axiom
corestream grc

# A practical step-by-step guide to the Third-Party Risk Management lifecycle

# Introduction

Third parties keep modern businesses running. Vendors host systems, process data, deliver critical services, and sit inside day-to-day operations. That reality creates two truths at once:

1. Third parties help you **move faster.**
2. Third parties **multiply your exposure.**

The problem is not that teams do not understand the risk. The problem is that a lot of third-party risk management (TPRM) programs were built for a simpler world. They struggle with scale, speed, and **proof**.

And when something goes wrong, nobody asks what you intended. They ask what you can prove.

This guide walks through a modern third-party risk management program end to end: what to capture, how to tier risk, how to right-size due diligence, what continuous monitoring really means, how to handle exceptions, and what “good evidence” looks like when leadership, auditors, or regulators ask hard questions.

Why this matters now:
Incident data and regulator expectations are both pushing in the same direction.

Gartner reported that **45% of organizations experienced third-party related business interruptions over a 2-year window.**

So, if your program is still driven by spreadsheets, email chasing, and one overloaded reviewer, it will not hold up under pressure.

# What Third-Party Risk Management (TPRM) is, and what it isn't

Third-party risk management (TPRM) is the governance of risk across the full lifecycle of third parties. It covers onboarding through offboarding, with **consistent decision-making, monitoring, and evidence.**

Michael Rasmussen, GRC 2020
*"As GRC Pundit, Michael Rasmussen frames it, the point is governance of the extended enterprise, not a disconnected set of vendor checks."*

What it is not:

- Just security questionnaires
- A once-a-year review
- A procurement admin workflow
- A standalone cyber exercise

Of course, cyber matters, but a real third-party risk management program also covers:

- Service continuity and resilience
- Subcontractors and fourth parties
- Regulatory exposure and auditability
- Ethics, worker welfare, and modern slavery risk
- Data processing, privacy, and safety obligations

# The 3 real reasons most third party risk management fails

## 1. It's too manual, so it doesn't scale

These involve processes like **manual chasing, recording on spreadsheets** and relying on inbox approvals.

Staff spend half the week **chasing third parties for assessments, certificates, and updates.** Evidence ends up scattered across inboxes and shared drives. Fragmented tech stacks and manual work create redundancy and make compliance **harder to demonstrate.**

That is not a process.

## 2. It's too slow, so the business routes around it.

Speed matters because the business has deadlines. If third-party approvals take weeks, teams will do what teams always do: find a workaround.

That can look like:

- Using **"temporary" tools** that quietly become permanent
- Signing with vendors before review is complete, then **asking risk to bless it afterwards**
- **Splitting the scope** into smaller pieces to avoid the tiering threshold
- **Reusing old due diligence** from a different service or engagement because "it's the same vendor"
- **Treating contract signature as the end of oversight**, when it's really the start

The risk here isn't only that unsafe vendors get through. It's that the organization loses visibility over what third parties are actually in the environment, what they can access, and what obligations they're operating under. Once that happens, third-party risk becomes hard to manage and easy to underestimate.

A practical rule: if your third-party risk management process is slower than procurement's need to buy, people will build a parallel process. And that parallel process won't be documented.

## 3. It's not provable, so when something goes wrong, nobody can reconstruct the decision trail fast enough.

Even if the team "did the work," the organization often cannot reconstruct:

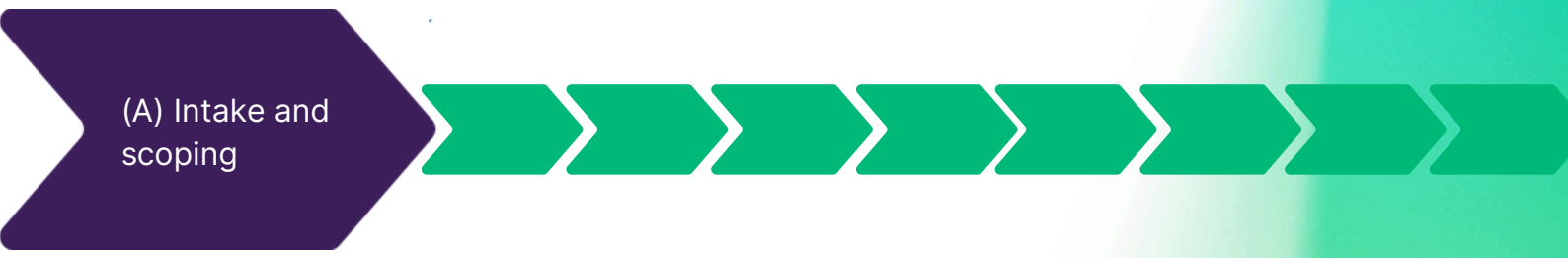
- why the vendor was approved
- what evidence supported the decision
- what exceptions were accepted and why
- what monitoring was in place
- what changed after onboarding
- what the organization did about it

