FRAMEWORK · COMPANION · V1.0

Second-order observation

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

When a framework must remain adequate to a world that changes faster than the framework can be redesigned, it needs to observe its own observing apparatus. This piece names that pattern, specifies its conditions, and shows it in four domains.

AUTHOR

Joacim Brandell

Written for GRC practitioners · with worked instances from PDCA+, peer review, financial regulation, and production observability

Why this piece exists

The cybernetics destination to the PDCA+ whitepaper identifies the second-order observation as one of the structural properties that distinguishes the framework from siloed management-system integration. The companion treated the property briefly. This piece develops it.

The development is not just elaboration. Second-order observation is a pattern that appears in many places — wherever a framework is operating in a world that changes faster than the framework can be redesigned. PDCA+ is one instance; there are others, in domains a GRC practitioner will already partly know. Naming the pattern and showing how it appears in different domains is what lets the pattern be recognised, designed for deliberately, and used as a diagnostic question about any framework: does this one have it?

This piece is written for GRC practitioners and assumes no prior cybernetics background. It introduces second-order observation as a precise idea, distinguishes it from the looser readings that often appear in management literature, names four conditions a framework must satisfy to genuinely instantiate the pattern, walks through PDCA+ as the first worked instance, and then walks through three further instances from other domains. It closes with a diagnostic checklist a practitioner can use on any framework they encounter.

What second-order observation is

First-order observation observes the world. A thermometer observes temperature. A monitoring dashboard observes a system's operational state. A risk register observes the organisation's risk landscape. The observer and the observed are distinct: the thermometer is not itself a temperature, the dashboard is not itself an operational state, the risk register is not itself a risk.

Second-order observation observes the observing. The thermometer-calibration procedure observes whether the thermometer is reporting reliably; the monitoring system's self-test observes whether the dashboard is seeing what it should be seeing; an audit of the risk register observes whether the categories the register uses still distinguish the risks the organisation actually faces. The observer is now observing not the world directly, but the apparatus by which the world is being observed.

The distinction is older than cybernetics, but Heinz von Foerster articulated it most precisely in the 1970s, when he distinguished a first-order cybernetics that took the observer to be outside the system from a second-order cybernetics that included the observer as part of what is being observed. The shift is structural, not just philosophical. A framework that does only first-order observation can be very precise about what it sees in the world and entirely blind to whether its way of seeing is still adequate. A framework that does second-order observation can detect when its own ways of seeing are drifting relative to what they were designed to capture.

Why this matters structurally

All frameworks have to translate the world they operate in into representations they can act on. A risk register represents risks as entries; a control framework represents adequacy as compliance status; a configuration baseline represents desired state as a set of declarations. The representations are not the things themselves — they are reductions of the things into a form the framework can handle. This is necessary; without representation there is no framework. But the world keeps changing, and the representations were designed for the world the framework was designed in.

When the world changes slowly relative to how often the framework is redesigned, the gap between representation and represented stays small enough that first-order observation suffices. When the world changes faster, the gap opens. The framework continues to operate against its representations as if they still accurately captured the world. The framework's reports continue to look adequate. The world continues to diverge from what the reports describe. Eventually the divergence becomes visible — usually through a failure that the framework had no way to anticipate, because the framework was looking at the wrong thing all along.

Second-order observation is the structural answer to this problem. A framework that observes whether its own representations are still adequate can detect the drift before the drift becomes a failure. It does not eliminate the gap between representation and represented — that gap is necessary — but it makes the gap visible, and visible gaps can be closed by deliberate action rather than by crisis.

> THE CORE IDEA, IN ONE SENTENCE
>
> *First-order observation sees the world; second-order observation sees whether the apparatus that sees the world is still seeing it correctly. The first is necessary; the second is what allows a framework to remain adequate to a world that does not stay still.*

What it is not

Second-order observation gets confused with several other things, and the confusions are worth naming because each of them is close enough to mislead.

