SILICON MEETS CODE · TECHNICAL PAPER TP-01 · NOVEMBER 2026

# Using Cause-and-Effect Diagrams in Research and Development

**Henrik Andersen**
MSc in Electrical Engineering · Independent electronic engineering consultant
Silicon Meets Code · siliconmeetscode.com · info@siliconmeetscode.com

**Abstract**

Cause-and-effect diagrams, also known as fishbone or Ishikawa diagrams, provide a structured way to generate and organise candidate explanations for an observed problem. In research and development (R&D), their value lies in widening a team's search space and making assumptions visible; the diagram itself neither establishes causality nor proves a root cause. This paper presents an evidence-led workflow for using the method in engineering investigations, covering problem definition, category selection, cause generation, prioritisation, verification, corrective action and recurrence monitoring. A worked microcontroller example shows how a symptom such as an unexpected halt can be decomposed into testable hypotheses involving power integrity, reset sources, clocking, software faults and the test set-up. The method is most useful when it is combined with engineering records, measurements, controlled reproduction and appropriate analytical tools. The paper is a practice-oriented synthesis and reports no new experimental results.

**Keywords:** cause-and-effect diagram; Ishikawa diagram; root cause analysis; hypothesis testing; embedded systems; corrective action

**Cite as:** Andersen, H. (2026) *Using cause-and-effect diagrams in research and development*. Silicon Meets Code Technical Paper TP-01.
Available at: https://siliconmeetscode.com/downloads.html

## 1 Introduction

Engineering investigations often begin with an observed effect and incomplete evidence. A product may reset in the field, a laboratory result may vary between runs, or a prototype may fail to meet a performance requirement. A cause-and-effect diagram gives a team a shared structure for listing and organising plausible causes before it commits to a test or a corrective action. The method is commonly called a fishbone diagram because the completed diagram resembles a fish skeleton, and an Ishikawa diagram after Kaoru Ishikawa (American Society for Quality [ASQ], no date a; Ishikawa, 1986).

The diagram is a tool for generating and organising causes. It can support root cause analysis, but it does not by itself identify or validate a cause: a branch records a hypothesis, not a finding. The distinction matters in R&D, where interacting hardware, firmware, requirements, manufacturing, test and operating conditions can produce similar symptoms. A defensible conclusion requires evidence that discriminates between the hypotheses and a corrective action whose effectiveness has been verified.

This paper sets out a practical workflow for applying the diagram to engineering problems. The workflow adapts the familiar categories to the system under investigation and connects brainstorming to objective evidence, controlled tests, corrective action and follow-up. The worked example is illustrative: it does not describe a measured product failure or claim a particular cause.

## 2 Scope and method

This is a practice-oriented technical synthesis based on established descriptions of cause-and-effect diagrams and of engineering root cause analysis. It reports no new experiments, field data or quantified comparison of investigation methods. Statements about process benefits are therefore framed as intended uses rather than as measured improvements in performance. The example is a procedural demonstration, not a case study.

The terms *candidate cause* and *hypothesis* are used for explanations entered in the diagram. A *contributing factor* is a condition that influences the event but may not be sufficient to cause it. *Root cause* is reserved for a cause that the investigation supports and that is linked to an effective action preventing, or materially reducing, recurrence. Several root causes typically contribute to an undesired outcome (National Aeronautics and Space Administration [NASA], no date), so the analysis should not force a single-cause conclusion when the evidence does not support one.

## 3 What the diagram does and does not establish

A cause-and-effect diagram starts with a defined effect, usually placed at the head of a spine that points to the right. Major branches represent categories; smaller branches represent candidate causes and their sub-causes. The team uses the structure to ask which conditions could produce the observed effect, to capture ideas in a visible form and to identify where further investigation is warranted. Categories are prompts, not a mandatory taxonomy. The familiar manufacturing categories can be useful, but an R&D team should choose categories that reflect the system boundary and the failure under investigation (ASQ, no date a).

The diagram does not establish that a listed factor occurred, that it preceded the failure or that changing it would prevent recurrence. It does not quantify causal strength, probability or risk, and the visual placement of an entry does not imply a proven causal chain. Prioritisation must therefore lead to the collection of evidence and to testing (ASQ, no date b). Guidance from NASA's Goddard Space Flight Center draws the same distinction: it lists the fishbone diagram as one logic-flow tool and describes the subsequent investigation as confirming or rejecting a prioritised list of possible explanations (National Aeronautics and Space Administration Goddard Space Flight Center [NASA GSFC], 2019).

## 4 An evidence-led workflow

Figure 1 summarises the workflow. The diagram itself supports only the generation stage (Sections 4.3 and 4.4); the stages before it frame the problem, and the stages after it establish the cause with evidence.