This is why leading research in this area pushes for **clear roles, better information sharing, and monitoring**, is so relevant.

# The core model: the third-party risk management lifecycle

This section offers a clear, **step-by-step process to help you effectively assess and manage a third party**. We encourage you to tick off each step as you progress through the lifecycle.

| A third-party risk management program that holds up under scrutiny follows a simple lifecycle: | |
|---|---|
| A | Intake and scoping |
| B | Tiering and inherent risk |
| C | Due diligence (by tier, not one-size-fits-all) |
| D | Decisioning and exceptions |
| E | Contracting and go-live controls |
| F | Continuous monitoring |
| G | Issues, incidents, remediation |
| H | Offboarding and closure evidence |




# Intake and scoping: the foundation of third party risk management

If you get the intake wrong, every downstream step becomes inconsistent. The most common trap is treating “vendor” as the unit of risk. In practice, it is usually:

***vendor + service + access + data + geography + criticality***

Use this equation as a reference and think of intake as scoping the engagement so you can assess the right risk, not just collect admin details.

**Breaking down the equation: what to capture at intake:**

- ☐ **Vendor (who)**
  The vendor’s legal entity **name, where they operate, and who owns the relationship internally.** Why it matters: you cannot manage risk you cannot assign to a named owner.

- ☐ **Service (what)**
  What you are buying, the **specific use case**, and whether it changes an existing process or introduces a new one. Why it matters: the same vendor can be low risk for one service and high risk for another.

- ☐ **Access (how)**
  **Access level required** (none, limited, privileged) and how access **will be granted** (accounts, SSO, API keys, integrations). Why it matters: the access model is one of the fastest predictors of impact if something goes wrong.

- ☐ **Data (what they touch)**
  Data types involved (**personal data, regulated data, sensitive IP**) and whether data is **stored, processed, or only viewed**. Why it matters: this drives both regulatory exposure and the depth of due diligence required.

*Mark as completed* ☐

## (A) Intake and scoping

- [ ] **Geography (where)**
  **Where the service is delivered** and where **data is processed or stored**, including cross-border transfers. Why it matters: geography changes your legal, regulatory, and resilience expectations.

- [ ] **Criticality (what happens if they fail)**
  If the **service stops** for 24 - 72 hours, what breaks? Can you operate manually? Is there a **realistic fallback**? Why it matters: this prevents critical dependencies from being treated like routine suppliers.

Other considerations;

- [ ] **Fourth parties (who else is involved)**
  Any known **subcontractors, sub-processors**, or hosted components the service relies on. Why it matters: your exposure often sits one layer down, where you have the least visibility

This is where third-party risk management either holds up or reminds everyone it was mostly theatre. The organizations that perform best here tend to do 3 things consistently:

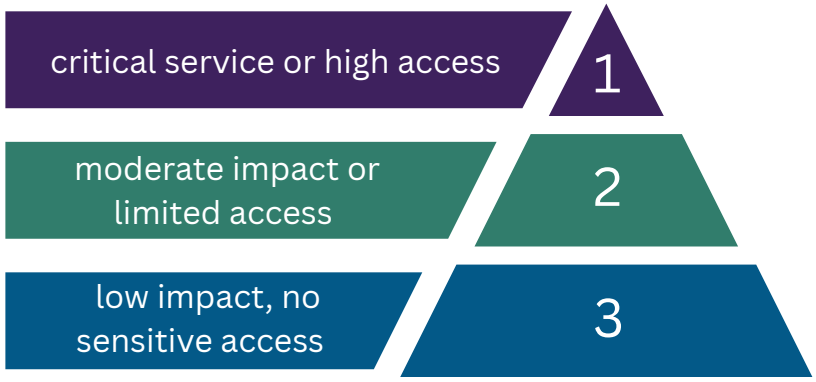
1. They assign **clear ownership**,
2. They share **cross-functional information** properly,
3. They treat **monitoring as ongoing** rather than point-in-time.

