Beyond Monte Carlo: The Risk Chain Your Model Cannot See

PROBABILISTIC CHAIN ANALYSIS

The Case Study

What it is. Why it matters. How it works.

Explained so simply your intern could present it tomorrow.

Nikhil Dhand, PMP

Creator of Probabilistic Chain Analysis™ and RiskPulseV12

Senior Project Controls Consultant | Quantitative Risk Specialist

ORCID: 0009-0000-6526-8399 | Zenodo: 10.5281/zenodo.19436269

Contact: nikhildhand.research@gmail.com | LinkedIn: https://www.linkedin.com/in/nikhil-dhand/

Read this first

This case study will take you about 12 minutes. When you finish, you will understand something that most risk professionals with 20 years of experience have felt but never been able to name.

You will understand why risk registers lie to you. Not because anyone is dishonest. Because the tool itself cannot see the thing that actually goes wrong on complex projects.

> **The thing that goes wrong is never a single risk. It is the chain between risks.**

This case study explains Probabilistic Chain Analysis — PCA — the methodology I built to make that chain visible. No equations. No jargon you need to Google. Just the logic, the diagrams, and the reason it matters.

Let’s go.

The problem

You are managing a £2 billion infrastructure programme. You have a risk register with 200 risks. Each risk has a probability, an impact, and an owner. You run a Monte Carlo simulation. It gives you a P80 cost estimate of £2.3 billion. You present it to the board. The board approves the funding.

Two years later, the programme is at £2.7 billion and nine months late.

What happened?

Nothing on your register was wrong. Every risk was real. Every probability was reasonable. Every mitigation had an owner.

The problem is what your register could not show you:

```
YOUR RISK REGISTER:

  Risk A              (Testing delay)        —  P = 20%

  Risk B              (Documentation backlog) —  P = 30%

  Risk C              (Safety evidence gap)   —  P = 15%

  Risk D              (Opening delay)        —  P = 25%

  Risk E              (Prolongation cost)     —  P = 10%

  Five separate items. Five separate probabilities.
  No connections between them.
  The register sees five dominoes lying on a table.
```

Now here is what was actually happening on your programme:

```
WHAT WAS ACTUALLY HAPPENING:

  Testing delayed
       │
       ▼  (because testing was late, docs piled up)
  Documentation backlog formed
       │
       ▼  (because docs were late, safety evidence was incomplete)
  Safety evidence gap opened
       │
       ▼  (because safety evidence was missing, opening slipped)
  Opening delayed
       │
       ▼  (because opening slipped, costs kept running)
  Prolongation costs hit £400M

  The dominoes were not lying on a table.
  They were standing in a line.
  And the first one had already fallen.
```

Diagram 1: What the register sees vs what actually happens

> **The register counted the dominoes. Nobody noticed they were standing in a line.**

What the industry does now

The standard tool is called Monte Carlo simulation. Here is how it works, in the simplest terms possible.

Imagine you have five risks. Each one has a range of possible costs — best case, most likely, worst case. The Monte Carlo rolls dice on all five risks, thousands of times, and adds up the results each time. After 10,000 rolls, it draws a graph. The graph tells you: there is a 50% chance your project costs less than X, an 80% chance it costs less than Y.

That is useful. It gives you a range instead of a single number.

But here is the thing it does not do:

```
WHAT MONTE CARLO DOES:

  Roll dice on Risk A.  (independent)

  Roll dice on Risk B.  (independent)

  Roll dice on Risk C.  (independent Maybe Co-related consider A its Twin)

  Roll dice on Risk D.  (independent)

  Roll dice on Risk E.  (independent | Maybe Co-related consider B its Twin)

  Add them all up.

  Repeat 10,000 times.

  Draw a graph.

  Think of it like asking five strangers to each guess
  a number between 1 and 100. You add up their guesses.

  But the strangers don’t know each other (or maybe they are Twin – co-joint movement).
  Their guesses are unrelated.
```

Diagram 2: How Monte Carlo simulation works

“But Nikhil, Monte Carlo can add correlations.”

