Decision-Making & Thinking Methods — When to Use Each

Elimination-based methods

- **Process of elimination** — Systematically ruling out options that fail a hard requirement until only viable ones remain. Use it when you have many options and at least one clear disqualifying criterion (budget cap, hard deadline, legal requirement) — it's fast because you never have to score the options you kill off. Bad fit when no option can be truly disqualified and everything is a matter of degree; then you need scoring, not elimination.
- **MECE exclusion** (Mutually Exclusive, Collectively Exhaustive) — Breaking a decision space into buckets that don't overlap and don't leave gaps, then eliminating whole buckets at once instead of individual items. Use it when the option space is large and unstructured (e.g. "how do we reduce costs") — forcing MECE buckets first prevents you from redundantly comparing options that are really the same idea in different words.
- **Veto voting** — Each participant can strike out options they find unacceptable, and whatever survives every veto wins. Use it in group decisions where avoiding a bad outcome matters more than optimizing for a great one — picking a restaurant for a group with dietary restrictions, or approving a shortlist where any single stakeholder can kill a candidate.

Branching / tree-based methods

- **Decision trees** — Mapping out sequential decision points as branches, with the outcomes and probabilities at each leaf. Use it when a decision unfolds in stages and later choices depend on earlier ones — should we invest now or wait for more data, given different ways the market could move. Overkill for one-shot decisions with no sequential dependency.
- **Backward induction** — Starting from the desired end state or last decision point and reasoning backward to figure out what has to be true at each earlier stage. Use it for

planning problems with a known target and multiple paths to it — game theory, negotiation prep, or working out the last responsible moment to make a call. Only works cleanly when the end state is well-defined; useless for open-ended exploration.
- **Minimax / worst-case branching** — At each decision branch, assume the opponent or environment will pick the outcome worst for you, and choose the branch that minimizes that worst case. Use it in adversarial or highly uncertain situations where you can't assign real probabilities — competitive strategy, security threat modeling, negotiating with an unpredictable counterparty. Too conservative for situations where you actually have decent probability estimates; expected-value methods do better there.

Comparative / scoring methods

- **Weighted decision matrix** — List options as rows, criteria as columns, assign a weight to each criterion, score each option per criterion, and sum. Use it when you have 3+ options, multiple criteria that matter to different degrees, and you want a defensible, auditable comparison — vendor selection, hiring between finalists, picking between architecture options. Don't bother for a 2-option call with one dominant criterion; that's just a comparison sentence.
- **Pairwise comparison** — Comparing every option against every other option one-on-one, then tallying wins, instead of scoring all of them on an abstract scale at once. Use it when abstract scoring feels arbitrary or when direct comparison is easier for people than absolute rating — ranking a shortlist of candidates, or prioritizing a backlog where "how much better is A than B" is easier to answer than "rate A from 1-10."
- **Pros/cons list** — The simplest form: two columns, upsides and downsides, no weighting. Use it for quick, low-stakes personal decisions where you mainly need to externalize your thinking, not calculate an optimum. Weak for anything with more than a couple of criteria of very different importance — a long pros list can outweigh one severe con it shouldn't.
- **Cost-benefit analysis** — Converting the expected costs and benefits of each option into comparable units (usually money or time) and comparing net values. Use it for decisions with quantifiable costs and benefits — capital investment, whether to build vs. buy, whether a project is worth the resourcing. Breaks down when the biggest factors are genuinely unquantifiable (reputation, morale); forcing a number onto those can mislead more than it helps.

Root-cause & diagnostic methods

- **Five Whys** — Asking "why" repeatedly (usually 5 times) on an observed problem until you reach the root cause rather than a symptom. Use it right after an incident or failure, when the visible problem is probably a downstream effect of something deeper — a production outage, a recurring team conflict, a process that keeps breaking. Weak when a single problem has multiple independent root causes; the format's single-chain structure

will make you stop at the first plausible-sounding answer instead of finding all of them.
- **Fishbone / Ishikawa diagram** — Mapping many possible contributing causes, grouped into categories (people, process, tools, environment), converging on one effect. Use it when a problem plausibly has several independent contributing factors and you want to explore breadth before committing to one root cause — quality defects, complex incident postmortems. More effort than Five Whys, but better suited to multi-causal problems.
- **First principles thinking** — Stripping a problem down to its fundamental truths and reasoning up from there, instead of reasoning by analogy to how it's normally done. Use it when the standard approach to a problem is expensive, slow, or just assumed rather than justified — redesigning a process everyone "always" does a certain way, or challenging an industry convention. Costly to do for routine decisions; reserve it for decisions where the payoff of a genuinely novel answer is high.