*Mark as completed* ☐

# Tiering and inherent risk (the part most orgs botch)

Tiering is not busywork. It is what makes the program **scalable and defensible.**

A pragmatic tier model:




Optional **restricted** category (sanctions, legal constraints, unacceptable red flags)

**What to include in scoring:**
Score each dimension 0–3 (0 = low, 3 = high), then map total score to tier thresholds:

- Data sensitivity
- Privileged access
- Operational dependence
- Regulatory exposure
- Concentration risk (single points of failure)
- Geography
- Fourth parties (sub-processors)

2 quick rules that make tiering usable:

- Tier the **engagement**, not the company name. Same vendor can be Tier 1 in one use case and Tier 3 in another.
- Make the tier logic **explainable** in one minute. If you cannot explain it quickly, the business will not trust it.

*Mark as completed* ☐

(C) Due diligence (by tier, not one-size-fits-all)

# Due diligence (by tier, not one-size-fits-all)

If you ask everyone for everything, you get 2 bad outcomes: vendor fatigue and low-quality responses. Remember: **streamlined questionnaires outperform exhaustive ones.**

**Baseline checks for everyone:**

These are your "non-negotiables," regardless of tier. They help confirm you know who you are dealing with.

- **Identity and entity validation**
- **Ownership and control basics**
- **Sanctions screening** (where relevant)
- **Adverse media review** (where relevant)

**Tier-based evidence (what to ask, and why):**
Tiering is what keeps your vendor risk management program scalable. The goal is not "more diligence." The goal is the **right diligence.**

☐ **Tier 3 (low impact, low access):**
Use this tier to keep the business moving while still **capturing essentials.**
Ask for:

- **Short questionnaire** tied to the service
- **Basic policy attestations** only where needed
- Clear escalation contacts and response expectations
- Why this works: Tier 3 should be **quick to approve, but still traceable**. You want **enough to justify the decision** without creating a two-week delay for a low-risk supplier.

*Mark as completed* ☐

## (C) Due diligence (by tier, not one-size-fits-all)

☐ **Tier 2 (moderate impact or limited access):**
This is where you confirm the vendor can meet expectations **without turning the process into an audit.**

Ask for:

- **Expanded questionnaire** aligned to the engagement
- **Certification evidence** where it's genuinely relevant
- **Business continuity summary** (how they recover and communicate)
- **Insurance confirmations** where appropriate
- Why this works: Tier 2 is where most organizations waste time with irrelevant demands. **Keep evidence tied to actual exposure. If you can't explain why you asked for it, don't ask**.


**Learn more about CoreStream GRC's TPRM solution**

☐ **Tier 1 (critical or high access):**
This is where **deeper assurance** is justified because the operational and reputational blast radius is real.

Ask for:

- **Deeper assurance evidence** (for example, SOC reports where relevant)
- **Incident response expectations** and notification timelines
- **Subcontractor transparency** (who else touches the service/data)
- **More frequent refresh cadence** defined up front
- Why this works: Tier 1 diligence should **reduce surprises later**. The point is to **understand control coverage**, **response capability**, and **dependency chains** before you are locked in.

**Domain solutions, based on vendor type:**

Build solutions you can attach based on the engagement:

- **Anti-bribery and corruption**
- **Cybersecurity**
- **Worker welfare and modern slavery**
- **Health and safety**
- **Privacy and data handling**
- **ESG requirements** where relevant

This is where supplier risk management software should feel practical, not punitive. You are not trying to "catch" vendors. **You are trying to make a confident, documented decision.**

*Mark as completed* ☐


# Decisioning and exceptions: the governance moment

Third-party risk management is not only about collecting information. It is about making decisions you can defend, consistently, and at speed. This is the point where a vendor risk management program either becomes trusted or becomes something the business tries to work around.

**Standardize decision outcomes:**
Keep outcomes simple and repeatable so approvals are comparable across teams and time:

- **Approved**
- **Approved with conditions** (approved, but specific actions are required before or during engagement)
- **Approved with exception** (approved despite a known gap, with risk explicitly accepted)
- **Rejected**

A quick rule: if reviewers are inventing new outcomes in emails, your process is already drifting.