It is not reflexivity. Reflexivity in the loose sense — being thoughtful about one's assumptions, recognising that one is part of the situation one is observing — is a useful disposition but does not by itself constitute second-order observation. A reflexive practitioner can still operate within a framework that does no second-order observation; their reflexivity helps them work around the framework's limits but does not improve the framework. Second-order observation is a structural property of the framework, not a virtue of its operators.

It is not continuous improvement. Continuous improvement programmes typically observe the framework's outputs and improve the framework based on what they see. This is useful, but it is first-order observation applied to the framework's products rather than second-order observation of the framework's apparatus. A continuous-improvement

programme can be running effectively while the framework's representations are silently drifting; the programme will improve the products within the existing representations rather than question the representations themselves.

It is not audit. Audit observes whether the framework is being operated in conformance with its own specifications. This is necessary but again first-order: it observes the framework against its own representations, not the representations against the world. An audit can pass while the framework is diverging structurally from its purpose, because audit measures conformance to the framework, not the framework's adequacy.

It is not self-awareness. Anthropomorphising frameworks is a common rhetorical move and an unhelpful one. Second-order observation does not require the framework to have any property analogous to consciousness or intention. It requires specific structural features — which the next section names — that produce a particular kind of visibility about the framework's own operation. The features are mechanical, not psychological.

Each of these is useful in its own right, and most frameworks need some combination of reflexivity, continuous improvement, and audit to operate well. None of them substitutes for second-order observation, and treating any of them as if they did is one of the most common ways frameworks remain blind to their own drift.

Four conditions

A framework genuinely instantiates second-order observation when it satisfies four conditions. The conditions are minimal — each is necessary; together they are sufficient. A framework that meets all four has the pattern; a framework that fails any one of them does not, regardless of how close it appears to come.

1. **A first-order observation apparatus.** The framework must do regular observation of the world it operates in. Second-order observation cannot exist without first-order observation to observe. A framework without normal regulatory work — without representations, without monitoring, without reports — has nothing for second-order observation to be about. This condition is trivial to satisfy but worth stating because some discussions of second-order observation drift into pure self-reflection, which is something else.
2. **An explicit construct that represents how the first-order apparatus is supposed to work.** The framework must contain, somewhere accessible, a representation of what its own observing apparatus is supposed to be doing. Not just the apparatus itself, but a separate description of the apparatus's intended function. Without this, there is no benchmark against which the apparatus's actual operation can be compared. The construct can be informal — a documented theory of how the framework's representations capture the world — but it has to be retrievable and distinct from the apparatus itself.
3. **A surfacing mechanism that makes discrepancies visible.** When the first-order apparatus and its intended-function construct disagree, the disagreement must become

visible to someone who can act on it. This is the condition most frameworks fail. The apparatus may be drifting, the construct may still be on record, and yet nothing structurally surfaces the gap. The surfacing mechanism can take many forms — continuous instrumentation, periodic review, ad-hoc reporting from operators who notice — but if there is no mechanism, the second-order observation does not operate.

4. **A locus of revision.** There must be somewhere the apparatus or the construct can be revised in response to surfaced discrepancies. If discrepancies surface but nothing can be done about them, second-order observation collapses to a complaint channel. The locus of revision can be the framework's governance, the operator community, an external standards body — but it has to exist, have authority, and be reachable from the surfacing mechanism.

> **THE FOUR CONDITIONS, IN ONE SENTENCE**
> *Second-order observation requires first-order observation to observe, an explicit construct describing what the observing is supposed to do, a mechanism that surfaces gaps between intention and operation, and a locus where the construct or the apparatus can be revised.*

A useful diagnostic property of the four-condition spec is that the conditions can fail independently. A framework can satisfy three of the four and still not have the pattern. The failure mode for each condition is distinct, and naming the failure mode often points to a specific intervention.

