a result matters enough that you want to understand why it changed.

Field	Record
Date / task ID	
Model and version	
Prompt template version	
Files / context	
Tools enabled	
First-pass accepted?	
Repair minutes	
Verification minutes	
Retry used? Why?	
Final decision	
Failure or learning to retain	

21. Reusable worksheet: Shiny Machine Gate

Before adding a newly launched model to production, make the novelty pay rent.

Gate	Pass condition	Result
Reason to test	A real capability, reliability, cost, migration or strategic trigger exists.	
Comparable test	The incumbent and candidate are tested on the same representative tasks.	
Quality	Candidate clears the required acceptance threshold.	
Cost	Effective cost is acceptable after review and verification.	
Risk	No new failure mode breaches the task risk controls.	
Consistency	Performance is stable enough for the intended use.	
Change control	Rollback or previous workflow remains available during transition.	
Promotion decision	Evidence justifies a portfolio role, not simply curiosity.	

22. Closing principle: know when to cash out

The Knowledge Casino is not an argument against model choice. Choice is useful. Nor is it an argument against randomness. Variability can be an extraordinary creative resource. The framework is an argument against unmanaged uncertainty being mistaken for progress.

When a task is exploratory, enjoy the floor. Try models, compare styles, chase surprise and see what happens. When a task becomes production work, change the rules. Define the win. Match the model to the stakes. Hold the conditions steady. Measure the first pass. Verify what matters. Stop when the job is done.

The framework in one sentence: Choose the smallest capable model, define success before generation, verify in proportion to consequence and stop pulling the handle once the output has genuinely won.

Relationship to the companion research paper

This framework is designed as the practical companion to The Knowledge Casino: Why choosing an AI model can feel like pulling a slot-machine handle, research snapshot 17 August 2026. The paper provides the deeper discussion of probabilistic output, model choice, pricing, retries, novelty effects, verification and the limits of the casino analogy. This document converts those ideas into a reusable operating method.

Work, Considered
Chris D Benson
17th August 2026
www.knowledge-casino.com