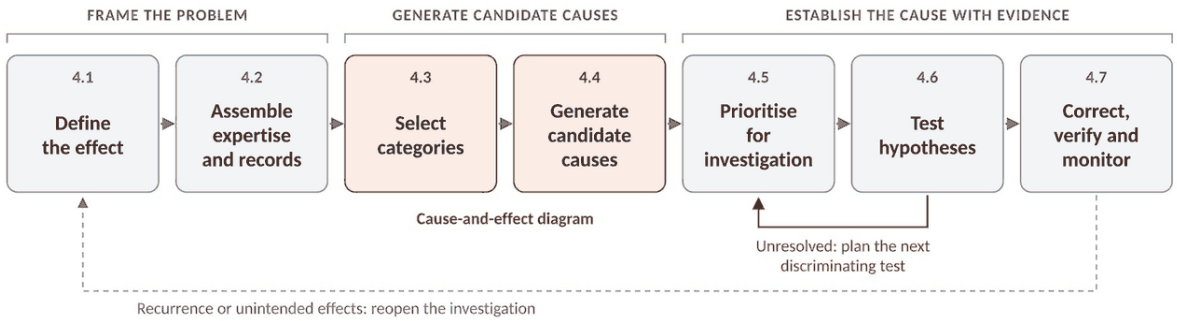



**Figure 1.** Evidence-led workflow for a cause-and-effect investigation. Unresolved hypotheses return to prioritisation; recurrence or unintended effects reopen the investigation.

### 4.1 Define the effect precisely

The effect should be written as an observable condition, not as an explanation. It should identify the affected item or process, the deviation from expected behaviour and the conditions under which the

deviation occurs. Where possible, it should also state the frequency, timing, operating mode, build or configuration, and the source of the evidence. For example, ‘controller failure’ is too broad, whereas ‘the prototype controller resets within 200 ms of enabling the motor from a 24 V supply in 7 of 10 trials’ is testable, provided that records support those details. Values that have not been measured should not be inserted.

The system boundary and the event of interest should also be defined. If two failure modes have different signatures or operating conditions, they should be analysed separately before the results are combined. A well-bounded effect reduces ambiguity and makes it possible to judge whether a proposed cause explains the specific observation.

### 4.2 Assemble relevant expertise and records

The team should include people with direct knowledge of design, implementation, integration, test, manufacturing and operation, as relevant. Available evidence should be reviewed before or during the session: requirements, configuration records, test logs, schematics, source revisions, change history, measurement data, environmental conditions and event timelines. Observed facts should be kept separate from interpretations. A facilitator should record dissenting explanations and unknowns rather than resolving them by seniority or majority vote.

### 4.3 Select categories that fit the system

A small number of useful top-level branches works best. For an embedded product, suitable starting categories might be power and signal integrity; hardware and components; firmware and configuration; interfaces and integration; requirements and design; manufacturing and assembly; test set-up and instrumentation; and operating environment. Categories can be adapted or merged as the team learns more. The traditional ‘6M’ categories (people, machine, method, material, measurement and environment) are one possible starting point, but they are neither a requirement nor a complete model of R&D.

### 4.4 Generate and structure candidate causes

For each branch, the team asks which condition could produce the effect under the observed circumstances. Entries should be phrased as specific, falsifiable statements. ‘Power problem’ is not actionable; ‘the 3.3 V rail falls below the microcontroller’s specified operating range during motor start-up’ is a testable hypothesis, provided that the device specification and a rail measurement are available. Sub-causes should be recorded where they clarify a proposed mechanism, and a cause may appear under more than one category when the relationship genuinely cuts across them.

At this stage the aim is a broad but relevant set of explanations. Branches should not be labelled ‘root causes’ before they have been investigated. Recording the source of each idea, such as a log entry, a measurement, a previous failure or an expert suggestion, allows the team to distinguish evidence from speculation later.

### 4.5 Prioritise for investigation, not by vote alone

Candidate causes should be selected for follow-up against explicit criteria: consistency with the event timeline, the supporting or contradictory evidence available, the ability to reproduce the failure, the potential severity, and the cost and safety of the test. Multi-voting can help a group to agree which hypotheses to examine first, but a vote is not evidence of causality. Hypotheses that are improbable but of high consequence should be retained when the risk warrants it.

### 4.6 Test hypotheses and record the result

For each prioritised cause, the team should specify the predicted observation, the measurement or analysis that could discriminate it from the alternatives, the test conditions and the decision rule. The least intrusive test that can resolve the question while preserving the relevant operating conditions is preferred. Depending on the hypothesis, suitable methods include oscilloscope measurements, trace or log analysis,

configuration comparison, code inspection, environmental monitoring, fault injection, controlled reproduction, statistical analysis and design of experiments. Tools should be selected according to the causal question; no single method suits every investigation.

