Simplified Total Risk Management (STORM): A Quantitative Risk Methodology at Qualitative Information Cost

Andrew T. Robinson
RESCOR LLC, Enfield, Maine, USA
atr@atra.us / arob@rescor.net

Abstract—Simplified Total Risk Management (STORM) lets an organization run any standard risk management framework — NIST 800-30, OCTAVE, ISO 27005, FAIR, COBIT — and replace that framework's qualitative or bespoke-quantitative internals with objective, repeatable, and comparable numeric measurements of risk, at essentially the same information cost as a conventional qualitative assessment. The mechanism is the quaLitative-to-quaNtitative (L2N) transition: a small set of STORM Transforms convert ordinary qualitative judgments about assets, threats, vulnerabilities, and controls into bounded scalars, and a diminishing-impact aggregation combines those scalars into a single comparable measurement at any level of the enterprise. STORM inherits its mathematics from RSK — a process developed by Robinson, Slobodzian, and Krikken between 1999 and 2007 — and is the modern, Transform-based evolution of RSK Risk Mode. This paper presents the public foundations (assumptions, measurement requirements, the L2N transition, and the framework mapping) so that an executive, auditor, or academic can evaluate the methodology without reference to proprietary internals. Specific transform constants and the diminishing-impact function's exact form appear in the non-disclosure (NDA) companion document.

Index Terms—Information security, risk assessment, security testing, penetration testing, vulnerability analysis, threat assessment, enterprise risk management, quantitative risk

A. Introduction

STORM is the descendant of a method first developed under the name RSK in 2000 and refined through 2007. The problem RSK was built to solve has not gone away — in fact, it has grown substantially worse. Every modern organization must make periodic judgments about how much risk it faces, how that risk is changing, and whether its security investments are paying off. The three options the industry has traditionally offered for answering those questions are all deficient in ways that STORM directly addresses.

1) The three traditional approaches: Quantitative risk assessments are grounded in the actual monetary value of assets, the severity of vulnerabilities, and the potential of threats. They produce defensible, decision-grade numbers — but at substantial cost in time and data collection, and the underlying monetary valuations are usually proprietary, which means the results of two quantitative assessments cannot be compared across organizations without breaching confidentiality.

Qualitative risk assessments substitute low-medium-high labels for monetary values. They are cheaper, but the labels carry wildly different meanings in different hands, and two qualified evaluators assessing the same environment routinely produce different qualitative results. Comparison across assessments is effectively impossible.

Security tests — vulnerability scans and penetration tests — assume a single hostile threat agent who is 100% effective at exploiting any discovered vulnerability, and assume every asset is of equal value. They are a useful indicator of risk, but they are not risk assessments: they conflate vulnerability with risk.

2) The gap STORM fills: STORM bridges the gap between these approaches. It delivers the direct-measurement advantages of a quantitative assessment without requiring proprietary monetary data to be shared. It delivers the consistency and repeatability that qualitative methods lack. And it turns the output of routine security tests into a measurement that can be tracked, compared, and aggregated — something the tests themselves cannot do.

3) Information cost is the point: The most important property of STORM is what it does not require. A traditional quantitative risk assessment demands monetary asset valuations, actuarial threat frequencies, and probabilistic loss-exceedance curves — data that most organizations cannot produce and cannot legally share even if they could. A qualitative assessment requires only ordinary business judgment about what matters, what could go wrong, and how bad it would be. STORM requires only the same ordinary business judgment that a qualitative assessment requires, and converts it into measurements with the analytical properties of a quantitative one. The same inputs a qualitative program is already collecting — informal severity, informal asset importance, informal threat likelihood — pass through the STORM Transforms and produce bounded, comparable, statistically analyzable numbers. The information cost of going from a qualitative program to a STORM program is near zero.

That is the entire value proposition. A STORM program is not more expensive than the qualitative program it replaces. It is not slower. It does not require new data. It

Manuscript revision 1.0 — April 2026.

does not force disclosure of proprietary valuations. It just produces outputs the qualitative program cannot produce: numbers you can compare, track over time, and aggregate up to a single board-level indicator.

STORM measurements can also be developed incrementally. An organization can begin with the information it already has and refine the measurement as additional information becomes available. Because the marginal measurement cost is low, STORM can be performed almost continuously — a property that converts risk assessment from an annual compliance exercise into an operating discipline.

B. Evolution: RSK/VM to RSK/RM to STORM

By 1998, commercial security firms — RESCOR's predecessor, NMI LLC, among them — were routinely assigning a numeric value between 1 and 10 to each vulnerability identified during security testing. The limits of this practice were quickly apparent: a simple average of vulnerability scores across a network of any significant size produced misleadingly low numbers, because the severe issues that actually characterize the risk were diluted by a long tail of trivial findings.

