

OpsPilot

Option Analysis / MCDA — User Manual

Defensible Structured Decisions · Kepner-Tregoe / ISO 31010 · AI Engineering Co-Pilot



AI-GENERATED CONTENT · INDEPENDENT VERIFICATION REQUIRED

This manual was produced with AI assistance and is only as good as the information it was given. Every statement, figure, standard reference and conclusion must be independently verified by a competent, suitably qualified person before it is relied upon. It is a draft aid to your judgement — not a finished, authoritative, or certifying document. Professional and legal responsibility for any reliance rests with you and your organisation. See the full Engineering Disclaimer at opsinnovatech.com/engineering-disclaimer.

What this guide covers — what option analysis is, how the OpsPilot module structures a defensible decision, what to have ready, and the output you receive.

1. What is option analysis?

Option analysis (Multi-Criteria Decision Analysis) is how you make a defensible choice between options when there's no single obvious winner — by naming the criteria, justifying their weights, scoring each option repeatably, and testing whether the answer holds up. Its real job is protecting the decision from the three things that quietly corrupt it: confirmation bias, the sponsor's favoured option, and decision-by-loudest-voice. A scored, weighted, sensitivity-tested matrix is far harder to hijack than a meeting.

OpsPilot structures it per *Kepner-Tregoe Decision Analysis, ISO 31010 §B.7 (MCDA)* and — where complexity demands — *Saaty's AHP*, separating must-have from want-have criteria.

2. What the OpsPilot module does

Role	Responsibility
AI Coach — Decision Analyst (OpsPilot)	Structures the must-have / want-have criteria split, the weighted scoring matrix, sensitivity analysis and trade-off documentation — forcing criteria to be named, weights justified, scoring repeatable, and dissent recorded.
Decision-Maker / Sponsor / Participants (you)	Provide the decision context, the options, the criteria and the scoring inputs. You own the final decision; OpsPilot structures the analysis but cannot make the call.

3. How it works

- Separate must-have criteria (pass/fail screens) from want-have criteria (weighted).
- Screen options against the must-haves — any that fail are out, regardless of other merits.
- Weight the want-have criteria, with the weighting justified.
- Score each surviving option against each criterion, repeatably.
- Run sensitivity analysis — does the winner change if the weights shift?

4. What you will be asked — have this ready

- The decision and the options on the table.
- The criteria — what genuinely matters — split into must-haves and want-haves.
- The relative importance (weights) of the want-have criteria.
- The information to score each option.

5. What you receive — the output

A complete Option Analysis (Word): options screened against must-haves, scored against weighted want-have criteria, sensitivity-tested, with the trade-offs made explicit and any dissent recorded — a decision that's defensible to a board or a regulator.

6. Worked example (illustrative)

Choosing between three pump suppliers. A must-have screen (“must meet the rated duty and be deliverable within 16 weeks”) eliminates one outright — no amount of low price rescues an option that fails a must-have. The remaining two are scored against weighted want-haves (life-cycle cost, reliability track record, service support, lead time). The sponsor quietly favours the cheaper one, but the matrix shows the other wins on weighted score because reliability and service carry more weight than upfront price. The sensitivity analysis then asks the honest question: how much would the weights have to change to flip the result? If a small, reasonable shift flips it, the decision is close and that's documented; if only an implausible shift flips it, the winner is robust. Either way the choice is reasoned and recorded — not the loudest voice in the room.

7. Getting the best result

- **Use must-haves as hard screens.** An option that fails a must-have is out, however attractive otherwise.
- **Justify the weights.** Unjustified weights are where bias hides — make them explicit.
- **Run the sensitivity test.** Knowing how robust the winner is matters as much as who won.
- **Record dissent.** A defensible decision shows the disagreement was heard, not buried.

OpsPilot — AI Engineering Co-Pilot. Learn more at opsinnovatech.com