Framing & structuring methods

- **SWOT analysis** (Strengths, Weaknesses, Opportunities, Threats) — A four-quadrant framing that separates internal factors (strengths/weaknesses) from external ones (opportunities/threats). Use it early in strategic planning, when you need a shared, structured starting point for a discussion rather than an actual decision — it's a framing tool, not a decision procedure, so don't expect it to output an answer by itself.
- **Six Thinking Hats** — Deliberately switching between six modes of thinking (facts, feelings, caution, optimism, creativity, process) in turn, usually in a group. Use it when a group discussion keeps getting stuck because people default to one mode (usually caution or optimism) and talk past each other — it forces everyone to visit the same mode together instead of arguing across modes.
- **Cynefin framework** — Classifying a situation as simple, complicated, complex, or chaotic, and matching your decision approach to that classification (best practice, good practice, emergent practice, or rapid stabilizing action respectively). Use it as a first step before choosing any other method on this list — it tells you whether a decision even has a knowable right answer (simple/complicated) or whether you should be probing and adapting instead (complex/chaotic).

Iterative / adaptive methods

- **OODA loop** (Observe, Orient, Decide, Act) — A fast repeating cycle for decisions under time pressure and incomplete information, where you act, then immediately re-observe and adjust. Use it in fast-moving, high-uncertainty situations where waiting for full information means missing the window to act — incident response, competitive/adversarial situations, live operational decisions. Wrong tool for decisions where you have time to gather full information up front; the speed advantage isn't worth the lost rigor.

- **PDCA / iterative improvement** (Plan, Do, Check, Act) — Running small experiments in a loop, checking results, and adjusting before scaling up. Use it for process improvement where you can test cheaply before committing — a new workflow, a process change, an experimental feature. Not suited to one-shot, high-stakes, irreversible decisions where you don't get a second iteration.
- **Satisficing** — Choosing the first option that meets a "good enough" threshold, rather than searching for the optimum. Use it when the cost of continued search or analysis exceeds the expected gain from finding a marginally better option — routine purchasing decisions, hiring under time pressure, anything where "good enough, now" beats "perfect, eventually."

Group / consensus methods

- **Delphi method** — Anonymously collecting expert opinions in rounds, sharing an aggregated summary between rounds, and repeating until opinions converge. Use it for forecasting or expert judgment on questions with no hard data, where you want to avoid groupthink and status-driven bias — long-range technology forecasts, expert risk assessments. Slow; not for time-pressured decisions.
- **RAPID / RACI** — Frameworks that assign explicit decision roles (who Recommends, who Agrees, who Performs, who has Input, who Decides / who's Responsible, Accountable, Consulted, Informed). Use it when a decision keeps stalling because it's unclear who actually has the authority to decide, not because the content itself is unclear — organizational decisions with many stakeholders. Doesn't help at all with the substance of the decision, only the process of making it.
- **Dot voting / brainstorming convergence** — After open brainstorming generates many options, each participant places a limited number of votes (dots) on their preferred ones, and the highest-vote options move forward. Use it to converge a large, unstructured set of ideas from a group down to a shortlist quickly, without a full scoring exercise — workshop facilitation, feature prioritization sessions.

Heuristics / shortcuts

- **Pareto principle (80/20)** — Recognizing that a small fraction of causes/inputs typically drives most of the effect/output, and focusing effort there instead of treating all factors equally. Use it for prioritization when you have many candidate causes or actions and limited time — which bugs to fix first, which customers to focus retention efforts on.
- **Occam's razor** — Among competing explanations that fit the evidence equally well, preferring the simplest one. Use it during diagnosis or root-cause analysis when multiple explanations are consistent with what you've observed — don't reach for an exotic explanation when a mundane one accounts for the same facts.
- **Premortem** — Before committing to a plan, imagining it has already failed and working backward to figure out why. Use it right before locking in a significant decision, as a check

against overconfidence and groupthink — it surfaces risks that "will this work" framing tends to suppress because everyone's already invested in saying yes.
- **Regret minimization** — Choosing the option that minimizes your future regret if it turns out badly, rather than the option with the best expected value. Use it for personal, high-stakes, largely irreversible decisions where the emotional cost of "what if" outweighs a purely numerical optimum — career changes, major life decisions.

Revision #1
Created 8 September 2026 15:09:58 by N8N
Updated 8 September 2026 15:09:58 by N8N