1) RSK Vulnerability Mode (RSK/VM): In 2000, NMI's principal investigators — [Andrew] Robinson, [Eugene] Slobodzian, and [Ramon] Krikken — developed RSK, the name being the shared initial of the three principals. RSK was a more sophisticated model that measured risk based on the observable characteristics of an asset base rather than on the raw scores of individual vulnerabilities. The initial version produced measurements based on vulnerability severity alone; threat potential and asset value were not considered. That version is referred to as RSK Vulnerability Mode (RSK/VM).

RSK/VM assumes: - all assets have the same value (a server adjustment factor was added later to handle obvious role differences); - there is a single hostile threat agent who is 100% effective at exploiting any identified vulnerability; - measurements are based on visible properties of the asset base, discoverable using standard security testing techniques.

RSK/VM is the right tool when the objective is a quick, repeatable indicator of technical risk from a security test. It provides more insight than a list of findings, but it cannot represent non-technical risks, and its fixed-threat / fixed-value assumptions mean it always overestimates technical risk and ignores business context.

2) RSK Risk Mode (RSK/RM): In 2004, RSK was expanded to support: - threat enumeration and assessment — probability and impact of specific threat agents acting on specific assets; - asset valuation — asset-specific values reflecting the business importance of each asset; - finding confidence — the degree to which the testing agency is certain a particular finding is genuine; - white-box / insider inputs — information not normally accessible to automated security tools.

That version is RSK Risk Mode (RSK/RM). RSK/RM addressed most of RSK/VM's weaknesses. Because threat and asset factors are probabilities between 0 and 1, RM measurements are substantially different from VM measurements of the same environment: only the genuinely high-impact, high-likelihood exposures contribute meaningfully to the risk total, while low-confidence or low-asset-value findings are correctly de-emphasized.

3) STORM: STORM is RSK/RM with the addition of specific Transforms to simplify asset valuation, threat assessment, vulnerability classification, and control evaluation. Everything RSK/RM did, STORM still does. What STORM adds is the tooling that makes the methodology usable at qualitative-program information cost:

- a full suite of STORM Transforms that standardize the construction of asset, threat, vulnerability, and control inputs from ordinary qualitative judgments (see the Transforms section);
- native support for enterprise risk — operational, financial, legal, reputational, compliance, and cyber risk all expressed on one comparable scale;
- expression of measurements as an approximate percentage of the asset at risk, which is more intuitive than the original RU scale for non-technical audiences;
- explicit treatment of the entropy / energy dynamic that makes risk a moving target;
- direct mapping onto NIST 800-30, OCTAVE, ISO 27005, FAIR, and COBIT.

When this paper uses "STORM" without qualification, it refers to the current framework. "RSK/VM" and "RSK/RM" refer specifically to the vulnerability-mode or risk-mode behavior — both of which remain available inside STORM as operating modes.

C. Requirements for Risk Measurements

Risk is not a physical quantity that can be directly measured. For the purposes of this paper, a risk measurement is an algorithmic approximation of risk based on the observable characteristics of a asset base. For that approximation to be useful, STORM requires that every measurement satisfy four properties.

Objective meaning. The interpretation of a measurement must be immediately apparent to a reader, regardless of their expertise. It should not require substantial supporting verbiage. A STORM measurement expressed as "43 RU" is roughly "43% of this asset is at risk" — a form that is intelligible to boards, executives, auditors, and technologists alike.

Repeatability. Any two measurements taken of the same asset base under identical conditions must be identical — even if two different testing agencies take them. A methodology that produces different answers in different hands is a methodology that describes the testers, not the environment.

Comparability. Any two measurements must be directly comparable, and the comparison must have meaning. Measurements of the same organization at different points in time, of two organizations in the same industry, or of two organizations in completely different industries must all be comparable. This also implies that STORM measurements can be statistically analyzed.

Scalability. The measurement of the asset base and the measurement of any subset of its components must be directly comparable. The measurement must remain within defined bounds regardless of the size of the asset base. A measurement of a single application running on a host is comparable to the measurement of that host, which is comparable to the measurement of the network it lives on, which is comparable to the measurement of the entire enterprise.

These four requirements sit above the methodology. Every design decision in STORM — including the specific form of the L2N transition described below — can be justified by the need to preserve at least one of these four properties.

D. Assumptions

STORM rests on a small number of explicit assumptions. They are intentionally conservative. Where they cannot be satisfied, the measurement is either not produced or is annotated with a confidence adjustment.

1) Assumptions common to all STORM measurements: Worst-case risk assumption. The overall risk to a asset base is greater than or equal to the risk associated with the most severe risk factor in that domain. The risk level of a system is always at least the risk of its weakest link. This assumption is why STORM measurements do not average away severe issues the way naive scoring systems do.

Multiple-exposure assumption. Risk increases as the number of risk factors increases. Multiple exposures provide multiple threat vectors and more opportunity for dependent exposures to chain together. Risk never decreases when a new exposure is added.

Diminishing-impact assumption. The incremental impact of each successive exposure is less than the impact of the one before it. Once a given risk level is already established by one severe exposure, the addition of a second exposure of the same class changes the aggregate risk less than the first exposure did. This is the property that keeps STORM measurements bounded no matter how long the list of findings grows — see the L2N Transition section.

