

# PBRCIM – Pattern-Based Root Cause Isolation Model

*Codex Version: V2 – Broad-Spectrum Engineering Edition*

**Formulated & Engineered by CursorLord (Usenobong Jonah)**

---

## **What is PBRCIM?**

PBRCIM is a high-clarity diagnostic model for isolating root causes of system failures in digital environments.

It identifies silent bugs and structural anomalies by analyzing mismatches between intended logic patterns and real-world outcomes even when no errors are thrown.

It operates beyond logs, tests, or breakpoints tracing **behavior fractures** and logic derailments in a structured way.

---

## **Origin & Model Purpose**

Conceived during a national infrastructure project in 2023, PBRCIM was built to resolve deployment irregularities in high-stakes CI/CD pipelines.

It now functions as an elite fault-analysis model for any software layer - frontend, backend, distributed systems, or compute pipelines.

PBRCIM is **not tied to any language, stack, or platform.**

It is a *thinking tool*, a pattern lens, and a debugging methodology.

---

## First Use Case

The model was first deployed to resolve a failed deployment sequence in a CI/CD pipeline.

Logs showed success, but deploys silently skipped.

PBRCIM revealed the logic fracture: deploys were conditional on Git tags. The activation pattern was misaligned. Correcting this restored system behavior.

---

## How PBRCIM Works

PBRCIM operates as a stepwise mental framework:

1. **Observe** the failure (e.g., job executes but output is invalid).
  2. **Define** the intended logic or behavior flow.
  3. **Trace** each output or system checkpoint.
  4. **Compare** actual vs expected behavior pattern.
  5. **Isolate** the step where divergence or mismatch occurs.
  6. **Diagnose** what fractured - syntax, data shape, timing, state, or dependency.
  7. **Patch** the misalignment and retest the system's behavior.
- 

## Other Use Cases

### CI/CD Deployment Skipped

- **Expected:** Commit → Build → Test → Deploy
- **Observed:** Deploy skipped despite green tests
- **PBRCIM Insight:** Hidden logic condition – deploys only triggered from tagged branches. Pattern updated.

## API Chain - Microservice Timeout

- **Expected:** Auth → Fetch → Process → Respond
- **Observed:** Intermittent 504 timeouts
- **PBRCIM Insight:** Retry setting mismatch. Gateway retried twice, but service required three to stabilize. Fixed in config.



## Database Job – Missing Data After Migration

- **Expected:** Backup → Migrate → Verify → Delta Import
- **Observed:** Delta records missing
- **PBRCIM Insight:** Schema mismatch – field `created_at` expected; source had `inserted_at`. Corrected.



## Frontend UI – Modal Button Inactive

- **Expected:** User click → JS handler fires → Modal animates
- **Observed:** Visually okay, but no response
- **PBRCIM Insight:** Selector mismatch: handler used `.btn-start`, DOM had `.start-btn`. Alignment fixed.

---

## Philosophy of Use

PBRCIM doesn't wait for crash logs. It hunts the **intent mismatch** - the place where the system's inner design fell out of sync with its output.

It is not a library or dependency; it is a *mental model* and diagnostic discipline.

---