Yes. You can add a correlation matrix. This is a table that says: Risk A and Risk B are 60% correlated, meaning they tend to move together. When A is bad, B tends to be bad too.

That sounds helpful. But there are three problems with it.

The three gaps in the current approach

Gap 1: Correlation has no direction

```
CORRELATION SAYS:

  Risk A  <──── 60% correlated ────>  Risk B

  They move together. But which one caused the other?

  Did A cause B? Did B cause A? Did something else cause both?

  Correlation does not tell you.

  It is like saying “umbrellas and rain are correlated.”

  True. But buying an umbrella does not cause rain.

  If you want to stop getting wet, you need to know
  which one comes first.
```

Diagram 3: Correlation has no direction

This matters because if you want to intervene — if you want to break the chain — you need to know where the chain starts. Correlation cannot tell you. It just says “these two things tend to happen together.”

Gap 2: The model does not update when risks fire

```
WHAT HAPPENS IN REAL LIFE:

  January:   Risk A fires. Testing is delayed.
  February:  You update the register. Risk A = happened.

  But what about Risk B?

  In your Monte Carlo model, Risk B is still sitting at 30%.
  It has no idea that Risk A just fired.
  It does not know that the testing delay has made
  a documentation backlog much more likely.

  You know this. Your gut knows this.
  Your model does not.

  You will find out at the next quarterly risk review.
  Three months from now.
  By then, B has already fired too.
```

Diagram 4: The model does not learn

> **Your gut updates in real time. Your model updates quarterly. That gap is where projects go wrong.**

Gap 3: The model tells you the number, not the path

```
WHAT MONTE CARLO TELLS THE BOARD:

  “Your P80 is £2.3 billion.”

  THE BOARD ASKS:

  “What do we do about it?”

  MONTE CARLO SAYS:

  “...”

  It cannot answer. It knows the total.
```

```
It does not know which risks caused which.
It cannot point to a specific node and say:
“Break it here, and the P80 drops to £1.9 billion.”
```

Diagram 5: The model gives you a number, not a plan

This is where experienced PMs get frustrated. They sit in review meetings, they look at the P80 number, they know something is wrong, and they cannot prove it. The dashboard is green. Their gut says red. And there is no tool that connects the two.

Enter PCA

Probabilistic Chain Analysis is the methodology I built to close all three gaps. Here is what it does, step by step.

Step 1: Turn risks into a chain

Instead of listing risks as independent items, PCA connects them. Every risk becomes a node. Every node has edges connecting it to the nodes it can trigger.

The question changes:

```
OLD QUESTION:         “What is the probability of Risk B?”
                       Answer: 30%.

PCA QUESTION:          “If Risk A fires, what happens
                        to the probability of Risk B?”
                       Answer: it jumps from 30% to 65%.
```

Diagram 6: The question that changes everything

That single shift — from “how likely” to “what does it trigger” — is the entire foundation of PCA.

Step 2: Map the full chain

Now do this for every risk on your programme. Connect them. You get a chain:

```
THE PCA CHAIN:

  [Testing delay]       P = 20%
       |
       ▼  A fires. B’s probability changes.
  [Docs backlog]        P was 30%  ➝  now 65%
       |
       ▼  B fires. C’s probability changes.
  [Safety evidence gap] P was 15%  ➝  now 70%
       |
       ▼  C fires. D’s probability changes.
  [Opening delay]       P was 25%  ➝  now 85%
       |
       ▼  D fires. E’s probability changes.
  [Prolongation cost]   P was 10%  ➝  now 90%

  The register said you had five 10-30% risks.
  PCA says you have one chain that compounds to near-certainty.
```