Effective-discovery assumption. The discovery techniques used are as effective as technologically possible at enumerating the hosts, services, assets, and processes that make up the asset base, and at identifying the implementation names and versions of active services.

A historical note on the authoritative-database assumption. Earlier revisions of RSK carried a fifth assumption — that the underlying risk-factor database was authoritative and complete. When RSK was drafted in the early 2000s, no reliable public enumeration of technical vulnerabilities existed; the assumption was essentially a disclaimer ("data we don't have won't inform the measurement"). With the maturation of NVD, CVE, and the vendor advisory ecosystem, that assumption is now satisfied in practice for technical vulnerabilities and is not called out separately in contemporary STORM. It has never applied to non-technical risk factors, which are enumerated case-by-case.

2) Additional assumptions specific to RSK/VM: Visible-properties assumption. RSK/VM measurements are based on properties of the asset base that can be enumerated using standard security testing techniques. Internal architecture, physical security, operational practice, and similar non-visible properties are not considered.

Fixed-value assumption. All assets are assumed to have the same value. A later refinement added a role adjustment so that a server can be treated as more critical than a workstation, but RSK/VM does not differentiate among servers or among workstations.

Fixed-threat assumption. A single hostile threat agent is assumed, 100% effective at exploiting any identified risk factor — regardless of whether the testing agency itself can or will exploit it.

The low-impact assumption carried in earlier revisions of RSK has been deprecated as of RSK version 9 and does not appear in STORM.

3) RSK/RM and STORM relax these further: RSK/RM and STORM relax the three RSK/VM-specific assumptions: assets have specific values, threats have specific probabilities and impacts, and findings have specific confidences. The result is a measurement that reflects business context rather than a worst-case technical approximation.

E. Risk Measurement vs. Vulnerability Measurement: A Narrative

STORM's distinction between Risk Mode (RM) and Vulnerability Mode (VM) is the most frequently misunderstood aspect of the methodology. The best illustration of the distinction is narrative rather than numeric. The NDA companion document contains the explicit transforms; this section presents the same idea without the math.

Consider two hosts on a small business network.

Host A is the CFO's laptop. It holds and transmits financial records, tax filings, client banking information, and communications with legal counsel. The CFO travels frequently. The laptop has a full-disk-encryption policy but the CFO, in practice, leaves the machine suspended in her hotel room. A security test finds that the machine has two high-severity Windows vulnerabilities that are remotely exploitable over SMB, plus a handful of medium- and low-severity findings typical of a corporate laptop.

Host B is a dedicated print-queue management workstation in the facilities closet. It holds no confidential

data, exists on an isolated VLAN reachable only by the print servers and their administrators, and is physically inaccessible to anyone outside facilities staff. A security test finds that Host B has the same two high-severity Windows vulnerabilities as the CFO's laptop, plus roughly the same handful of medium- and low-severity findings.

In RSK/VM, these two hosts receive very similar measurements. The vulnerabilities are nearly identical; the measurement is based on visible properties only; and by VM's fixed-value and fixed-threat assumptions, both hosts face the same 100%-effective hostile threat and are assumed to be of equal value. A security test that stops at VM will prioritize them similarly.

In RSK/RM (and in STORM Risk Mode), the two hosts receive very different measurements. The CFO's laptop has a high asset value (financial and legal data concentrated on it), a realistic threat probability (a laptop that moves through hotel rooms is physically accessible to opportunistic threats and is a recognized target for targeted threats), and high confidence that the two remotely exploitable vulnerabilities can in fact be exploited given realistic network exposure. The facilities print-queue workstation has a low asset value (no sensitive data), a very low threat probability (isolated VLAN, facilities-only physical access), and low effective confidence that the SMB vulnerabilities can be reached at all by any threat that matters. The identical vulnerability list produces a very different risk number — because vulnerability is not risk.

This is the fundamental message: a vulnerability is a property of a system; a risk is a property of a system in context. RSK/VM measures the former. STORM Risk Mode measures the latter. Both are useful — VM gives you a fast, cheap indicator of technical exposure; RM gives you the measurement you can actually base a budget on — but they answer different questions and must not be confused.

F. The L2N Transition

The central design commitment of STORM is the L2N transition — the move from quaLitative to quaNtitative. This is the property that distinguishes STORM from the three traditional approaches described in the Introduction. A qualitative assessment stays qualitative; a quantitative assessment requires expensive monetary inputs before it can produce anything; STORM takes the same qualitative inputs a qualitative assessment uses — severity descriptions, asset descriptions, threat narratives, control narratives — and converts them into bounded numeric measurements suitable for comparison, aggregation, trending, and statistical analysis. Every measurement requirement and every assumption is realized through the L2N transition.

The L2N transition has two stages.