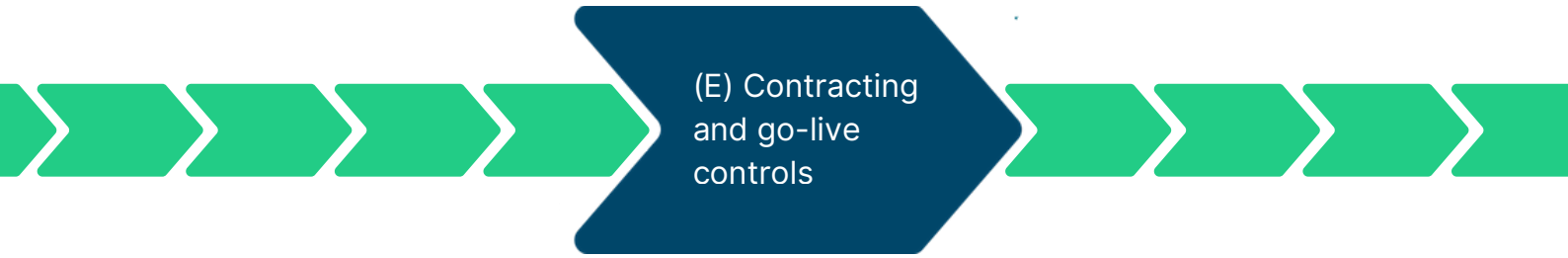
**Your exception register should be non-negotiable:**
Exceptions are not a failure. They are normal in the real world. The failure is accepting exceptions informally and losing track of what was accepted, why, and for how long.

Every exception should capture:

- **What failed** (the specific requirement not met)
- **Why it was accepted** (business justification, urgency, alternatives considered)
- **Compensating controls** (what reduces the risk in practice)
- **Approval authority** (who signed off and at what level)
- **Expiry date and review trigger** (when it must be re-checked or closed)

This is where third-party risk management solutions often fall apart. Exceptions get agreed in email, then forgotten. The risk does not disappear. It just becomes invisible, right up until the incident or audit that forces you to explain it.

*Mark as completed* ☐




# Contracting and go-live controls

A mature vendor risk management program treats the contract as part of the control environment, not a legal formality. This matters most for **Tier 1 relationships, where the operational impact of failure is serious** and you cannot rely on "good intentions" once the service is live.

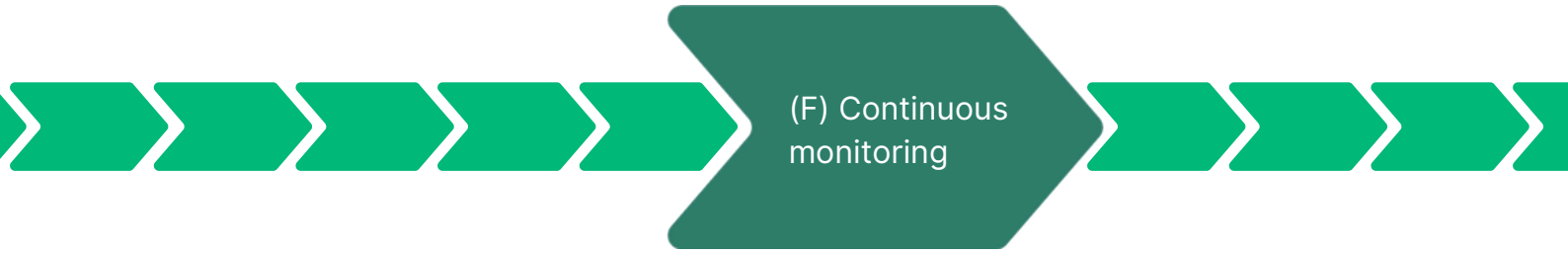
Contracting is also the point where you turn assessment findings into enforceable expectations. If a risk was identified during due diligence, this is where you either address it through contract terms, conditions to go live, or a documented exception.

☐ **A starting point: a practical clause checklist to align with legal:**

- **Incident notification timelines** (clear expectations, not vague "promptly")
- **Subcontractor disclosure and approval** (where feasible), including subprocessors that touch data or deliver the service
- **Minimum security and continuity expectations** tied to the service risk
- **Audit and assurance rights** that match your leverage and regulatory needs
- **Data return or destruction obligations** at termination, plus confirmation requirements
- **Exit support expectations and termination rights** so you are not trapped if performance drops or risk changes

If the contract does not match the risk tier, the approval decision is not real. It means you assessed risk, then chose not to control it.

