Using Cause-and-Effect Diagrams in Research and Development

Henrik Andersen

M.Sc. in Electrical Engineering

Abstract

Cause-and-effect diagrams, also called fishbone or Ishikawa diagrams, provide a structured way to generate and organize candidate explanations for an observed problem. Their value in research and development (R&D) lies in broadening a team's search space and making assumptions visible; the diagram itself does not establish causality or prove a root cause. This paper presents an evidence-led workflow for using the method in engineering investigations. It covers problem definition, category selection, cause generation, prioritization, 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 test setup. The method is most useful when paired with engineering records, measurements, controlled reproduction, and appropriate analytical tools. The paper is a practice-oriented synthesis and reports no new experimental results.

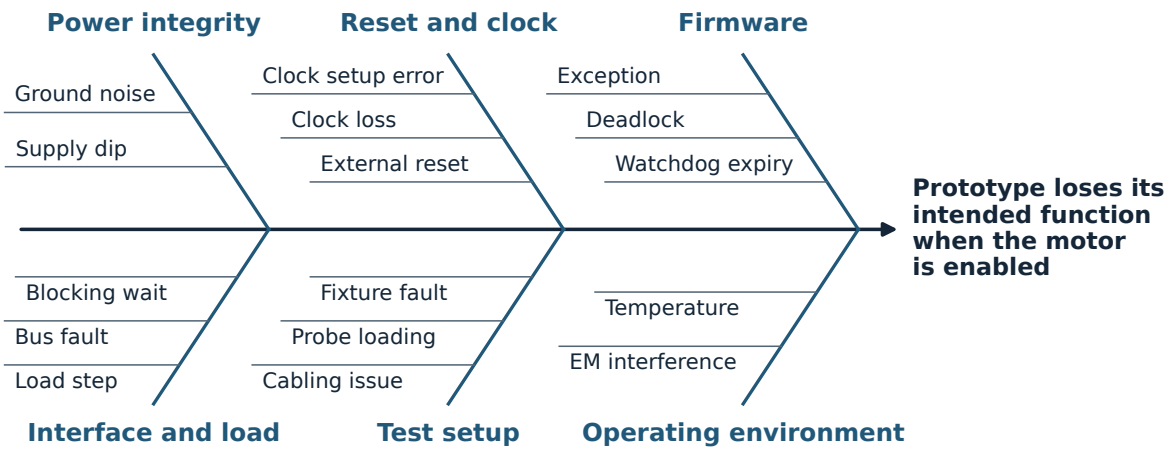


Figure 1. Candidate causes for the microcontroller example in Section 5.

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 organizing plausible causes before committing to a test or corrective action. The method is commonly called a fishbone diagram because the completed diagram resembles a skeleton, and an Ishikawa diagram after Kaoru Ishikawa (ASQ, n.d.-a; Ishikawa, 1986).

The diagram is a cause-generation and organization tool. It can support root-cause analysis, but it does not independently 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 among the hypotheses and a corrective action whose effectiveness is verified.

This paper sets out a practical workflow for applying the diagram to engineering problems. It 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 engineering root-cause analysis. It reports no new experiments, field data, or quantified comparison of investigation methods. Accordingly, statements about process benefits are framed as intended uses rather than measured performance improvements. The example is a procedural demonstration, not a case study.

The terms *candidate cause* and *hypothesis* are used for explanations entered in the diagram. *Contributing factor* denotes a condition that influences the event but may not be sufficient to cause it. *Root cause* is reserved for a cause supported by the investigation and linked to an effective action that prevents or materially reduces recurrence. In practice, an event can have multiple interacting causes; the analysis should not force a single-cause conclusion when the evidence does not support one (NASA, n.d.).

3. What the diagram does and does not establish

A cause-and-effect diagram starts with a defined effect, typically placed at the head of a right-facing spine. Major branches represent categories; smaller branches represent candidate causes and their sub-causes. The team uses the structure to ask what conditions could produce the observed effect, capture ideas in a visible form, and 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 being investigated (ASQ, n.d.-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. Nor does visual placement imply a proven causal chain. These limitations mean that prioritization must lead to evidence collection and testing (ASQ, n.d.-b). NASA's software-engineering guidance similarly separates the generation and prioritization of possible causes from further exploration or testing to identify the actual cause (NASA, n.d.).