1) Stage A — qualitative input to bounded scalar: Every risk-relevant qualitative judgment in STORM is converted into a scalar between 0 and 1 (or an equivalent bounded integer) by a purpose-built transform. The transforms are the subject of the Transforms section:

- The Asset Transform turns the qualitative question "how important is this asset?" into a bounded value on a fixed scale.
- The Threat Transform (HAM533) turns the qualitative question "how likely is this threat, with what access, and with what means?" into bounded values for history, access, and means, which combine into a threat probability and impact.
- The Vulnerability Transform turns the qualitative question "how exposed is this asset by this weakness?" into a bounded exposure value, either directly (CVSSA for CVSS-scored technical findings), by estimation (SEM), or by structured scoring (CRVE).
- The Control Transform turns the qualitative question "how effective is this control?" into a bounded effectiveness value between 0 and 1.

Each transform is constructed so that the same qualitative input produces the same bounded output in different hands — the repeatability requirement — and so that outputs from different transforms can be combined in a common unit — the comparability requirement.

2) Stage B — variable-length list of scalars to single measurement: Once a asset base has been described through the transforms, it presents as a variable-length list of individual risk factors with scalar measurements. The second stage of the L2N transition is the aggregation of that list into a single bounded measurement.

Let a asset base (or any subset of it — an asset, a host, an application, a department, an enterprise) present a list of $n+1$ risk factors with individual measurements $v_0, v_1, \ldots, v_n$, ordered so that

$$v_0 \geq v_1 \geq \cdots \geq v_n \geq 0.$$

The aggregate STORM measurement x for the domain is given by

$$x \;=\; \sum_{i=0}^{n} f(i,\, v_i)$$

where $f : \mathbb{N} \times [0,\, v_{\max}] \to [0,\, \infty)$ is the diminishing impact function. f is constrained by five properties:

1. Non-negativity. $\forall\, i :\; f(i,\, v_i) \geq 0$.
2. Zero at zero. $\forall\, i :\; f(i,\, 0) = 0$. A risk factor with measurement zero contributes nothing.
3. Worst-case identity. $f(0,\, v_0) = v_0$. The largest risk factor contributes in full.
4. Monotone in the factor. $\dfrac{\partial f}{\partial v} \geq 0$. A worse finding never lowers the aggregate.
5. Strictly diminishing in rank. $\forall\, i, \forall\, v > 0 :\; f(i+1,\, v) < f(i,\, v)$. Equivalently, there exists a constant $\alpha \in (0, 1)$ such that $f(i+1,\, v) \leq \alpha \cdot f(i,\, v)$, which guarantees geometric convergence of the series:

$$\lim_{n\to\infty} \sum_{i=0}^{n} f(i,\, v_i) \;\leq\; x_{\max} \;<\; \infty.$$

Properties (3) and (4) preserve the worst-case assumption; (1) preserves monotonicity under additional factors; (5) captures the diminishing-impact assumption directly and guarantees the boundedness required by the scalability requirement stated in the Requirements section. The specific form of f — the tuning constants that determine α and the shape of the decay — is proprietary and appears in the NDA companion document.

For readers rendering the document without a math engine, the same formulas in ASCII are:

```
v_0 >= v_1 >= ... >= v_n >= 0
x = Σ (i = 0 to n) f(i, v_i)
f(i, v_i) >= 0,   f(i, 0) = 0,   f(0, v_0) = v_0
 f/ v >= 0
    (0, 1): f(i+1, v) <=   · f(i, v),  so  lim x <= x_max < ∞
```

3) What Stage B guarantees: Five properties follow directly from the shape of f, and they are what make STORM measurements behave the way the Requirements section prescribes:

- Worst case is preserved. Because f(0, v_0) = v_0 (the largest factor contributes in full) and every subsequent f(i, v_i) is non-negative, the aggregate x is always at least as large as the worst individual exposure.
- Monotonic in count. Because every f(i, v_i) >= 0, the aggregate never decreases when an additional risk factor is added. Adding a finding to a asset base cannot reduce its measurement.
- Diminishing incremental impact. Because f decreases in i, the second-worst factor contributes less than the worst, the third-worst less than the second, and so on. A hundred minor findings can never "average away" a single severe one; and stacking additional low-severity findings onto a domain that already has a severe one does not meaningfully move the measurement.
- Boundedness. Because f decreases fast enough, the infinite series converges. No matter how many risk factors are present, x remains within a defined range. A domain with ten thousand findings has a measurement in the same numeric range as a domain with ten findings — which is what makes cross-domain comparability possible in the first place.
- Scale-independence. The same aggregation applies at every level — a host, a group of hosts, a department, an enterprise. Sub-aggregates combine into super-aggregates through the same aggregation, which is why a host measurement is directly comparable to a network measurement or an enterprise measurement.

4) Visualization: Conceptually, f acts like a weighted tail-sum. The worst exposure sets the floor of the measurement; each successive exposure nudges the measurement upward, but by progressively less. The curve flattens quickly. This is why experienced STORM practitioners observe that a measurement is "controlled" by roughly the top half-dozen findings in any given domain — the rest of the tail is mathematically present but contributes less than the measurement precision.

