

PAYMENTS LEADERSHIP REPORT

The Unknown Risk Report

What Payment Teams Are Not Validating

PruTAN | 2026

AUDIENCE	Payments Leadership
FOCUS	Hidden operational exposure and risk discovery
PLATFORM	Powered by PruTAN — Agentic Payment Assurance Platform

The risk that causes the incident is rarely the risk on the register.
THE PREMISE OF THIS REPORT

Executive Summary

Every payment organisation can describe what it has tested. Very few can describe what it has never asked.

Payment assurance is usually discussed in the language of coverage — how much of the estate is tested, how much is automated, how many defects were found, whether certification passed. These are measures of work that has been done. They say nothing about work that was never scoped.

This report is concerned with a third category: payment behaviour that exists in production, has never been validated anywhere, and appears on no backlog — because no one has yet identified it as a question.

ZONE 1 — VALIDATED Behaviour that has been tested and evidenced.	ZONE 2 — KNOWN GAPS Behaviour the organisation knows it has not covered.	ZONE 3 — UNKNOWN RISK Behaviour no one has thought to ask about.
--	--	--

Zone 1 is reported. Zone 2 is managed. Zone 3 is the zone that produces incidents — and it is the only zone that no dashboard currently shows.

Unknown risk is not the product of negligence. It is a structural consequence of how payment estates are built, changed and assured: distributed ownership, unavailable dependencies, happy-path bias, certification scope that stops at the scheme boundary, and a rate of change that outpaces the rate of validation.

This report sets out where unknown risk accumulates in a payment estate, why conventional metrics conceal it, and how PruTAN turns discovery into a repeatable engineering discipline rather than an outcome of production failure.

The practical question for payments leadership is not “are we testing enough?” It is “what do we not know about our own payment behaviour — and how would we find out?”

1. The Validation Boundary

Every payment estate has a boundary. Inside it, behaviour has been exercised, observed and evidenced. Outside it, behaviour exists and executes in production — but has never been proven anywhere.

The boundary is rarely drawn deliberately. It is the accumulated residue of what was convenient to test, what an environment allowed, what a certification demanded and what a deadline permitted.

What defines the boundary in theory	What defines it in practice
Payment risk	Environment availability

What defines the boundary in theory	What defines it in practice
Business criticality	Dependency cooperation
Regulatory obligation	Historical test inventory
Customer impact	Release timelines

When the boundary is set by practicality rather than risk, the resulting exposure is invisible by construction. Nothing in the reporting chain is wrong — the reporting simply describes the inside of the boundary.

2. Why Unknown Risk Is Structural

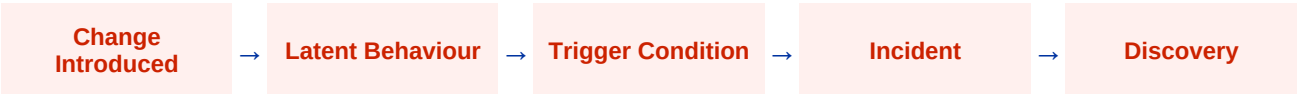
Unknown risk persists in well-run organisations with capable engineers and mature QA functions. It survives because six structural forces work against discovery.

Structural force	Effect on validation
Distributed ownership	No single team can see the end-to-end payment journey, so no single team can validate it
Dependency unavailability	External hosts and third parties cannot be driven into failure states on demand
Happy-path bias	Scenarios are written from the requirement, and requirements describe success
Certification scope	Scheme certification proves conformance, not operational behaviour under real conditions
Change velocity	Configuration, rules and routing change faster than scenarios are re-derived
Environment fidelity	Lower environments cannot reproduce production state, volume or timing

None of these are failures of diligence. They are properties of the architecture. Which means unknown risk cannot be solved by testing harder — only by changing how questions are generated and how conditions are reproduced.

3. The Anatomy of a Payment Incident

Payment incidents rarely begin at the moment they are detected. The characteristic pattern is a latency between introduction and discovery — often weeks or months.



The behaviour that caused the incident was usually present, deterministic and reproducible for the entire period. It was not detected because no scenario ever placed the system in the condition that exposed it.

Most payment incidents are not new failures. They are old behaviours meeting a new condition for the first time.

This reframes the objective. The goal of assurance is not to react faster to incidents. It is to reach the trigger condition in a controlled environment before production does.

4. The Eight Blind Spots

Across payment estates, unknown risk concentrates in eight recurring areas. Each is characterised by the same signature: the behaviour is essential, the failure is financial, and the validation is partial or absent.