4. An evidence-led workflow

4.1 Define the effect precisely

Write the effect as an observable condition, not an explanation. Include the affected item or process, the deviation from expected behaviour, and the conditions under which it occurs. Where possible, state frequency, timing, operating mode, build or configuration, and the evidence source. For example, "controller failure" is too broad; "prototype controller resets within 200 ms of enabling the motor under a 24 V supply, in 7 of 10 trials" is testable if those details are supported by records. Do not insert values that have not been measured.

Define the system boundary and the event of interest. If two failure modes have different signatures or operating conditions, analyze them separately before combining results. A well-bounded effect reduces ambiguity and makes it possible to determine whether a proposed cause explains the specific observation.

4.2 Assemble relevant expertise and records

Include people with direct knowledge of design, implementation, integration, test, manufacturing, and operation as relevant. Review available evidence before or during the session: requirements, configuration records, test logs, schematics, source revisions, change history, measurement data, environmental conditions, and event timelines. Keep observed facts 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

Use a small number of useful top-level branches. For an embedded product, starting categories might be power and signal integrity; hardware and components; firmware and configuration; interfaces and integration; requirements and design; manufacturing and assembly; test setup and instrumentation; and operating environment. Adapt or merge categories when the team learns more. The traditional "6M" categories (people, machine, method, material, measurement, and environment) are one possible starting point, not a requirement and not a complete R&D model.

4.4 Generate and structure candidate causes

For each branch, ask what condition could produce the effect under the observed circumstances. Phrase entries as specific, falsifiable statements. "Power problem" is not

actionable; “the 3.3 V rail falls below the MCU’s specified operating range during motor startup” is a testable hypothesis, provided the device specification and rail measurement are available. Record sub-causes where they clarify a proposed mechanism. A cause may appear under more than one category when the relationship is genuinely cross-cutting.

At this stage, optimize for a broad but relevant set of explanations. Do not label branches “root causes” before investigation. Record the source of each idea where useful, for example, a log entry, a measurement, a prior failure, or an expert suggestion, so that the team can distinguish evidence from speculation later.

4.5 Prioritize for investigation, not by vote alone

Choose candidate causes for follow-up using explicit criteria: consistency with the event timeline, available supporting or contradictory evidence, ability to reproduce the failure, potential severity, and the cost and safety of the test. Multi-voting can help a group identify which hypotheses members wish to examine first, but a vote is not evidence of causality. Preserve low-frequency but high-consequence hypotheses when the risk warrants it.

4.6 Test hypotheses and record the result

For each prioritized cause, specify the predicted observation, the measurement or analysis that could discriminate it from alternatives, the test conditions, and the decision rule. Use the least intrusive test that can resolve the question while preserving the relevant operating conditions. Depending on the hypothesis, suitable methods may include oscilloscope measurements, trace or log analysis, configuration comparison, code inspection, environmental monitoring, fault injection, controlled reproduction, statistical analysis, or design-of-experiments methods. Select tools based on the causal question; no single method is appropriate for every investigation.

Record results 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 the test had adequate sensitivity and representative conditions. Where evidence is inconclusive, retain uncertainty and plan the next discriminating test.

4.7 Correct, verify, and monitor

Link corrective actions to the supported cause or causes. Define an acceptance criterion before implementation where practical. Verify both that the change was implemented correctly and that it addresses the failure under representative conditions. Check for unintended effects and similar failure mechanisms elsewhere in the design, code, process, or test system. Monitor recurrence and relevant performance measures over a period and sample size appropriate to the risk and event rate. Close the analysis only when the evidence supports the conclusion and the action’s effectiveness has been assessed.

5. Worked example: unexpected microcontroller halt

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

Boundary and evidence. Identify the exact board revision, MCU part and configuration, firmware build, power source, connected loads, test fixture, and operating conditions. Capture timestamps and available reset-cause, fault, watchdog, and communication logs. Instrument the relevant supply, reset, clock, and interface signals without materially changing the system behaviour. Whether a device exposes a particular reset-cause register or fault trace is MCU-specific and must be checked in its datasheet and reference manual.