Operators sometimes find this counterintuitive at first ("you mean fixing ten low findings won't move my score?"), but it is exactly the property that the comparability requirement demands. A measurement that changed significantly every time a low-severity finding was opened or closed would be unusable for trend analysis or cross-organization comparison.

5) Why the common alternatives fail: The question "why not just count / average / sum my qualitative findings?" is asked often enough that this section answers it directly. Each common shortcut fails at least one of the four measurement requirements.

Counting findings by level ("100 High, 400 Medium, 50 Low"). The most common informal dashboard in the industry. It has no objective meaning — "High" spans a range from "take me, I'm yours" to "inconvenient under the right conditions," and three counts collapse that range invisibly. It is not comparable: two organizations that each have 100 Highs may be in entirely different risk postures. It is not scalable: a bigger asset base tends to have larger counts at every level regardless of actual risk. And it gives no meaningful indication of progress until an entire qualitative level is eliminated, because moving from 100 Highs to 99 Highs tells you nothing about whether the remaining 99 are better or worse than the one that left.

Simple average of numeric qualitative levels. Assign L=1, M=2, H=3 and take the mean. The resulting number ("your risk is 2.1") is imprecise to the point of meaninglessness — and it averages severe findings away. A single "take me, I'm yours" finding is invisible in a sea of moderate findings. This fails objective meaning and comparability.

Sum of numeric qualitative level values. Add up the L=1, M=2, H=3 values across the asset base. The result grows with organization size and technological complexity, so a larger organization will always look riskier than a smaller one even when the smaller one is in materially worse shape. Fails comparability, scalability, and objective meaning simultaneously.

Maximum level. "Your risk level is the worst finding you have." Objective and repeatable, trivially satisfies the worst-case assumption, but not comparable across organizations (everyone with any High finding looks identical) and not scalable (it ignores how many Highs exist). Cannot show progress until every finding at the current maximum level is resolved.

Full quantitative risk assessment (monetary ALE, loss-exceedance curves). Objective and meaningful if the inputs can be obtained, but in practice the input requirements are prohibitive, the monetary valuations are proprietary and cannot be shared, and the results of two such assessments are generally not comparable across organizations.

The comparison is direct:

STORM is the only approach in the table that satisfies all four measurement requirements at qualitative-assessment information cost. The aggregation chosen for Stage B of the L2N transition — the diminishing-impact function f with the five constraints given above for Stage

Approach	Objective meaning	Repeatability	Comparability	Scalability	Information cost
Count by level (100 H, 400 M, 50 L)	No	Yes	No	No	Low
Simple average of level values	No	Yes	No	No	Low
Sum of level values	No	Yes	No	No	Low
Maximum level	Partial	Yes	No	No	Low
Full quantitative (monetary ALE)	Yes	Partial	No	Partial	Very high
STORM (L2N)	Yes	Yes	Yes	Yes	Low

TABLE I
Measurement Requirements vs. Common Approaches to Qualitative Aggregation

B — is the smallest mathematical commitment that makes this possible. The specific form of f is where the practical engineering lives, and is the portion of the methodology treated as proprietary.

G. STORM Transforms

A STORM measurement is only as good as the inputs fed into the L2N transition. The STORM Transforms are the standard procedures for constructing those inputs — one transform each for assets, threats, vulnerabilities, and controls.

1) Asset Transform: The Asset Transform converts business information about an asset into a quantitative value between 0 and 1, representing the relative share of the organization's value concentrated on that asset. Two methods are supported:

- Basic Criticality (BC) — assigns a 1-10 or 1-100 value. Simple, fast, dramatically more precise than qualitative labels, and sufficient for most assessments.
- Container-Content-Process (CCP) — separately scores the container (the infrastructure), the content (the information), and the processes the asset supports, using a maturity index from 0 to 5 on each. CCP provides more nuanced valuation for environments where a single label obscures meaningful differences.

2) Threat Transform — HAM533: The Threat Transform quantifies threat probability and impact using the History-Access-Means (HAM) model:

- History (H) — the frequency or likelihood of the threat occurring, informed by industry data, historical incident rates, and the threat's own motivation.
- Access (A) — the degree of access the threat agent has, or could readily obtain, to the threatened assets.
- Means (M) — the capability, resources, and technical sophistication of the threat agent.

The "533" refers to the default number of gradations on each axis. A HAM533 assessment is expressible as a small table and computable by hand or in a spreadsheet; the STORM software toolkit provides an interactive calculator. The HAM model is appropriate for adversarial threats, environmental threats, and operational failure modes alike.

3) Vulnerability Transform: The Vulnerability Transform converts vulnerability information into a quantitative value between 0 and 1 that represents the percentage of an asset exposed by the vulnerability. Three methods are supported, suited to different kinds of vulnerabilities:

- CVSS Adaptation (CVSSA) — for technical vulnerabilities with a Common Vulnerability Scoring System (CVSS) score. Transforms a CVSS score directly to the STORM exposure value (a CVSS 9.3 becomes 0.93).
- Simple Exposure (SEM) — for non-technical vulnerabilities where the exposure can be estimated as a percentage (1% to 100%) based on operational knowledge.
- Capability-Resource-Visibility-Effects (CRVE) — a more structured transform that models any kind of vulnerability by assigning 1-3 values to capability, resources, visibility, and the confidentiality-integrity-availability effects.