Blind spot	What is rarely validated	Typical consequence
Failure paths	Timeouts, malformed responses, partial approvals, service unavailability	Incorrect customer outcome; inconsistent state
Reversal & refund integrity	Reversal after timeout, refund after partial reversal, cancellation races	Financial leakage; disputes
Duplicate & idempotency	Retry storms, duplicate submission, replayed messages	Double posting; double debit
Transaction state	State agreement across gateway, switch, ledger and core banking	Orphaned and mismatched transactions
Dependency behaviour	Fraud, tokenisation, 3DS and processor behaviour under degradation	Cascading failure; unsafe fallback
Reconciliation & settlement	Cut-off boundaries, late presentments, partial files, exception handling	Settlement breaks; manual remediation
Configuration drift	BIN ranges, limits, fraud rules, routing tables, MCC handling	Silent behaviour change without code change
Degraded-mode behaviour	Correctness under load, latency and partial connectivity	Integrity loss precisely when volume is highest

The remaining sections examine the five that most consistently escape validation — and what it takes to close each one.

5. Blind Spot: The Failure Path

A payment requirement describes an approval. The scenarios derived from it describe an approval. The regression suite that follows describes an approval. Failure behaviour is defined by exception — and exceptions are documented only after they occur.

What is validated	What actually executes in production
Approved	Approved
Declined — code 05	Declined across the full response-code space
	Issuer timeout, then retry, then late response
	Partial approval on a debit or prepaid card
	Malformed or truncated response
	Connection reset mid-authorisation
	Approval received after the customer has abandoned

The asymmetry is the point. One happy path is validated repeatedly; a dozen failure paths are validated once, or never. Yet it is the failure paths that determine whether money moves correctly when the ecosystem does not behave.

6. Blind Spot: Transaction State

A payment is not a request and a response. It is a state that must remain consistent across several systems that each hold their own opinion of it.



Conventional testing validates each system against its own expectation. Almost nothing validates that all of them agree after an interrupted, retried or reversed transaction.

A transaction that is “approved” in the switch, “failed” at the gateway and “posted” in the ledger has passed every individual test and broken the payment.

State divergence is the most expensive category of unknown risk, because it is discovered downstream — in reconciliation, in dispute volumes, or in a customer's statement.

7. Blind Spot: Dependency Behaviour

Payment platforms are increasingly assemblies of third-party services: fraud scoring, tokenisation, 3DS, orchestration, processors and payment rails. Each is validated in its normal condition, because that is the only condition a shared environment will produce.

Dependency condition	How often it is validated	How often it occurs in production
Normal response	Continuously	Almost always
Slow response	Rarely	Daily, under load
Timeout	Occasionally, if at all	Predictably, during incidents
Malformed response	Almost never	After every upstream change
Service unavailable	Almost never	During every dependency outage
Behaviour change without notice	Never	At the vendor's release cadence

The exposure is not the dependency itself. It is the payment platform's untested reaction to the dependency — the fallback, the retry, the timeout budget and the state left behind.

8. Blind Spot: Configuration Drift

A significant proportion of payment behaviour is not defined in code. It is defined in configuration: BIN ranges, velocity limits, fraud rules, routing tables, MCC handling, currency and country parameters, terminal profiles and scheme parameters.

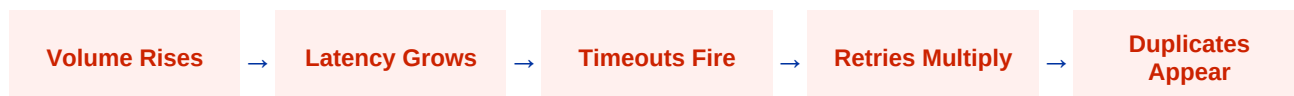
Configuration changes routinely bypass the assurance pipeline entirely.

Code change	Configuration change
Enters version control	Often changed directly in an operational console
Triggers a build and test cycle	Triggers no validation at all
Is reviewed and approved	Is approved operationally, not technically
Has a rollback path	Rollback is manual and frequently undocumented
Is visible in change records	Is visible only to the team that made it

The result is a class of behaviour change that is invisible to engineering, unvalidated by QA, and fully live in production.

9. Blind Spot: Behaviour Under Pressure

Performance testing answers a throughput question: how many transactions per second can the platform sustain? It does not answer the reliability question: does payment behaviour remain correct while it does so?



Every step in that chain is a correctness question disguised as a performance metric. A platform can meet its TPS target and still double-post, mis-sequence reversals or leave transactions in an indeterminate state — and a load report will show none of it.