Condition 1 fails when a framework tries to be entirely meta — pure self-reflection without engaging the world. Condition 2 fails when the framework's theory of itself is implicit, held only in the heads of its designers, and cannot be retrieved when needed. Condition 3 fails when discrepancies exist but the framework has no mechanism to make them visible — this is the most common failure. Condition 4 fails when discrepancies are visible but there is no authority to act on them, or when the authority exists but cannot reach the surfacing mechanism without distortion. Each failure is correctable, but the correction is different in each case.

Instance one: PDCA+ and the conjunction

PDCA+ is the framework these companion pieces have been developing, so it is the natural first worked instance. The pattern appears in the conjunction's continuous observation of its own substrate.

The conjunction holds schemas, taxonomies, translation rules, and measurement constructs as services to bound disciplines. These are the first-order apparatus. They observe the world the disciplines operate on — configuration states, risks, incidents, changes — by classifying, translating, and measuring what flows between disciplines. The first-order observation is what the whitepaper describes most fully.

The substrate also contains, as part of its structure, an account of what it is supposed to do. The schemas have specifications; the taxonomies have definitions of what each category is

meant to capture; the translation rules have stated scope conditions; the measurement constructs have stated definitions of adequacy. This is the explicit construct — the framework's theory of its own observing apparatus. A bound discipline can ask, of any conjunction service, what it is supposed to do and get an answer that is distinct from the service's actual operation.

The conjunction's continuous observation is the surfacing mechanism. When the taxonomy starts misclassifying findings — when the categories it provides no longer cleanly fit what is being asked to be classified — the conjunction sees this. It sees it because it holds both the actual flows and the construct that says what the flows are supposed to be, and the gap between them is observable as a structural property of holding both. The TLS-finding example in the whitepaper traces exactly this: a recurring misclassification surfaces because the conjunction has the substrate to make the misclassification visible.

The locus of revision is the conjunction discipline itself, in consultation with bound disciplines. The whitepaper describes substrate revisions happening at quarterly governance forums in the worked example, but the structural point is that the revision authority exists, is reachable from the surfacing mechanism, and operates without needing to wait for an external standards revision to do its work.

Condition	How this instance satisfies it
1. First-order apparatus	Schemas, taxonomies, translation rules, measurement constructs holding artefacts that flow between bound disciplines
2. Explicit construct	Specifications of what each substrate element is supposed to do — held distinctly from the elements themselves
3. Surfacing mechanism	Continuous observation of what passes through the substrate; gaps between specification and operation become visible as misclassifications, schema violations, translation failures
4. Locus of revision	The conjunction discipline, with input from bound disciplines, revising substrate in governance forums or as findings warrant

PDCA+ instantiates the pattern fully. Each condition is satisfied; the satisfaction is structural rather than incidental; the framework's design language already names the elements that satisfy them. This is what the cybernetics companion was identifying when it called PDCA+'s second-order observation a structural property rather than a virtue.

Instance two: scientific peer review and meta-research

Scientific peer review is the institutional mechanism by which a research community evaluates claims about the world before they enter the literature. It is, in cybernetic terms, a first-order observation apparatus: claims are observed against the community's standards of evidence and method, and the apparatus produces a judgement (accept, revise, reject).

For most of its history, the peer-review apparatus has had three of the four conditions of second-order observation in place but lacked the fourth — and in recent decades has been visibly constructing what was missing.

The first-order apparatus is the review process itself. The explicit construct exists in the form of disciplinary standards: what counts as good methodology, what evidence is sufficient, what claims are supportable. These standards are written down, taught in graduate programmes, refined through methodological literature.

The surfacing mechanism has historically been weak. Individual reviewers might notice methodological problems in individual papers, but the system as a whole had no mechanism to surface that the peer-review apparatus itself was producing unreliable judgements at scale. The replication crisis — most visibly in psychology and biomedicine in the 2010s — is what happened when the unreliability finally became too large to ignore. Meta-research, as a field, is the surfacing mechanism that was missing.