4) Control Transform: The Control Transform quantifies the effectiveness of a control at reducing the risk associated with a specific vulnerability. A control with an effectiveness value of 0.75 reduces the residual exposure by 75%.

Worked example: an asset has a value of 0.4, a threat probability of 0.5, and a vulnerability exposure of 0.5. The uncontrolled risk is 0.1 (10% of the asset's value). Applying a control with effectiveness 0.75 yields a residual risk of 0.025 (2.5%). The same structure supports "what-if" analysis — the likely impact of a proposed control can be measured before any money is spent on it.

5) Interoperability: All STORM Transforms produce values that feed the L2N aggregation described above. Individual transforms can be replaced with domain-specific variants without affecting the rest of the pipeline, and STORM inputs can be accepted from external tools via CSV, XML, YAML, or JSON.

H. ATRA — Advanced Total Risk Assessment

STORM lowers the information cost of producing a quantitative risk measurement to that of an ordinary qualitative assessment. Advanced Total Risk Assessment (ATRA) is the next step along the same curve: it is

STORM with AI-assisted population of the Transform inputs. Where STORM converts human qualitative judgment into bounded scalars, ATRA reduces the cost of producing that judgment in the first place — so the same L2N transition can be applied across a larger asset base, more frequently, and at less marginal analyst time per measurement.

1) What ATRA does: ATRA provides AI-driven candidate-generation for each of the four STORM Transforms:

- Asset Transform. ATRA ingests configuration management databases, cloud resource inventories, SaaS application registries, identity directories, and business-process documentation, and proposes asset classifications and valuations. Basic Criticality or Container-Content-Process scores are drafted from the evidence.
- Threat Transform (HAM533). ATRA ingests threat-intelligence feeds, historical incident records, sector-specific breach reports, and organizational context, and proposes History / Access / Means scores for the threat agents relevant to each asset class.
- Vulnerability Transform. ATRA ingests vulnerability-scanner output, penetration-test reports, public vulnerability databases (NVD, CVE, vendor advisories), configuration audits, and human-language findings, and proposes CVSSA, SEM, or CRVE exposure values.
- Control Transform. ATRA ingests policy documents, control-test evidence, audit reports, SOC 2 / ISO 27001 artifacts, and log-based monitoring data, and proposes control-effectiveness values.

2) What ATRA does not do: ATRA is a population tool, not a replacement for the methodology. It does not:

- change the STORM Transforms or the L2N aggregation;
- replace human review of values used in decision-making;
- assign numbers that have not been approved by a human analyst with authority over the asset base in question;
- eliminate the need for insider knowledge — in fact, it amplifies the value of insider inputs by freeing analyst time to focus on them.

Every AI-proposed value passes through the same Transform and aggregation pipeline as a manually-assigned value, and every value carries provenance metadata recording whether it was AI-proposed, human-authored, or human-approved AI.

3) Why ATRA preserves STORM's auditability: A well-founded objection to AI-driven risk tooling is that the resulting numbers are opaque. ATRA avoids this in a structural way: the AI layer produces Transform inputs, not risk outputs. The Transforms and the aggregation remain deterministic, inspectable, and reproducible. An auditor or engagement partner can, for any ATRA-produced measurement:

1. See the full list of contributing risk factors and their scalar values.
2. See which values were AI-proposed, which were human-authored, and which were human-approved.
3. See the evidence (documents, feeds, scans) from which each AI-proposed value was derived.
4. Recompute the measurement from the scalar values using the published L2N transition.

The AI is a producer of candidate qualitative judgments. The methodology is the same STORM methodology this paper describes.

4) Confidence carries forward: RSK/RM introduced a confidence factor representing the degree of certainty in a given finding. ATRA uses that same mechanism to represent AI confidence: values proposed with low AI confidence (weak supporting evidence, ambiguous context, or model self-reported uncertainty) are weighted accordingly until a human reviews them. A measurement produced predominantly from high-confidence, human-approved values is stronger than one produced from low-confidence, unreviewed AI proposals — and STORM makes this visible through the confidence-weighted aggregation, without any new mathematics.

5) Relationship to STORM: ATRA is not a fork. Any STORM measurement can be produced with or without ATRA assistance, and the results are directly comparable because they pass through the same Transforms and the same aggregation. An organization can adopt ATRA incrementally — starting with, for example, AI-assisted vulnerability classification while continuing to populate asset and threat inputs manually — and extend the AI layer as confidence builds. ATRA is available from the author at atr@atra.us.

I. Entropy and Energy

Risk is not static. It changes over time under two opposing forces.

Entropy is the natural tendency of risk to increase. New assets are deployed. New vulnerabilities are discovered in existing software. New threat capabilities emerge. Employees come and go. Business processes change. Configurations drift. If nothing is done, measured risk always rises.

