
FRAMEWORK · COMPANION · V1.0

A variety-engineering vocabulary for GRC

Bearing east · Third piece · PDCA+ v2.0 · Public Review

Nine terms from the cybernetic tradition, defined precisely and applied to management-system integration — a working diagnostic kit for practitioners who need to say why an integration is failing in a way that points to where to intervene.

AUTHOR

Joacim Brandell

Written for GRC practitioners · worked examples throughout · no prior cybernetics background assumed

Why a vocabulary

GRC has no shared structural vocabulary for talking about why integration is hard. The standards give us control names and clause numbers; the platform vendors give us product names; the integration discourse gives us process language. None of it lets a practitioner say, in a way that points to where to intervene, why a particular integration is failing.

"The disciplines aren't talking to each other" is a complaint. "The hand-off is broken" is a description. "We need better governance" is a wish. None of these names what is structurally wrong, and none of them tells a practitioner what kind of intervention would actually help. The result is integration work that consists of stitching things together by hand, hoping the stitching holds, and stitching again when it doesn't.

The cybernetic tradition has been refining a vocabulary for these questions since the 1950s. It is not the only vocabulary that could work, but it has the advantage of being precise enough to support diagnosis and old enough to have been tested against a wide range of integration problems across many domains. The terms collected in this piece are nine of them — the subset most useful for management-system integration, defined in plain language with worked examples drawn from GRC practice.

Each entry has the same shape. A term definition in plain language. A short note on what the term means specifically for GRC. A diagnostic question the term lets a practitioner ask. A worked example showing the question being applied. The piece closes with a single integration failure walked through three times, each time using the vocabulary to sharpen the diagnosis.

This is not a glossary to memorise. It is a kit to keep working with. The terms become useful through use, and the diagnostic questions are the use. A practitioner who walks away with three of the nine in active vocabulary is doing better than a practitioner who can recite all nine but doesn't reach for them when the next integration problem arrives.

The nine terms

1. Variety

DEFINITION

The number of distinguishable states a system can be in. A coin has variety two (heads, tails); a six-sided die has variety six; a server's TLS configuration has variety in the thousands or millions depending on what counts as a distinguishable state. Variety is always relative to the observer — what counts as distinguishable depends on what the observer can perceive and what difference is meaningful.

In GRC, variety shows up everywhere once you start looking for it. The set of risks an organisation faces has variety. The set of configurations a system can be in has variety. The set of incidents that can occur has variety. The set of regulatory obligations across

jurisdictions has variety. Each is countable in principle and usually uncountable in practice, but the fact that the number is large is the point — variety is what makes integration hard.

Diagnostic question — *What is the variety of the thing being managed, relative to the variety of the apparatus managing it?*

Take a configuration management programme covering a thousand servers across three environments, six operating system variants, and twelve major software stacks. The variety of the managed estate is the product of those dimensions — easily several orders of magnitude beyond what any human team can directly observe. The variety of the configuration-management apparatus — the policies, the tooling, the team — has to match this somehow, and "somehow" is where the engineering work lives. Diagnosing the mismatch first is what the term gives you.

2. Requisite variety

DEFINITION

Ashby's law, formulated in 1956: any system that regulates another system must have at least as much variety — as many distinguishable states — as the system it regulates. A regulator with less variety than the regulated system will inevitably miss states the regulated system can occupy, and the regulation will fail at those states. The law is mathematically simple and operationally severe.

In GRC, requisite variety is the law that explains why hiring more competent people doesn't fix integration problems. The variety of cross-discipline findings scales combinatorially with the number of bound disciplines; a GRC team scales linearly. The mismatch is structural. No amount of diligence will close it, because diligence is not what the situation lacks — variety is.

Diagnostic question — *Does the regulator have enough variety to handle the variety of what it is regulating? If not, where is the gap, and is it a gap in attenuation, amplification, or both?*

When an organisation says "we need to hire another risk analyst" after a near-miss, the implicit theory is that the regulator's variety is sufficient and the problem is execution. Often that theory is wrong. The variety of cross-system findings genuinely exceeds what any human regulator can hold, and the right intervention is not to add another analyst but to change the structure of the regulation — usually by adding attenuation upstream (so fewer distinct findings reach the analyst), amplification of the analyst's reach (so one decision covers more cases), or both.

