

MECE

Issue Trees

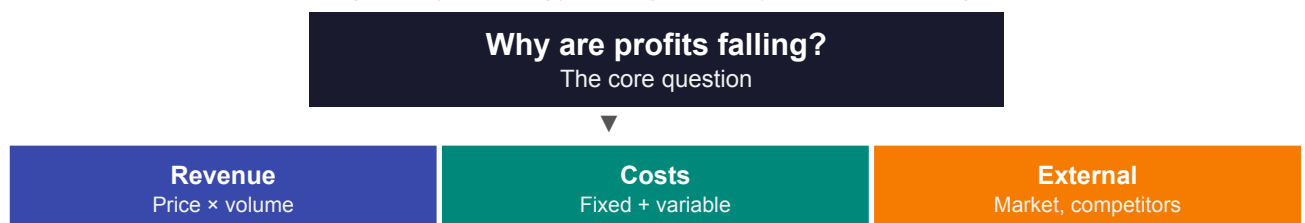
A way to break a big, messy problem into a clear tree of smaller parts. 'MECE' means the branches are Mutually Exclusive (no overlaps) and Collectively Exhaustive (nothing missing). Popularised at McKinsey, an issue tree turns a vague question into a structured map you can actually solve, piece by piece.

Best for: Problem structuring · Root-cause analysis · Consulting cases · Breaking down anything complex

01 — AT A GLANCE

Start with one core question at the top, then split it into a few non-overlapping branches, and split those again until each piece is small enough to tackle.

Each branch is separate (no overlap) and together they cover the whole problem.



Profit = Revenue – Costs, so the very first split is already MECE.

02 — THE CORE COMPONENTS

Four ideas make an issue tree work.

The Core Question	One sharp, specific question at the top, e.g. 'Why has profit dropped 20%?' Everything below exists to answer it.
Mutually Exclusive (ME)	Branches must not overlap. 'Revenue' and 'Costs' are clean; 'Revenue' and 'Online sales' are not, because they double-count.
Collectively Exhaustive (CE)	The branches together must cover everything, with no gaps. If a real cause could sit outside all your branches, the tree is incomplete.
Logical Layers	Each level breaks the one above into smaller drivers, often using a formula (Profit = Revenue – Cost) or natural categories, until each leaf is testable.

The MECE test

Before you go deeper, check every split twice: could any item belong to two branches (not mutually exclusive)? Could a real cause fall outside all of them (not collectively exhaustive)? A clean split prevents double-counting and blind spots, the two ways analysis quietly goes wrong.

03 — WHEN TO USE IT & WHEN NOT TO

This framework isn't right for every situation. Used at the wrong moment, it can point you in the wrong direction.

✓ USE IT WHEN...	✗ DON'T USE WHEN...
• Facing a broad, fuzzy problem you must structure • Diagnosing root causes methodically • Splitting work cleanly across a team • Preparing a consulting-style case or analysis	• Tiny, obvious problems with one clear cause • When speed matters far more than completeness • Forcing a simple 80/20 problem into endless branches • When the core question itself is still undefined

04 — IN ACTION

DostCafé — Diagnosing a Profit Drop

DostCafé, a 20-outlet coffee chain, sees profit fall 18% in a year. The founder is tempted to blame 'the economy' and cut staff. An issue tree forces a cleaner look before any cuts.

How the framework was applied:

Step 1	Set the core question — 'Why did annual profit fall 18%?' goes at the top of the tree.
Step 2	First MECE split — Profit = Revenue – Costs, so the tree splits into 'Revenue down' and 'Costs up' (no overlap, fully covers profit).
Step 3	Branch the revenue side — Revenue = customers × average bill. Footfall is flat, but the average bill has dropped, so the issue is pricing and mix, not traffic.
Step 4	Branch the cost side — Costs split into rent, ingredients and labour. Ingredient cost per cup has jumped 22% while rent and labour are steady.
Step 5	Find the leaf — The real drivers: milk and bean prices rose, and a discount combo quietly lowered the average bill. Two specific leaves, not 'the economy'.

The numbers at a glance

Profit –18% the question	First split Rev – Cost MECE	Real driver Inputs +22% cost per cup	Also Lower bill discount mix
--	---	--	--

Why the tree beat a guess

'The economy' would have led to layoffs that fixed nothing. The issue tree isolated two concrete, fixable leaves, input costs and a margin-eroding combo, so the founder could renegotiate supply and redesign the offer instead of cutting staff.

05 — KEEP IN MIND

Even experienced teams slip up here. Keep these in mind to keep your analysis honest and useful.

1. Overlapping branches that double-count the same cause.
2. Gaps that let the real driver hide outside the tree.
3. Splitting endlessly instead of stopping at testable leaves.
4. Mixing different levels of idea inside one set of branches.
5. Building the tree before the core question is sharp.