Meta-research observes whether the peer-review apparatus is producing claims that hold up under independent replication. Ioannidis's 2005 paper "Why Most Published Research Findings Are False" is one of the founding documents of the field. Replication projects, registered reports, and pre-registration norms are all instruments of the same surfacing mechanism: making visible whether the first-order apparatus is producing what it is supposed to produce.

The locus of revision is the disciplinary community and the journals collectively. Revisions to peer-review practice have been substantial in the last fifteen years — registered reports, mandatory data sharing, statistical-review specialists — and each is a response to discrepancies surfaced by meta-research. The locus existed all along but was rarely reached from the surfacing mechanism; what changed is the mechanism, not the locus.

Condition	How this instance satisfies it
1. First-order apparatus	Peer review: evaluation of claims against disciplinary standards
2. Explicit construct	Methodological standards, evidentiary norms, statistical-inference frameworks
3. Surfacing mechanism	Meta-research, replication projects, registered reports — built deliberately over the last two decades
4. Locus of revision	Journal editorial policy, professional society norms, training programmes; reachable, but only recently

The instance is instructive because it shows the pattern being constructed deliberately in a domain where it had been absent. For most of its history, peer review was a first-order apparatus with no working second-order observation. The replication crisis was, in cybernetic terms, exactly the kind of failure that first-order-only frameworks produce: the apparatus continued to operate normally while its outputs diverged from what they were supposed to capture, and the divergence became visible only through accumulated failure. The field's response has been to build the missing pattern, which is the work meta-research has been doing.

Instance three: financial regulation after 2008

Financial regulation provides a second worked instance, and one a GRC practitioner will likely know more intimately than the previous example. The 2008 financial crisis was, structurally, a failure of first-order-only regulation, and the post-2008 regulatory architecture is partly an attempt to add the missing second-order observation.

Pre-2008 financial regulation was largely microprudential — it observed individual financial institutions against capital, liquidity, and risk-management standards. Basel I and Basel II are the canonical articulations: a framework of standards by which the regulatory apparatus observed the world (the banks). This is a first-order apparatus, and within its scope it functioned reasonably well. Individual bank-level supervision improved across the decades the standards were in force.

What was missing was systematic observation of whether the regulatory apparatus itself was producing systemically adequate outcomes. The Basel II framework's reliance on banks' internal models, on credit-rating-agency judgements, on market-discipline assumptions — each of these was part of the apparatus, and each was drifting in ways that the apparatus had no built-in way to surface. The crisis was, in part, the visible accumulation of that drift.

Post-2008 architecture has added explicit second-order elements. The Financial Stability Board, established in 2009, has the mandate to observe whether the international financial regulatory apparatus is producing systemically adequate outcomes and to coordinate revisions. Macroprudential policy frameworks observe whether microprudential regulation is collectively producing the system-level outcomes it is supposed to produce. Stress-testing regimes are partly first-order (testing individual banks) and partly second-order (testing whether the regulatory framework's assumptions hold under stress conditions). These are not separate regulators on top of the existing apparatus; they are second-order observation built into the regulatory architecture.

Condition	How this instance satisfies it
1. First-order apparatus	Microprudential bank supervision under Basel standards; conduct regulation; market supervision
2. Explicit construct	Capital adequacy frameworks, liquidity standards, supervisory expectations — what regulation is supposed to ensure
3. Surfacing mechanism	Financial Stability Board systemic monitoring, macroprudential frameworks, stress-testing regimes
4. Locus of revision	International standard-setters (BCBS, IOSCO), national regulators with statutory revision authority

The instance is instructive for two reasons. First, it shows the pattern operating at very large scale across multiple jurisdictions and institutions — second-order observation does not require a single framework with a single governance. It can operate as an architecture of mechanisms across many frameworks, provided the four conditions are satisfied at the architectural level. Second, it shows how the failure mode of first-order-only regulation looks at scale: not as continuous gradual drift but as long stable operation punctuated by large-scale

failure. This is the failure mode every GRC practitioner should recognise, because it is the mode their own first-order-only programmes will eventually produce if the framework does not develop a working second-order observation.