Branch	Candidate hypothesis	Discriminating evidence or test
Power and signal integrity	A supply transient or ground disturbance coincides with the event.	Capture supply and reset signals at the MCU pins with adequate bandwidth and triggering; compare event and non-event trials.
Reset and clock	An external reset, clock loss, or clock configuration error causes execution to stop or restart.	Observe reset and clock behaviour; inspect boot and reset-cause records; compare 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, bus fault, or electrical disturbance.	Capture interface activity and status; isolate or substitute the load under controlled conditions while preserving other variables.
Test setup and environment	The fixture, probe, cabling, temperature, or electromagnetic environment triggers or masks the behaviour.	Repeat with a documented alternate setup or controlled environmental condition; verify instrumentation and configuration.

Table 1. Example hypotheses and follow-up evidence. Entries are candidate explanations, not findings.

Illustrative evidence

The team prioritised the supply-dip hypothesis under Power integrity because the failure coincided with motor enable and the MCU supply voltage could be measured directly during the event.

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

Root cause and corrective action

In this example, the shared supply has insufficient transient current capacity for motor inrush. The rail sags below the MCU brownout threshold and triggers a reset. Add a dedicated MCU supply sized for the expected input transients. Verify rail margin and the absence of resets during repeated motor starts at the specified input-voltage, load, and temperature limits.

6. Common failure modes in using the method

- **Vague effect statements:** broad wording produces broad, difficult-to-test branches. Bound the event using observations and configuration.
- **Confusing ideas with evidence:** a plausible branch is still a hypothesis. Attach a test and evidence record before calling it causal.
- **Premature closure:** selecting the most familiar explanation can exclude interacting or systemic factors. Compare the candidate with the timeline and alternative explanations.
- **Overloaded diagrams:** excessive detail can obscure the investigation. Keep the main diagram readable and move evidence, test plans, and detailed causal chains into linked records.
- **Category rigidity:** generic categories may omit requirements, integration, software configuration, or measurement-system effects. Tailor branches to the system.
- **Unstructured prioritization:** popularity or seniority should not substitute for evidence, risk, and a discriminating test.
- **Unverified corrective action:** an implemented change is not proof of effectiveness. Define acceptance criteria, verify under representative conditions, and monitor recurrence.

7. Integration with other engineering methods

A fishbone diagram is one element of an investigation, not a complete lifecycle or a substitute for engineering analysis. Use event timelines to establish sequence; logs and measurements to characterize the event; fault trees when a formal logical decomposition of a hazardous or complex event is needed; failure mode and effects analysis to examine prospective failure modes and their effects; statistical tools to assess patterns or relationships; and controlled experiments when competing mechanisms need discrimination (ASQ, n.d.-b). NASA's guidance treats the diagram as one possible logic-flow approach and describes further investigation to confirm or reject prioritized explanations (NASA GSFC, 2019).

Maintain traceability between the problem statement, candidate causes, evidence, test result, conclusion, action, verification, and closure decision. A diagram that is revised as 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 organize candidate explanations and make their reasoning visible. Their engineering value depends on what follows: a bounded problem, system-relevant categories, explicit hypotheses, discriminating tests, objective records, and verification of corrective action. The diagram can structure the search for causes; measurements, analysis, and controlled tests establish which explanations are supported. Used with this discipline, it provides a practical entry point to a broader engineering investigation without overstating what the diagram alone can prove.

References

American Society for Quality (ASQ) (n.d.-a) 'What is a fishbone diagram? Ishikawa cause and effect diagram'. Available at: https://asq.org/quality-resources/fishbone (Accessed: 19 November 2026).

American Society for Quality (ASQ) (n.d.-b) 'Root cause analysis tools'. Available at: https://asq.org/quality-resources/root-cause-analysis/tools (Accessed: 19 November 2026).

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

National Aeronautics and Space Administration (NASA) (n.d.) 'SWE-204: Process assessments', *NASA Software Engineering Handbook, Version C*. Available at: https://swehb.nasa.gov/spaces/SWEHBVC/pages/50889445/SWE-204%2B-%2BProcess%2BAssessments (Accessed: 19 November 2026).

National Aeronautics and Space Administration Goddard Space Flight Center (NASA GSFC) (2019) *Guideline for forming and operating failure review boards and anomaly review boards*. GSFC-HDBK-8700, Change 0. Available at: https://standards.nasa.gov/node/2402 (Accessed: 19 November 2026).