Diagram 7: The chain nobody was modelling

> **Five small risks on a register. One loaded mechanism in reality.**

Step 3: Simulate it

PCA does not just draw the chain. It runs it. The simulation engine — RiskPulseV12 — uses Bayesian network logic with conditional probability tables at each node. It runs thousands of simulations, just like Monte Carlo. But in each simulation, the risks are not independent. They fire in sequence. When A fires, B’s probability changes. When B fires, C’s probability changes. The output is a P50, P80, P90 distribution that reflects the chain, not just the sum.

```
MONTE CARLO OUTPUT:

  P50 = £2.1B    P80 = £2.3B    P90 = £2.5B
  (based on independent risks, maybe some correlation)


PCA OUTPUT:

  P50 = £2.2B    P80 = £2.6B    P90 = £2.9B
```

```
(based on chain propagation with conditional probabilities)

AND:
“The highest-leverage intervention point is Node B.
If you prevent the documentation backlog from forming,
the P80 drops from £2.6B to £2.0B.”
```

Diagram 8: PCA gives you a number AND a plan

Step 4: Update in real time

This is the part that makes PCA live rather than static. When Risk A fires on your actual programme — the testing delay happens — you tell the model. And the model does not just tick a box. It recalculates every downstream probability immediately.

```
BEFORE RISK A FIRES:

  A = 20%  ➝  B = 30%  ➝  C = 15%  ➝  D = 25%


RISK A FIRES. PCA UPDATES INSTANTLY:

  A = HAPPENED  ➝  B = 65%  ➝  C = 42%  ➝  D = 55%

  No waiting for the quarterly review.
  No hoping someone remembers to update the spreadsheet.
  The chain recalibrates itself.
```

Diagram 9: The model that learns

> **Your Monte Carlo is a photograph. PCA is a live feed.**

The three pillars of PCA

Everything above is built on three structural components. Here they are, as simply as I can put them.

```
┌───────────────────┐  ┌───────────────────┐  ┌───────────────────┐
│  PILLAR 1          │  │  PILLAR 2          │  │  PILLAR 3          │
│                    │  │                    │  │                    │
│  Chain Mapping     │  │   Simulation       │  │   Live Learning    │
│                    │  │                    │  │                    │
│  Draw the chain.   │  │  Run the chain.    │  │  Update the chain. │
│  Which risk        │  │  Simulate 10,000   │  │  When a risk fires │
│  triggers which?   │  │  scenarios with    │  │   in real life, the│
│  At what           │  │  conditional       │  │   model recalculates│
│  probability?      │  │  probabilities.    │  │   everything       │
│                    │  │                    │  │  downstream.       │
└─────────┬─────────┘  └─────────┬─────────┘  └─────────┬─────────┘
          │                      │                      │
          └──────────────────────┼──────────────────────┘
                                 │
                                 ▼
                   Programme outcome distribution
                   — before it happens.
```

Diagram 10: The three-pillar architecture of PCA

The mitigation problem

There is one more thing PCA does that the standard approach cannot. It tests your mitigations.

On every programme, you write down mitigations: "We will hire more testers." "We will start documentation early." The model assumes these mitigations work. Funding is sized on the basis that they will succeed.

But what if they do not? What is the probability that a mitigation fails?

Standard models do not ask this question. PCA does.

```
STANDARD APPROACH:

  Risk: testing delay       P = 40%
  Mitigation: hire testers  ➝  Risk reduced. Done.
  (No probability assigned to mitigation failing.)


PCA APPROACH:

  Risk: testing delay       P = 40%
       │
       ▼
  Mitigation: hire testers
       │                         │
       ▼                         ▼
  Works (P = 70%)          Fails (P = 30%)
  Risk contained.          Chain resumes.
                           Downstream risks fire.

  Mitigation is not a magic wand.
  It is a node with its own probability of failure.
```

Diagram 11: PCA treats mitigation as testable, not assumed

> **Every other model assumes your mitigations work. PCA asks: what if they don't?**

Where to break the chain

This is the part PMs care about most. Not the theory. The action.