*Mark as completed* ☐




# Continuous monitoring (where modern third-party risk management lives or dies)

Point-in-time reviews fail because third parties change. Ownership changes. Subcontractors change. Certifications expire. Security posture shifts. The service scope expands.

This is why ongoing monitoring is more effective than point-in-time efforts and is a core part of a mature program.

**What "continuous" actually means:**
It means two things working together.

☐ **i) Event-driven monitoring**

Monitor for signals that materially change risk:

- Sanctions and watchlist changes
- Adverse media signals
- Ownership or control shifts
- Cyber risk signals

This is where always-on data matters. For example, our partners, Black Kite is commonly used for external cyber risk signals, and Xapien supports due diligence and entity intelligence, depending on how you structure screening.

**Why we prioritize best-in-class partners**: continuous monitoring only works if the signals are credible and usable. Otherwise, you just generate noise, and teams become fatigued by notifications and start ignoring alerts.

☐ **ii) Time-driven refresh**

Set cadence by tier:

- Tier 1: more frequent refresh
- Tier 2: periodic refresh
- Tier 3: lightweight refresh

Also track control drift:

- Certification expiry
- Coverage changes (for example, SOC scope changes)
- Unanswered reassessments and overdue reviews

If you want a single sentence, keep it honest: continuous monitoring is mostly about not being surprised.

**Learn more about our partners**

*Mark as completed* ☐

(G) Issues, incidents, remediation

# Issues, incidents, remediation

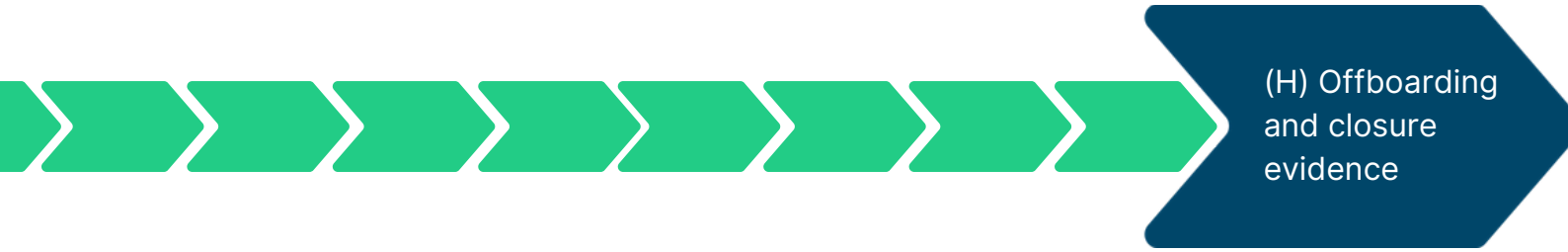
Monitoring without follow-through becomes theatre. The operational loop should be simple:

- **Intake the issue**
- **Assign an owner**
- **Set remediation plan and timeline**
- **Verify closure evidence**
- **Update risk rating if needed**
- **Report to leadership**

Your audit management software and internal audit teams will care about two things:

- Was the issue handled **consistently**?
- Can you show **evidence of closure**?

*Mark as completed* ☐




# Offboarding (the forgotten stage)

**Checklist:**

- [ ] Access removal confirmation
- [ ] Integration keys rotated
- [ ] Data returned or destroyed
- [ ] Subcontractors addressed if relevant
- [ ] Final attestations stored
- [ ] Lessons learned captured

**A real-world reminder of what happens when access isn't revoked:**
In Singapore, a former NCS employee was able to access a company system after termination due to what NCS described as "human oversight," and **later deleted 180 virtual servers.**

This is not an unusual scenario. It is a **basic offboarding failure: credentials and access** that outlive the relationship.

**And this is where the control model becomes real**. If your controls only exist as policies or intentions, you will not be able to prove what happened, who owned the action, or whether the right steps were completed at the right time.

That leads to the most important question for any third-party risk management program: if someone asked you today, **could you produce the evidence quickly?**

*Mark as completed* ☐

# The “Evidence Pack”

If you cannot produce these quickly, your program isn’t real. Not because you did nothing, but because you cannot demonstrate what was done, when, and by whom. That’s the difference between “we have a process” and “we can prove the process worked.”