3. Attenuation

DEFINITION

Reducing the variety of a system so that a regulator with less variety can handle it. Attenuation works by collapsing distinct states into equivalence classes that the regulator treats the same way. A traffic light attenuates the variety of approaching vehicles to three states (go, slow, stop); a credit-score model attenuates the variety of an applicant's financial history to a single number; a configuration management taxonomy attenuates the variety of possible system states to a small number of named categories.

Attenuation is the most common move in GRC, and it is usually invisible because it has already happened by the time a practitioner sees the work. The standards do attenuation; the policies do attenuation; the controls do attenuation. Every control framework is a massive attenuation exercise — collapsing the variety of possible operational states into the much smaller variety of "compliant" and "non-compliant."

***Diagnostic question** — Where in this process is attenuation happening? Is it happening at the right place, and is the equivalence class it produces actually equivalent for the purpose the regulator has?*

Attenuation can fail in two ways. It can be too aggressive — collapsing states that should have been distinguished. The TLS-finding example from the PDCA+ whitepaper is exactly this: "configuration deviation" and "change-authorisation failure" were being treated as equivalent when they were not, and the attenuation hid an important difference. It can also be too weak — preserving distinctions the regulator cannot act on. A risk taxonomy with two hundred categories that the regulator can only respond to with three actions is over-distinguishing. Either failure produces visible symptoms, but the diagnosis is structural: the equivalence classes are wrong for the regulation.

4. Amplification

DEFINITION

Extending the regulator's reach so that a single regulatory action covers more of the regulated system's variety. Where attenuation reduces the variety the regulator has to handle, amplification multiplies the effect of each regulatory action. A vaccine amplifies a single immunological decision across the recipient's lifespan; a policy document amplifies a single committee decision across thousands of operational situations; a translation rule amplifies a single classification across every downstream system that consumes it.

In GRC, amplification is what makes the conjunction's substrate worth building. One translation rule between ISO 27001 and NIST RMF vocabularies serves every discipline that crosses the boundary; one control-management capability is consumed by every bound discipline; one taxonomy classifies findings for every downstream consumer. The substrate amplifies the regulator's variety without requiring the regulator to grow.

***Diagnostic question** — Where is the regulator's reach being amplified? Is the amplification structural (built into the framework) or reconstructive (rebuilt every time)?*

Reconstructive amplification is the failure mode the PDCA+ whitepaper diagnoses most directly. When a GRC team translates between ISO 27001 and NIST RMF vocabularies in their

heads every time a cross-regime question comes up, the translation is being amplified per-conversation rather than per-rule. The same amplification work is being done over and over. Structural amplification — the translation rule living in the substrate — does the work once and consumes it many times. The diagnostic question is whether amplification is happening structurally, and if not, why not.

5. Channel

DEFINITION

Anything that carries variety from one part of a system to another. A reporting line is a channel; a dashboard is a channel; a hand-off between teams is a channel; a schema specifying what one system can send another is a channel. Channels have direction (which way the variety flows), capacity (how much variety they can carry), and noise (how much of what they carry is irrelevant to the receiver).

GRC integration problems are very often channel problems. The disciplines exist, the work they do is adequate, the people are competent — but the channels between them are wrong. Either the channel is too narrow to carry the variety that needs to pass through it (capacity is inadequate), or it carries too much that the receiver cannot act on (signal-to-noise is wrong), or it goes to the wrong receiver (the routing is structurally incorrect).

***Diagnostic question** — What channels carry variety between the parts of this system? Are they the right channels, in the right direction, with the right capacity?*

When event management identifies an unauthorised configuration change and the finding reaches configuration management within minutes but reaches risk management three weeks later when someone notices the pattern, the diagnosis is a channel problem. The channel from event management to configuration management exists and has adequate capacity; the channel from event management to risk management either doesn't exist or has so much delay that it is functionally absent. The intervention is not to make event management work harder — it is to build the missing channel.

6. Channel capacity

DEFINITION

The amount of variety a channel can carry per unit time. A channel with high capacity transmits many distinguishable states quickly; a channel with low capacity transmits few states slowly, regardless of how many states the sender is trying to send. Channel capacity is the constraint that determines whether a system can regulate at the rate its environment changes.