Energy is the money, time, and attention spent on security, governance, and compliance. Patches are applied. Training is delivered. Policies are updated. Controls are tested. Drift is corrected. Energy expenditure reduces measured risk.

Every organization is, at every moment, either: - applying more energy than entropy is generating — in which case risk is falling; - applying less energy than entropy is generating — in which case risk is rising; - or at an equilibrium that is either acceptable or not.

Qualitative risk methodologies cannot distinguish these cases. STORM can. By measuring quantitatively and continuously, STORM makes the rate of change of risk visible.

The question "are we winning or losing?" becomes a question with a specific, measurable answer. The question "is this program worth the money?" becomes a question of whether the measured slope of risk over time is negative, zero, or positive.

This is the reason the author states flatly that risk management must be continuous, not a one-time activity. An annual qualitative assessment, however well executed, captures a single point and says nothing about slope.

J. Framework Compatibility

STORM does not replace existing risk management frameworks. It maps directly onto them, replacing the framework's qualitative or bespoke quantitative internals with the STORM Transforms and L2N aggregation, while preserving the framework's process structure and compliance posture.

An organization already compliant with NIST 800-30 can adopt STORM without re-architecting its risk program. The compliance paperwork remains; the inputs become quantitative; the outputs become comparable.

K. Relative Risk Levels

STORM measurements are expressed in Risk Units (RU) — bounded positive integers, with larger values indicating greater risk. An RU can also be read as an approximate percentage of the asset at risk.

The following two tables are provided for organizations that need to map continuous measurements back to categorical labels (for example, to satisfy a framework or reporting requirement that insists on "low / medium / high"). The first is a five-level mapping and the second is a three-level alternate.

Neither mapping is normative. STORM's position is that an organization should define its own thresholds based on its own risk tolerance and reporting obligations, because "low" and "high" are risk-tolerance statements, not measurements. The two mappings above are provided only as defaults.

L. Application and Deployment

STORM is delivered as:

- A software development toolkit (SDK) for Node.js and Python.
- REST APIs for integration with applications written in any language.
- Batch import/export in CSV, XML, YAML, and JSON, for interoperability with spreadsheets, GRC platforms, vulnerability scanners, and SIEM/SOAR tooling.
- Interactive calculators — the HAM533 threat calculator and an aggregated risk explorer are available at rescor.net/storm/.
- Engineering services — RESCOR provides integration services for STORM deployment into existing business processes and applications on platforms from mobile to mainframe.

Reporting is hierarchical. Individual asset, threat, and vulnerability measurements aggregate through the L2N transition into host, application, departmental, business-unit, and enterprise measurements. The same mechanism supports drill-down from a single enterprise-level number to the specific remedial priorities beneath it.

M. Conclusions

STORM — and its RSK ancestor — have been applied in thousands of assessments since 1999 across financial services, transportation, healthcare, energy, government, and technology sector clients. The methodology consistently:

- Identifies the risk levels of arbitrary asset bases — from a single application to an entire enterprise — on a single comparable scale.
- Reflects the effects of change in the asset base. Remedial actions reduce the measurement; new exposures raise it. The measurement lacks sensitivity to small individual changes by design (worst-case and diminishing-impact), but responds meaningfully to changes in the dominant exposures.
- Supports direct comparison across domains and across organizations — even across industries — because the assumptions, the requirements, and the L2N transition are the same in every case.
- Makes the entropy/energy dynamic visible over time. Organizations can see whether they are winning.
- Does not require proprietary monetary data to be shared in order for results to be compared externally — the output is a bounded, dimensionless measurement.
- Maps onto every mainstream risk management framework without requiring the framework to be abandoned.

The sensitive details — the specific form of the diminishing impact function f, the precise transform constants, the risk-factor database schema, and the case studies containing client data — are covered in the non-disclosure companion document. The foundations presented here are, by contrast, intended to be freely discussed with prospective clients, auditors, academic collaborators, and the broader risk management community.

Framework	Framework Step	STORM Equivalent
NIST 800-30 / RMF	Prepare for risk assessment	STORM Asset Transform
	Identify threat events & likelihood	STORM Threat Transform (HAM533)
	Identify vulnerabilities	STORM Vulnerability Transform
	Determine risk	STORM Risk Calculation (L2N)
	Develop response strategy	STORM Control Transform
OCTAVE	Identify critical assets	STORM Asset Transform
	Identify threats	STORM Threat Transform (HAM533)
	Identify vulnerabilities	STORM Vulnerability Transform
	Develop protection strategy	STORM Control Transform
ISO 27005	Context establishment	STORM Asset Transform
	Risk identification	STORM Threat + Vulnerability Transforms
	Risk analysis	STORM Risk Calculation (L2N)
	Risk evaluation	STORM Aggregation & Reporting
	Risk treatment	STORM Control Transform
FAIR	Identify scenario & assets	STORM Asset Transform
	Evaluate loss event frequency	STORM Threat Transform (HAM533)
	Evaluate vulnerability	STORM Vulnerability Transform
	Derive risk	STORM Risk Calculation (L2N)
COBIT	Identify risk	STORM Threat + Vulnerability Transforms
	Analyze risk	STORM Risk Calculation (L2N)
	Maintain risk profile	STORM Aggregation & Reporting