Results should be recorded as supported, rejected, unresolved or not tested, with the evidence and configuration identified. A cause should not be rejected solely because one test failed to reproduce an intermittent event unless that test had adequate sensitivity and representative conditions. Where the evidence is inconclusive, the uncertainty should be retained and the next discriminating test planned.

### 4.7 Correct, verify and monitor

Corrective actions should be linked to the supported cause or causes, and an acceptance criterion should be defined before implementation where practicable. Verification should show both that the change was implemented correctly and that it addresses the failure under representative conditions. The team should also check for unintended effects and for similar failure mechanisms elsewhere in the design, code, process or test system. Recurrence and relevant performance measures should be monitored over a period and sample size appropriate to the risk and the event rate. The analysis should be closed only when the evidence supports the conclusion and the effectiveness of the action has been assessed.

## 5 Worked example: unexpected microcontroller halt

**Effect statement.** A prototype microcontroller-based unit stops performing its intended function during operation. This statement is deliberately incomplete: before testing, the team must establish whether the unit has halted, has reset, is blocked waiting for an event, is disconnected from its load or is merely failing to report its status. The distinction changes the candidate causes.

**Boundary and evidence.** The team identifies the exact board revision, microcontroller part and configuration, firmware build, power source, connected loads, test fixture and operating conditions. It captures timestamps and the available reset-cause, fault, watchdog and communication logs, and it instruments the relevant supply, reset, clock and interface signals without materially changing the behaviour of the system. Whether a device provides a particular reset-cause register or fault trace is specific to the microcontroller and must be checked in its datasheet and reference manual.

Figure 2 shows the resulting diagram once the effect has been refined to ‘the prototype loses its intended function when the motor is enabled’. Table 1 pairs each branch with the evidence or test that could discriminate between its hypotheses.

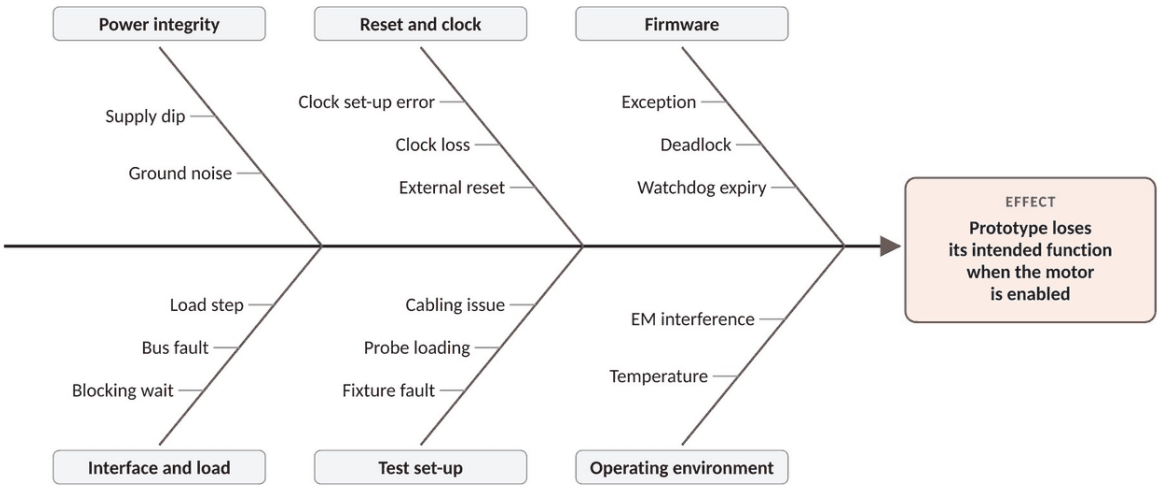



**Figure 2.** Cause-and-effect diagram for the worked example. Every entry is a candidate cause, not a finding.

**Table 1.** Example hypotheses and follow-up evidence. Entries are candidate explanations, not findings; the last row combines the test set-up and operating-environment branches of Figure 2.

| Branch | Candidate hypothesis | Discriminating evidence or test |
|---|---|---|
| Power and signal integrity | A supply transient or ground disturbance coincides with the event. | Capture the supply and reset signals at the microcontroller pins with adequate bandwidth and triggering; compare event and non-event trials. |
| Reset and clock | An external reset, a loss of clock or a clock configuration error causes execution to stop or restart. | Observe reset and clock behaviour; inspect boot and reset-cause records; compare the configuration with the device specification. |
| Firmware | A watchdog expiry, deadlock, exception or invalid state prevents expected progress. | Use trace, fault status, watchdog records, assertions or a targeted diagnostic build; test the predicted failure path. |
| Interface and load | A peripheral or connected load causes a blocking wait, a bus fault or an electrical disturbance. | Capture interface activity and status; isolate or substitute the load under controlled conditions while holding other variables constant. |
| Test set-up and environment | The fixture, probes, cabling, temperature or electromagnetic environment triggers or masks the behaviour. | Repeat with a documented alternative set-up or a controlled environmental condition; verify the instrumentation and configuration. |

### 5.1 Illustrative evidence

The team prioritised the supply-dip hypothesis on the power-integrity branch because the failure coincided with motor enable and the microcontroller supply voltage could be measured directly during the event.