In GRC, channel capacity shows up as latency, throughput, and fidelity. A quarterly review board is a low-capacity channel — it transmits a small number of decisions over a long time. A continuous monitoring dashboard is a high-capacity channel — it transmits many states quickly. Both have their place, but mistaking one for the other is a common diagnostic failure. Trying to regulate a real-time operational problem through a quarterly review channel will fail not because the people on the review board are inadequate but because the channel does not have the capacity for the variety the problem is producing.

Diagnostic question — *Is the channel's capacity adequate to the variety it is being asked to carry? If the channel is saturating, what variety is being lost, and what is the cost of losing it?*

A channel that saturates does not announce its saturation. It silently drops variety, and the receiver continues to act as if the channel were transmitting fully. This is one of the most consistent failure modes in GRC: an executive who reports that they are "comfortable with the risk posture" because they are not seeing concerning indicators, where the actual situation is that the indicators have been attenuated through several saturated channels before reaching them. The variety did not reach the regulator; the regulator does not know it did not reach them. The diagnostic move is to ask, at every channel along the path, whether the channel's capacity matches the variety being sent into it.

7. The regulator–regulated relationship

DEFINITION

The structural relationship between a system that is being regulated and the system that is regulating it. Naming the relationship explicitly — who is the regulator, who is the regulated, what variety flows in which direction — is often the first move that turns an undefined integration problem into a tractable one. Most integration failures involve some part of this relationship being implicit, miscast, or absent.

In GRC, the regulator–regulated relationship is often miscast. A common miscast is treating two disciplines as peers when one is structurally regulating the other. Configuration management regulates the system estate; change management regulates configuration management's authorisation pathway; risk management regulates the criteria under which both operate. These are not peer relationships, and treating them as peers produces predictable failures — usually visible as one discipline making decisions the other discipline should have informed.

Diagnostic question — *Who is the regulator and who is the regulated in this interaction? Is the relationship explicit, or is it being reconstructed by whoever happens to be in the room?*

The PDCA+ conjunction makes this relationship explicit through binding declarations and routing rules. Each binding declares what the discipline produces and what it consumes — which is, in cybernetic terms, declaring what variety the discipline sends and what variety it receives, and therefore which regulator–regulated relationships it participates in. The substrate does not have to guess; the routing is structural. The diagnostic value of the term is in seeing where the substrate has not yet been built, and where the relationship is being reconstructed by humans every time.

8. Variety mismatch

DEFINITION

The diagnostic shape of most integration failures. A variety mismatch occurs when the variety produced by one part of a system exceeds — or falls short of — the variety the receiving part can handle. Mismatches can occur at attenuation, amplification, channel capacity, or transduction. Naming the mismatch precisely is what turns "the integration is broken" into a sentence that points to where to intervene.

In GRC, almost every named integration failure can be re-described as a variety mismatch, and the re-description usually sharpens the diagnosis. "The risk register doesn't reflect what configuration management is seeing" is a variety mismatch between configuration management's output variety and the risk register's input variety. "The change advisory board is overwhelmed" is a variety mismatch between the variety of change requests and the channel capacity of the CAB. "Audit findings keep recurring" is often a variety mismatch between the variety of root causes and the variety of remediation actions the framework can express.

***Diagnostic question** — Where in this system is the variety mismatch? Is the producer over-producing for the consumer's capacity, or under-producing for the consumer's needs?*

The under-production case is easier to overlook and often more consequential. A risk register that receives findings from configuration management is supposed to receive enough variety to distinguish risks that warrant different treatments. If configuration management is producing a single "deviation detected" category for ten distinct underlying conditions, the risk register receives less variety than it needs, and risk decisions become coarser than they should be. The producer is under-producing; the consumer cannot detect this from the inputs alone. The diagnosis requires looking at both sides of the channel together.

9. The transducer problem

DEFINITION

A transducer is something that converts variety from one form to another — a microphone converts sound variety to electrical variety, a thermometer converts temperature variety to numerical variety, a translation rule converts variety in one standard's vocabulary to variety in another's. The transducer problem is the fact that transduction always loses or distorts some variety, and the choices made in designing the transducer determine which variety is preserved and which is not.

In GRC, transducers are everywhere and almost always unexamined. Every translation between standards is a transducer. Every classification that turns a free-text finding into a categorised one is a transducer. Every metric that turns a complex situation into a number is a transducer. Each one preserves some variety and discards some, and the discarded variety does not reappear downstream. The receiver cannot recover it; the receiver does not, typically, know it was discarded.