TABLE II
STORM Mapping to Mainstream Risk Management Frameworks

Risk Level	Low RU	High RU
Low	0	24
Moderate	25	49
High	50	74
Very High	75	99
Extreme	100	—

TABLE III
Five-Level Relative Risk Level (RRL) Mapping

Risk Level	Low RU	High RU
Low	0	39
Medium	40	69
High	70	—

TABLE IV
Three-Level Alternate Relative Risk Level (ARRL) Mapping

A. Appendix A — Terminology

Asset. Anything of value to the organization. Assets are physical (buildings, equipment), digital (data, software, systems), and intangible (reputation, relationships, personnel).

ATRA (Advanced Total Risk Assessment). STORM with AI-assisted population of the Transform inputs. ATRA proposes candidate qualitative judgments for assets, threats, vulnerabilities, and controls; those judgments pass through the same STORM Transforms and L2N aggregation as manually-authored values. ATRA is a population tool, not a change to the methodology.

Asset base (formal synonym: testing domain). The subset of assets that are to be evaluated. Asset base is the preferred term in STORM practice; testing domain is the original RSK term of art and appears in legacy documentation and academic references. The two terms are interchangeable.

Compromise. An asset is compromised when a threat is able to exploit a vulnerability to affect the confidentiality, integrity, or availability of that asset.

Control. A countermeasure that reduces the likelihood of an exposure being exploited, or reduces the impact if it is.

Dependent vulnerability. A vulnerability that cannot be exploited unless a related, primary vulnerability has first been exploited.

Entropy (in STORM). The natural tendency of measured risk to increase in the absence of intervention.

Energy (in STORM). The investment in security, governance, and compliance activities that reduces measured risk.

Exploit. A specific procedure used by a threat agent to compromise an asset by taking advantage of one or more vulnerabilities.

HAM533. The History-Access-Means threat assessment transform, defaulting to 5 history gradations, 3 access gradations, and 3 means gradations.

Information system. Any set of information processing resources — computers, software, network infrastructure, and the data they carry.

L2N transition. The quaLitative-to-quaNtitative transition — STORM's central design commitment. A two-stage process that first converts qualitative inputs (asset, threat, vulnerability, and control descriptions) into bounded scalar values via the STORM Transforms, then aggregates a variable-length list of those scalars into a single bounded measurement via a diminishing impact function of the form x = sum(i=0..n, f(i, v_i)).

Primary vulnerability. A vulnerability exploitable by a threat without first compromising any other host or service.

Risk. The loss potential of a specific combination of asset, threat, and vulnerability.

Risk assessment. Any process for determining the risk to an asset or group of assets as a function of threats and vulnerabilities.

Risk factor. Any factor — asset value, threat potential, or vulnerability — that contributes to risk.

Risk Mode (RSK/RM, STORM-RM). STORM operation using full asset, threat, vulnerability, and confidence inputs.

Risk Unit (RU). STORM's unit of measurement. A bounded positive integer corresponding approximately to the percentage of the asset at risk.

Security test. A restricted form of risk assessment using a fixed-threat, fixed-asset-value assumption. Examples: vulnerability scans, penetration tests.

STORM. Simplified Total Risk Management. The modern form of the RSK methodology.

Testing agency. An entity authorized by the owner or custodian of the assets in the asset base to evaluate risk.

Threat. Any process, entity, or event with the potential to affect the confidentiality, integrity, or availability of the assets in the asset base. Threats include intentional and accidental human activities and natural events.

Vulnerability. Any problem or condition that exposes an asset to a threat. For information systems, vulnerabilities include software errors, configuration errors, and poor operational or management practices.

Vulnerability Mode (RSK/VM, STORM-VM). STORM operation using fixed-value and fixed-threat assumptions. Appropriate for security tests; produces a conservative technical risk indicator.

B. Appendix B — Version History of the Methodology

For the non-disclosure companion document containing the specific form of the diminishing impact function, transform constants, and client case studies, contact atr@atra.us or arob@rescor.net.

Year	Milestone
1999	Initial experiments with vulnerability scoring at NMI LLC
2000	First algorithmic model (RSK) replaces simple averages
2004	RSK expanded to Risk Mode (RSK/RM) — threat, asset, confidence
2007	RSK draft revision 9 — low-impact assumption deprecated
2015	Methodology rebranded as Simplified Total Risk Management (STORM)
2018	STORM Transforms formalized (Asset, HAM533, Vulnerability, Control)
2022	STORM software development kit and REST API released
2024	Framework mapping published (NIST 800-30, OCTAVE, ISO 27005, FAIR, COBIT)
2026	ATRA (Advanced Total Risk Assessment) — AI-assisted population of STORM Transform inputs
2026	This white paper (Revision 1.0)

TABLE V
Methodology Version History