**Minimum evidence set**

- **Third-party inventory** with owners and criticality
- **Tiering rationale** per engagement
- **Due diligence evidence and dates** (what you reviewed, and how current it is)
- **Exceptions** with approvals and expiry (including compensating controls)
- **Monitoring log and alert trail** (what changed, what was flagged, what you did)
- **Incident and remediation history** (actions taken and closure evidence)
- **Offboarding proof** (access removal, data handling confirmations, final attestations)

This **“prove it” mindset is the strongest bridge between third-party risk management and the wider GRC program**. It connects third-party oversight to enterprise risk reporting, audit readiness, and controls testing. In plain terms, it turns vendor risk management from a set of tasks into **governance you can stand over.**

# Metrics that actually matter (not vanity metrics)

Metrics matter because they **shape behavior.** If you only measure how many vendors you assessed, people will optimize for throughput, not quality. The best third-party risk management programs track a small set of operational metrics that tell you two things: **how quickly decisions move** and **whether risk is actually being managed over time.**

**Track outcomes that change behavior:**

- **Time to approve** (by tier)
  This is your friction indicator. If Tier 3 takes as long as Tier 1, your process is mis-sized.
- **Percent of third parties with overdue reviews**
  This is your visibility indicator. Overdue reviews usually mean you are slowly losing control of the inventory.
- **Exception volume and ageing**
  High exceptions can be normal. Old exceptions are the problem. Ageing tells you where accepted risk has become forgotten risk.
- **Remediation closure time**
  This is your effectiveness indicator. If issues never close, monitoring becomes noise.
- **Concentration exposure** (top critical vendors)
  This is your resilience indicator. It forces a real conversation about single points of failure.
- **Number of high-risk vendors with privileged access**
  This is your blast-radius indicator. It highlights where access risk and vendor risk overlap.
- **Incident response time for vendor issues**
  This is your maturity indicator. How fast you can coordinate and act matters more than how perfect your policy reads.

If leadership sees these metrics improving, **trust** in the program rises. If they see cycle time increasing while risk remains unclear, the business will **route around you.**

# How to implement without stopping the business

The fastest way to kill third-party risk management is to roll out a heavyweight process to everyone on day one. The goal is not to make procurement slower. The goal is to focus effort where it matters, build trust, and scale without creating a parallel shadow process.

An example of a practical rollout plan:

- **Start with critical vendors and high-access services**
  Begin where the impact is real. You get the biggest risk reduction with the least internal resistance.
- **Build tiering first, then right-size due diligence**
  Tiering is what prevents a one-size-fits-all workload. Without it, everything turns into Tier 1 by default.
- **Automate reminders and evidence capture early**
  This is how you stop living in spreadsheets. Expirations, reassessments, and approvals should not depend on someone remembering.
- **Keep exceptions visible, time-bound, and owned**
  Exceptions will happen. The discipline is making sure they expire, get reviewed, and don't turn into a permanent unknown risk.
- **Align procurement, legal, risk, and security so the business isn't double-asked**
  If vendors get the same questions from multiple teams, response quality drops and timelines blow out. One coordinated intake and one evidence record fixes that.
- **Treat monitoring as part of operations, not a compliance calendar event**
  Reviews should be triggered by risk and change, not just "it's been 12 months."

The simplest success test: the process should be **fast for low-risk suppliers, thorough for critical vendors,** and **consistent enough that anyone can explain why a decision was made.**

axiom corestream grc

If you're trying to run this kind of program with spreadsheets and email, you'll hit the same walls: chasing, slow cycle time, and scattered evidence.

CoreStream GRC's intuitive and flexible **third-party risk management software**, with our **AI integration partners**, is designed around the lifecycle above: onboarding through offboarding, with tailored assessments, continuous monitoring, integrations, and reporting so you can **prove decisions without rebuilding the story later.**

That's the difference between a program that exists on paper and a program that holds up under scrutiny and is **truly embedded into a dynamic GRC program that works for your business.**

# For more information please contact:

**Email us to book your demo:**
demo@corestreamgrc.com

corestreamgrc.com

**Book a demo**

**Learn more about CoreStream GRC's TPRM solution**

Follow us on LinkedIn