***Diagnostic question** — Where in this process is variety being transduced? What is being preserved by the transducer, what is being discarded, and is the discard appropriate for the downstream regulation?*

The transducer problem is the structural account of why translation rules between standards are not optional. When ISO 31000's "risk owner" is translated to NIST RMF's "Authorizing Official," a transducer is being applied. If the transducer is implicit, each translation is being done freshly and the discards are unpredictable. If the transducer is explicit and lives in the conjunction, the discards are at least consistent and can be inspected. Inspection is what allows the question "are we losing the right variety in this translation?" to be asked at all.

THE KIT, IN ONE SENTENCE

Variety is what a system has; requisite variety is how much the regulator needs; attenuation reduces it; amplification multiplies the regulator's reach; channels carry it between parts; channel capacity limits what gets through; the regulator-regulated relationship organises who is doing what to whom; variety mismatch is the shape of most failures; the transducer problem is where translation lives.

Using the kit

The terms are individually useful, but they become most useful when applied together to a concrete integration failure. What follows is a single example — a recurring control-effectiveness reporting problem — walked through three times. Each walk-through uses a different subset of the vocabulary, and each produces a different sharpening of the diagnosis. The point is not that any one diagnosis is correct and the others are wrong. Real integration problems usually have multiple structural causes, and the vocabulary lets a practitioner see them in turn.

The problem

An information-security programme operating under ISO/IEC 27001 reports quarterly to a risk committee on control effectiveness. The reports are produced by the security team from data collected across configuration management, change management, and event management systems. For two consecutive quarters, the report indicated all critical controls operating as designed. In the third quarter, a significant incident occurred whose root cause was a control that had been failing for approximately six months. The control's failure was visible in the source data throughout that period. It was not visible in the reports.

This is the kind of integration failure that produces, in most organisations, a search for who to blame and a tightening of reporting procedures. The variety-engineering vocabulary lets a practitioner ask instead what was structurally wrong.

First reading: attenuation and the transducer problem

The reports were summaries. Summaries are attenuations — they collapse the variety of the source data into a smaller variety the report consumer (the risk committee) can act on. The summary process is also a transducer: it converts variety in the operational domain (configuration states, change patterns, event sequences) into variety in the governance domain (control effectiveness ratings).

The transducer was discarding the variety that mattered. A control rated "effective" in the report carried the same signal whether it had been continuously effective for the quarter or had been failing for six months but happened to be passing its checks on the dates the data was sampled. The transducer's design — what it preserved and what it discarded — determined whether the failure was visible. In this case, the transducer was preserving the wrong variety, and the discard was operationally critical.

The intervention this reading suggests is at the transducer: redesign the attenuation so that control state over time, not just at sample dates, becomes part of what the report transmits. This is a substantive change to the substrate's measurement constructs, not a process change.

Second reading: channel capacity and saturation

There were three channels involved: from configuration management to the security team, from change management to the security team, and from event management to the security team. Each carried operational data that the security team transduced into governance signal. Each had capacity constraints.

Two of those channels were producing data at a rate far in excess of what the security team could process for governance reporting. The team's de facto attenuation was to sample — to look at what was happening on specific dates rather than continuously. Sampling is a low-capacity transduction strategy: it preserves point-in-time variety and discards everything between sample points. The channels were saturated for governance purposes, and the saturation was being managed by sampling without the saturation being acknowledged.

The intervention this reading suggests is at the channel: increase the effective capacity of the operational-to-governance channel, either by adding processing capacity, by moving the transduction closer to the source (so each operational system pre-attenuates for governance use), or by changing what the channel carries (transmitting exceptions rather than samples).

Third reading: the regulator–regulated relationship

Whose job was it to regulate the controls? In the standards' vocabulary, the answer is the security team operating the ISMS. But the controls were being maintained by the operational teams that ran configuration management, change management, and event management. The relationship was: operations did the work, the security team observed it through reports, the risk committee observed the security team's observations. Three layers of regulator–regulated relationship, with each layer's regulation depending on the variety that came up through the previous layer.

The relationship was not miscast in principle — the structure is reasonable. But the channel capacities between the layers did not match the variety that needed to flow. Each layer was regulating with significantly less variety than the layer it was regulating possessed, and the cumulative attenuation across three layers produced a regulator (the risk committee) operating with variety so much smaller than the regulated system (the operational estate) that requisite variety was almost certainly violated.