Throughput proves the system survived the load. It does not prove the payments did.

10. Blind Spot: Reconciliation and Settlement

Authorisation receives the majority of assurance attention because it is synchronous and visible. Clearing, settlement and reconciliation receive far less — despite carrying the financial truth of the estate.

- Transactions that straddle the cut-off boundary
- Late presentments and delayed clearing files
- Partial, duplicated or out-of-sequence files
- Reversals presented after settlement has occurred
- Currency, fee and interchange edge cases
- Exception handling and manual adjustment paths

These behaviours are difficult to test precisely because they depend on time, sequence and external file delivery — the three conditions a conventional test environment cannot control. So they are validated in production, by the finance team, after the fact.

11. Why Coverage Metrics Conceal Exposure

The reporting layer is the reason unknown risk survives review. Coverage is calculated against the set of scenarios that exist. Scenarios that were never written cannot lower a coverage figure.

What the dashboard reports

Test pass rate · Coverage percentage · Automation percentage · Defect counts

What the estate actually contains

Failure paths · State divergence · Dependency behaviour · Configuration drift · Degraded-mode behaviour

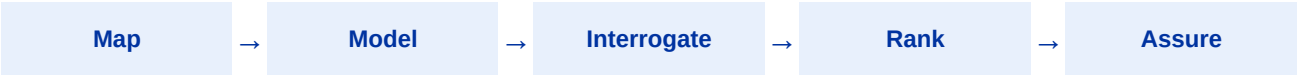
A 96% coverage figure and a critical unvalidated reversal path are entirely compatible statements. The metric is not dishonest — it is simply answering a different question from the one leadership believes it is asking.

The question the metric answers	The question leadership is asking
How much of our test inventory passed?	How much of our payment behaviour is proven?

12. Risk Discovery as an Engineering Discipline

If unknown risk is structural, then discovering it must be systematic. It cannot depend on the memory of experienced engineers, on incident history, or on the scope of the last audit.

PruTAN treats discovery as a defined method with five stages:



1. Map

Establish the real payment topology — journeys, interfaces, dependencies, protocols and configuration surfaces, as they exist rather than as documented.

2. Model

Derive the behaviours each journey must exhibit, including failure, reversal, duplicate, degraded and boundary behaviour.

3. Interrogate

Compare the modelled behaviour set against what has actually been validated, and expose the difference as named, specific unknowns.

4. Rank

Order the unknowns by payment risk — financial exposure, customer impact, journey criticality and change sensitivity.

5. Assure
Convert each ranked unknown into a deterministic, reproducible scenario that produces retained evidence — moving it permanently out of Zone 3.

13. RADAR: Making the Unknown Visible

PruTAN RADAR provides the payment-risk perspective that coverage metrics cannot. Rather than counting what exists, it reasons about what should exist.

Conventional reporting asks	RADAR asks
How many scenarios do we have?	Which payment behaviours have never been proven?
What is our coverage percentage?	Where does financial exposure concentrate?
Which tests failed?	Which critical journeys are currently unassured?
What changed in this release?	What payment behaviour could this change have altered?

The output is not a coverage number. It is a named inventory of exposure — the beginning of an unknown risk register that leadership can act on.

14. WISO: Asking What Has Never Been Asked

The hardest part of discovery is generative, not analytical. Someone, or something, must propose the conditions that no one in the organisation has yet considered.

PruTAN's WISO layer applies agentic payment intelligence to that problem. Reasoning across payment-domain knowledge and the target ecosystem, it can propose the scenarios, edge conditions, failure conditions and integration risks that a requirement-derived test inventory will never contain.

WISO proposes	Deterministic execution proves
The conditions worth asking about — including the ones no one has raised.	Whether the behaviour holds, with repeatable and retained evidence.

Intelligence widens the question set. Determinism keeps the answers trustworthy. Neither is sufficient without the other.

15. Making the Unknown Reproducible

Identifying an unknown risk changes nothing until the condition can be produced on demand. This is where most discovery efforts stall: the exposure is understood, but no environment can create it.

PruTAN capability	What becomes reproducible
Hostbox	Approvals, declines, timeouts, reversals, invalid responses, delayed responses and specific payment states
API Virtualisation	Dependency degradation — slow, malformed, unavailable and behaviourally changed responses
StressLab	Correctness under load, spike, sustained traffic and dependency stress
Trace	Transaction-level visibility that explains what the system actually did, and why

Together these convert a hypothesis about exposure into a controlled, repeatable condition — the difference between suspecting a risk and proving its absence.