In this hypothetical case, the shared supply enters current limiting at motor start, as indicated by current measurements and by its current-limit flag. At the microcontroller pins, a voltage dip precedes the restart, and the reset-cause register reports a brown-out. With the firmware, motor load and test sequence held constant, powering the microcontroller rail from a regulated bench supply eliminates both the dip and the restart; reconnecting the original supply reproduces both.

### 5.2 Root cause and corrective action

In this example, the shared supply has insufficient transient current capacity for the motor inrush. The rail sags below the microcontroller's brown-out threshold and triggers a reset. The corrective action is a dedicated microcontroller supply sized for the expected input transients. Verification confirms the rail margin and the absence of resets during repeated motor starts at the specified limits of input voltage, load and temperature.

## 6 Common pitfalls in using the method

- **Vague effect statements.** Broad wording produces broad branches that are difficult to test. The event should be bounded by observations and configuration.
- **Confusing ideas with evidence.** A plausible branch is still a hypothesis. A test and an evidence record should be attached before it is called causal.
- **Premature closure.** Selecting the most familiar explanation can exclude interacting or systemic factors. The candidate should be compared with the timeline and with alternative explanations.
- **Overloaded diagrams.** Excessive detail can obscure the investigation. The main diagram should remain readable, with evidence, test plans and detailed causal chains held in linked records.
- **Rigid categories.** Generic categories may omit requirements, integration, software configuration or effects of the measurement system. Branches should be tailored to the system.

- **Unstructured prioritisation.** Popularity or seniority should not substitute for evidence, risk and a discriminating test.
- **Unverified corrective action.** An implemented change is not proof of effectiveness. Acceptance criteria should be defined, the change verified under representative conditions and recurrence monitored.

## 7 Integration with other engineering methods

A fishbone diagram is one element of an investigation; it is neither a complete life cycle nor a substitute for engineering analysis. Event timelines establish sequence; logs and measurements characterise the event; fault trees provide a formal logical decomposition of a hazardous or complex event; failure mode and effects analysis examines prospective failure modes and their effects; statistical tools assess patterns and relationships; and controlled experiments discriminate between competing mechanisms (ASQ, no date b). NASA's guidance likewise treats the diagram as one possible logic-flow approach and describes further investigation to confirm or reject the prioritised explanations (NASA GSFC, 2019).

Traceability should be maintained between the problem statement, candidate causes, evidence, test results, conclusion, action, verification and closure decision. A diagram that is revised as the evidence changes is more useful than a visually polished diagram detached from the investigation record.

## 8 Conclusion

Cause-and-effect diagrams help R&D teams to organise candidate explanations and to make their reasoning visible. Their engineering value depends on what follows: a bounded problem, system-relevant categories, explicit hypotheses, discriminating tests, objective records and verified corrective action. The diagram can structure the search for causes, while measurements, analysis and controlled tests establish which explanations are supported. Used with this discipline, the method provides a practical entry point to a broader engineering investigation without overstating what the diagram alone can prove.

## References

American Society for Quality (no date a) *What is a fishbone diagram? Ishikawa cause & effect diagram*. Available at: https://asq.org/quality-resources/fishbone (Accessed: 20 November 2026).

American Society for Quality (no date b) *Cause analysis tools*. Available at: https://asq.org/quality-resources/root-cause-analysis/tools (Accessed: 20 November 2026).

Ishikawa, K. (1986) *Guide to quality control*. 2nd rev. edn. Tokyo: Asian Productivity Organization.

National Aeronautics and Space Administration (no date) 'SWE-204: process assessments', in *NASA software engineering handbook*, version D. Available at: https://swehb.nasa.gov/x/cgIfBg (Accessed: 20 November 2026).

National Aeronautics and Space Administration Goddard Space Flight Center (2019) *Guideline for forming and operating failure review boards and anomaly review boards*. GSFC-HDBK-8700. Greenbelt, MD: NASA Goddard Space Flight Center. Available at: https://standards.nasa.gov/node/2402 (Accessed: 20 November 2026).

### About the author

Henrik Andersen is an electronic engineer with thirty years of R&D, developing hardware and software for automotive, telecommunication, ATE, consumer and industrial products. He leads products from the first idea to start of production and works as an independent electronic engineering consultant under the name Silicon Meets Code.

© 2026 Henrik Andersen. Silicon Meets Code Technical Paper TP-01, November 2026. The examples in this paper are hypothetical and are provided for illustration only.