# Errors, Defects, Failures, Anomalies, Root Causes, and Root Cause Analysis (RCA)

## 1. Error (*Mistake by a human*)

**# Source**: **Chatgpt (GPT-5-Standard) at [8/12/2025]**

* **Meaning:** A wrong action, decision, or misunderstanding made by a **developer, analyst, tester, or any team member** during the software process.
* **Cause:** Misunderstanding requirements, wrong logic, typo in code, etc.
* **Example:**

****

* + A programmer writes > instead of >= in the code.

## 2. Defect (*Bug in the product*)

**# Source**: **Chatgpt (GPT-5-Standard) at [8/12/2025]**

* **Meaning:** An **imperfection in the software code or documentation** caused by an error.
* **When found:** During development or testing — before the product goes live.
* **Example:**

****

* + Because of the wrong > operator, the “discount eligibility” function skips valid customers.

## 3. Failure (*Wrong behavior when running*)

**# Source**: **Chatgpt (GPT-5-Standard) at [8/12/2025]**

* **Meaning:** When the **software does not perform as expected during execution** — often because a defect was triggered.
* **When found:** During execution in testing or in production.
* **Example:**
  + A customer with exactly 100 points should get a discount, but the system denies it when used.

**is all bugs make Failure**

No — **not all bugs (defects) cause a failure**.

**Why?**

A **failure** only happens if:

1. The defective part of the code **is executed**, and
2. The conditions exist for the bug to be triggered.

If the bug stays hidden (code path never used, feature not activated, data not triggering it), it won’t cause an immediate failure — but it’s still a risk.

**Examples**

* **Bug without failure:**
  + There’s a typo in a function that’s never called in the current version — no failure occurs *yet*.
* **Bug that causes failure:**
  + Login function bug that’s executed every time a user logs in → immediate visible failure.

💡 **Key idea:**

* **All failures are caused by bugs**,
* but **not all bugs cause failures right away** — some remain dormant until certain conditions bring them to life.

## 4. Anomalies

**# Source**: **Chatgpt (GPT-5-Standard) at [9/8/2025]**

**Definition (in testing):**  
An anomaly is **any event, behavior, or output that is different from what you expected** when you run a test.

It’s a **general term** — you don’t know yet if it’s a bug, a mistake in the test, or just unclear requirements.

**Examples of anomalies:**

* The software **crashes** when you click “Save”.
* A report shows the wrong **date format** compared to the specification.
* A screen **loads slower** than the performance requirement allows.
* A test case itself is **wrong** and gives a false alarm.
* Documentation says one thing, but the system does another (inconsistency).

**What happens next:**

1. You log the anomaly (record it).
2. You analyze it — is it:
   * a **defect** in the code?
   * a **problem** in the test itself?
   * a **gap** or **error** in the requirement?
3. Then you classify and report it accordingly.

💡 **In short:**

In testing, “anomaly” = anything unexpected during testing that **might** indicate a problem — a broad term used **before** confirming it’s really a defect.

## 5. Root Cause (*The underlying reason*)

**# Source**: **Chatgpt (GPT-5-Standard) at [8/12/2025]**

* **Meaning:** The **original source** of the defect — what actually caused the error in the first place.
* **Purpose:** Found through **Root Cause Analysis (RCA)** to prevent it from happening again.
* **Example:**
  + Root cause: The requirement document said “over 100 points” instead of “100 or more points,” and the developer coded it exactly as written.

## 6. Root Cause Analysis (RCA)

**# Source**: **Chatgpt (GPT-5-Standard) at [9/3/2025]**

**🔹 Root Cause Analysis (RCA)**

Root Cause Analysis is a **problem-solving method** used to identify the **fundamental reason** (the “root cause”) why a problem, defect, or failure occurred — instead of just fixing the **symptom**.

The idea is:

*If you remove the root cause, the problem won’t happen again (or will happen less often).*

**🔹 Key Points**

* **Symptom** → What you see (error message, system crash, failed test).
* **Immediate cause** → What directly caused the symptom (e.g., null pointer).
* **Root cause** → The deeper underlying reason (e.g., missing input validation in requirements).

**🔹 Common RCA Techniques**

1. **5 Whys**
   * Ask “why?” repeatedly until you reach the real root cause.
   * Example:
     + Problem: Website crashed.
     + Why? → Server overloaded.
     + Why? → Too many database queries.
     + Why? → Queries not optimized.
     + Why? → No performance testing done.
     + Root cause → Performance testing missing in process.
2. **Fishbone Diagram (Ishikawa)**
   * Categorizes causes under **People, Process, Tools, Environment, Materials, Management**.
3. **Fault Tree Analysis**
   * Start from the failure and break down into logical possible causes.

**🔹 Example in Software Testing**

**Failure (Symptom):** Mobile app crashes when user uploads a large image.

* Immediate cause: App runs out of memory.
* Root cause:
  + No image size validation in requirements.
  + Developers didn’t add compression logic.
  + Test cases didn’t cover large file uploads.

👉 Fixing the **root cause** (adding requirements for image limits + test cases + compression) prevents future failures.

✅ In short:  
**RCA = Finding the *why behind the why*, so you fix the real problem, not just the surface error.**

## Relationship Flow

**# Source**: **Chatgpt (GPT-5-Standard) at [9/3/2025]**

1. **Error** → made by a human.
2. **Defect** → appears in the software artifact/code.
3. **Failure** → happens when the defect is triggered during execution.
4. **Root Cause** → explains *why* the error happened.
5. **Root Cause Analysis (RCA)** - process used to identify and eliminate the root cause.
6. **Anomaly** → any unexpected behavior, output, or event noticed during testing

## 💡 Quick analogy:

**# Source**: **Chatgpt (GPT-5-Standard) at [9/3/2025]**

* **Error:** Chef misreads the recipe.
* **Defect:** Wrong amount of sugar in the cake batter.
* **Failure:** Cake tastes bad when eaten.
* **Root Cause:** Recipe text was unclear or misprinted.
* **Root Cause Analysis (RCA)** → Investigating why the recipe was unclear and fixing it (e.g., rewriting the recipe clearly).
* **Anomaly** → While baking, someone notices the cake batter **looks different than usual** (unexpected observation — may or may not be a real problem).