SCENARIO — REFUND AFTER PARTIAL REVERSAL

Given:
 Authorisation approved for \$500
 Partial reversal of \$200 accepted

When:
 A full refund is requested before settlement

Then:
 Refund is limited to the net captured amount
 No negative balance is created
 Ledger, switch and gateway state agree
 Reconciliation reflects a single net movement
 Evidence is retained for audit

16. The Unknown Risk Register

The practical output of discovery is a register that states, per payment capability, whether behaviour is assured, partially assured or unvalidated — in language leadership can act on.

Payment capability	Primary unknown	Assurance status
Authorisation	Late issuer responses	PARTIAL
Reversal	Reversal after timeout	UNVALIDATED

Payment capability	Primary unknown	Assurance status
Refund	Refund after partial capture	UNVALIDATED
Tokenisation	Token lifecycle edge cases	PARTIAL
Fraud integration	Degraded-mode fallback	UNVALIDATED
Routing & configuration	Rule change without validation	UNVALIDATED
Settlement	Cut-off boundary handling	PARTIAL
Chargeback & dispute	Representment sequencing	ASSURED

The value of the register is not the red entries. It is that the estate now has a vocabulary for exposure that is neither a defect count nor a coverage percentage.

17. Ten Questions Payments Leadership Should Ask

These questions are diagnostic. An estate with mature assurance can answer them with evidence; an estate carrying unknown risk will answer them with assumption.

- Which payment journeys are business-critical, and which of their failure paths have been validated?
- What happens to a transaction when the issuer responds after we have already timed out?
- Can we produce a reversal, a duplicate and a partial approval on demand, in a controlled environment?
- Do our gateway, switch, ledger and core banking systems agree on transaction state after an interrupted payment?
- When a fraud, tokenisation or 3DS dependency degrades, what is our proven behaviour — not our intended behaviour?
- Which configuration changes can alter payment behaviour without passing through validation?
- Does payment correctness hold at peak volume, or only throughput?
- Which settlement and reconciliation edge cases have never been exercised before production?
- For last month's highest-risk change, what evidence do we hold that dependent behaviour still works?
- If an incident occurred tomorrow, would we discover it — or would a customer?

The pattern of answers is itself the finding. Questions that cannot be answered with evidence are, by definition, Zone 3.

18. From Unknown Risk to Measured Assurance

Discovery is only the first movement. The objective is a durable measure of how much of the payment estate is actually proven — and a trend line that shows exposure reducing over time.



PruTAN expresses this through the Payment Assurance Index (PAI), which combines critical journey assurance, risk coverage, integration assurance, resilience assurance, change assurance, scenario freshness and production exposure into a single leadership measure.

Instead of reporting	Payments leadership can see
96% coverage, 27 open defects	Critical journeys assured: 9 of 12 · Unvalidated high-risk behaviours: 5 · Exposure trend: improving · Payment Assurance Index: 74

The precise model can be organisation-specific. What matters is that Zone 3 becomes a number that moves, rather than a category that surfaces only through incidents.

19. What Changes When Discovery Becomes Routine

Organisations that adopt systematic risk discovery report a consistent shift in where engineering effort lands and how failure is experienced.

Before	After
Risk is discovered in production	Risk is discovered in a controlled environment
Incident review generates scenarios	Discovery generates scenarios before incidents
Coverage is the reported measure	Proven behaviour is the reported measure
Failure paths are exercised by customers	Failure paths are exercised deliberately
Configuration changes are invisible	Configuration changes are assessed for impact
Assurance is a release gate	Assurance is continuous across change and BAU
Leadership sees activity	Leadership sees exposure

20. Conclusion

Payment organisations do not fail because they test badly. They are exposed because the questions that matter most are the ones no one has thought to ask — and because the conditions that would answer them cannot be produced in a conventional environment.

Unknown risk is therefore not a QA problem to be delegated. It is an operational exposure that belongs to payments leadership, and it responds to the same treatment as any other material risk: identify it, name it, rank it, evidence it, and track it until it is closed.

PruTAN exists to make that possible — combining payment-domain intelligence to widen the question set, deterministic simulation to reproduce the conditions, and retained evidence to prove the behaviour. The outcome is not more testing. It is less unknown.

You cannot assure what you have never asked. Discovery is where payment reliability begins.

THE PRINCIPLE BEHIND THIS REPORT

To speak with a PruTAN advisor, contact:

info@prutan.com | prutan.com | **WhatsApp: +65 8839 4410**