PCA does not just tell you the chain exists. It tells you which node to break.

```
THE FULL CHAIN:

  A (20%) ➝ B (65%) ➝ C (70%) ➝ D (85%) ➝ E (90%)

  Current P80 cost: £2.6 billion


OPTION 1: Intervene at Node A (prevent testing delay)

  Difficulty: Very hard. Testing complexity is structural.
```

```
  P80 if successful: £1.8B
  But probability of preventing A: only 40%.


OPTION 2: Intervene at Node B (prevent docs backlog)

  Difficulty: Moderate. Start documentation parallel to testing.
  P80 if successful: £2.0B
  Probability of preventing B: 75%.

  BEST OPTION. Highest leverage. Most achievable.


OPTION 3: Intervene at Node D (prevent opening delay)

  Too late. By the time D fires, C has already cascaded.
  You are fighting the chain from the wrong end.
```

Diagram 12: PCA tells you where to intervene — and where it is too late

This is the conversation PCA makes possible. Instead of "our P80 is £2.6 billion, what do we do?" the conversation becomes "our P80 is £2.6 billion because the testing-to-documentation chain is active, and the highest-leverage intervention is at Node B, which brings the P80 down to £2.0 billion if we start documentation in parallel now."

That is a sentence a board can act on. A P80 number is not.

Why Level 4 is not enough

The industry uses maturity models to measure how good an organisation is at risk management. The CIPFA model goes from Level 1 (Risk Naive) to Level 5 (Risk Enabled). Most well-run programmes aim for Level 4: Risk Managed.

But Level 4 measures process, not chain visibility. You can have impeccable process — every risk logged, every owner assigned, every review completed on time — and still not see the chain.

PCA extends the model:

```
Level 1 – Risk Naive       (we don't think about risk)
Level 2 – Risk Aware        (we know risks exist)
Level 3 – Risk Defined      (we have a process)
```

```
Level 4 — Risk Managed      (the process works well)
Level 5 — Risk Enabled      (risk drives decisions)

Level 6 — Chain Visible     ← PCA

  The organisation can model conditional propagation
  between risk nodes, simulate how chains compound,
  and intervene at the right node before the chain
  fires downstream.

  Most programmes reach Level 4.
  The hardest programmes need Level 6.
  Level 6 did not exist before PCA.
```

Diagram 13: The PCA maturity extension

The full comparison

Here is everything, side by side.

```
                    STANDARD MC              PCA
                    ───────────────────     ───────────────────

Risks:              Independent items        Connected chain
Question:           How likely?              What does it trigger?
Correlation:        Static matrix            Conditional propagation
Direction:          None                     A causes B causes C
Updates:            Quarterly                Real-time
Mitigation:         Assumed to work          Tested as a node
Output:             A number (P80)           A number + a node to break
Maturity:           Level 4                  Level 6
Board answer:       "It might cost £2.3B"   "Break Node B. P80 drops £600M."
```

Diagram 14: Monte Carlo vs PCA — the full comparison

Closing

A colleague read my original PCA paper and told me she could not understand it. She is a smart, experienced PM. If she could not get through it, that was my failure, not hers.

So I wrote this.

PCA™ is not complicated. The maths behind it — Bayesian networks, Iman-Conover copulas, conditional probability tables, Metropolis-Hastings MCMC — that is complicated. But what PCA does is simple.

It looks at your risk register and asks one question:

> **If this risk fires, what happens to the next one?**

That question — asked at every node, updated in real time, simulated thousands of times — produces something the industry has never had before: a model that shows you not just how bad it could get, but exactly how it gets there, and exactly where to intervene.

The chain is always forming. On every programme, right now, risks are triggering risks that are triggering risks. The register cannot see it. The Monte Carlo cannot model it. But experienced PMs can feel it. They just never had a tool that could say it out loud.

Now they do.

Nikhil Dhand, PMP

Creator of Probabilistic Chain Analysis and RiskPulseV12
Senior Project Controls Consultant | Quantitative Risk Specialist
Independent Researcher in Quantitative Risk Management
itsnik6850@gmail.com

Original Paper: Dhand, N. (2026). Probabilistic Chain Analysis (PCA): A Framework for Running Projects in a Probabilistic World. Independent Technical Whitepaper.