The intervention this reading suggests is structural: either reduce the number of regulatory layers, or substantially amplify each layer's variety, or build cross-cutting channels (an

algedonic path, in VSM terms) that bypass the layered structure when specific kinds of variety need to reach the top quickly. PDCA+'s conjunction is, structurally, what a permanent cross-cutting channel of this kind looks like.

What the three readings together produce

Each reading identifies a real structural issue. Each suggests a different intervention. None of them invalidates the others; the system has more than one structural problem, as integration failures usually do. The vocabulary lets the practitioner see them separately, which is what allows interventions to be chosen rather than improvised.

WHAT THE VOCABULARY BUYS YOU

The same problem looks like a transducer failure, a channel-capacity problem, and a regulator-regulated relationship problem — all three are true. The vocabulary lets you choose which to intervene on first, why, and what to expect from the intervention. Without the vocabulary, the choices are made by intuition, which is what most integration work currently is.

How to use this in practice

A vocabulary becomes useful through use, not through study. Three suggestions for getting the terms into active practice.

Pair them with what you already say

When you find yourself saying "the integration is broken" or "the hand-off is failing" or "we need better governance," stop and ask which of the nine terms more precisely names what you mean. Most practitioner sentences about integration problems have a more precise re-description in the vocabulary, and the re-description usually points to a more specific intervention. The point is not to replace the everyday language but to test it against the vocabulary and notice when the vocabulary sharpens it.

Use the diagnostic questions on cases you already know

Pick three integration failures from the last twelve months — ones where the diagnosis was clear in hindsight. Walk through each one using the diagnostic questions from three or four of the terms. The exercise is most useful on failures where you already know what was wrong, because it lets you check whether the vocabulary would have surfaced the diagnosis earlier than the actual investigation did. Often it would have, and seeing this directly is what makes the vocabulary feel earned rather than borrowed.

Apply it to a current problem in writing

Take a current integration problem and write a paragraph diagnosing it using at least three of the terms. The writing forces the vocabulary into use, and the paragraph is itself useful — it is something to share with colleagues, to test against alternative readings, to refine. A

practitioner who has written three diagnostic paragraphs has the vocabulary in active use. A practitioner who has only read the terms has them in passive memory, which is a different and much less useful place for them to live.

Limits of the vocabulary

The variety-engineering vocabulary is one tool, not a complete account of management-system integration. Its limits are worth naming.

It is most useful for diagnosing structural problems. It is less useful for diagnosing problems whose primary character is substantive — a control that is poorly designed for the risk it addresses is not, primarily, a variety problem. The vocabulary will not say much that is useful about why that control is wrong. It will say useful things about why the wrongness was not detected, which is often the relevant question, but it does not replace substantive expertise in the disciplines themselves.

It is also a vocabulary of analysis rather than intervention. Knowing that a channel is saturated does not tell you how to build the right replacement channel; knowing that a transducer is discarding the wrong variety does not tell you how to design the right transducer. The diagnostic precision the vocabulary offers is necessary but not sufficient for the engineering work that follows. The vocabulary tells you where to intervene; the intervention itself draws on different competencies.

Finally, the vocabulary's terms are interrelated, and the boundaries between them are sometimes more analytic than real. A channel-capacity problem and an attenuation problem are not always cleanly separable; a transducer problem and an amplification problem can shade into each other. This is fine in practice — the vocabulary is for diagnosis, not classification — but it means that two careful practitioners can describe the same failure in different combinations of terms and both be right. The point is the sharpening, not the labelling.

None of these limits undermines the vocabulary's usefulness. They are the conditions under which the vocabulary does its work, and being honest about them is what allows the vocabulary to be used well.

References

- Ashby, W. R. (1956). *An Introduction to Cybernetics*. London: Chapman & Hall.
- Ashby, W. R. (1958). "Requisite Variety and Its Implications for the Control of Complex Systems." *Cybernetica* 1(2), 83–99.
- Beer, S. (1972). *Brain of the Firm*. London: Allen Lane.
- Beer, S. (1979). *The Heart of Enterprise*. Chichester: John Wiley & Sons.
- Shannon, C. E. (1948). "A Mathematical Theory of Communication." *Bell System Technical Journal* 27(3), 379–423.
- Conant, R. C., & Ashby, W. R. (1970). "Every Good Regulator of a System Must Be a Model of That System." *International Journal of Systems Science* 1(2), 89–97.