Instance four: production observability and SLO frameworks

The third additional instance is contemporary and technical: how software-engineering organisations observe production systems. This will be partly familiar to GRC practitioners who work with operational teams; the structural shape is the same as the previous instances at a much shorter timescale.

First-order observation in production engineering is monitoring: dashboards, metrics, logs, traces that observe what the production system is doing. These are mature, well-tooled, and ubiquitous. Every contemporary engineering organisation has them.

The explicit construct is the service-level objective (SLO) — a written specification of what the system is supposed to do, expressed as measurable thresholds. "99.9% of requests served in under 100 milliseconds" is a typical SLO. The SLO is distinct from the monitoring: it is the theory of what the monitoring is supposed to be observing, written down somewhere retrievable.

The surfacing mechanism is the error budget. When the system's actual performance, observed through monitoring, differs from the SLO's specification, the difference accumulates in an error budget. The error budget makes the gap between intention and operation continuously visible. Importantly, the error budget is not just monitoring with a different name — it is the structural comparison between observed performance and the construct that says what performance should be. When the construct (the SLO) no longer matches what users actually need from the system, the error budget surfaces this too: an SLO consistently met but users still unhappy, or an SLO consistently missed without users complaining, both surface as discrepancies that prompt revision.

The locus of revision is the engineering team itself, sometimes with product input. SLOs are revised regularly — typically quarterly — based on what the error budget surfaces. The revision authority is local; the timescale is short; the revisions are routine rather than crisis-driven. This is what makes the SLO/error-budget pattern such a clean instance: all four conditions are designed in from the start, and the framework's normal operating cadence includes the revision loop.

Condition	How this instance satisfies it
1. First-order apparatus	Production monitoring: metrics, dashboards, logs, distributed traces
2. Explicit construct	Service-level objectives — written specifications of intended performance
3. Surfacing mechanism	Error budgets, SLO burn alerts; continuous comparison between observed and intended
4. Locus of revision	Engineering team's SLO review cycle; quarterly cadence; routine rather than exceptional

The instance is instructive because it shows the pattern operating at production-engineering tempo: short cycles, continuous surfacing, routine revision. The other instances operate at much slower timescales — peer review revises on the scale of years, financial regulation on the scale of decades. The four conditions are independent of timescale; what changes between instances is how the surfacing and revision cadences match the rate at which the world being observed is changing. The SRE/SLO pattern is what the conditions look like when the timescale is fast enough that the pattern is part of normal operations rather than a separate exercise.

What the four instances together show

Each of the four instances satisfies the same four conditions through different means. The differences are illuminating.

The conditions are realisable at any timescale

SRE/SLO operates at minute-to-hour timescales. PDCA+'s conjunction operates at days-to-quarters timescales. Peer review and meta-research operate at years timescales. Financial regulation operates at decades. The pattern does not require any particular cadence; it requires that whatever cadence is used matches the rate at which the world being observed is changing. A framework whose surfacing mechanism is slower than the world drifts will reliably miss the drift; a framework whose mechanism is faster than necessary will spend effort on noise. The right cadence is a design choice; the pattern's conditions do not specify it.

The locus of revision can be local or distributed

In SRE/SLO, revision is local to the engineering team and takes hours of meeting time per quarter. In financial regulation, revision is distributed across national regulators, international standard-setters, and legislative bodies and takes years. PDCA+ sits between: revision is local to the conjunction discipline but coordinated with bound disciplines. None of these is more correct than another; what matters is that the locus exists, has authority over the relevant construct, and is reachable from the surfacing mechanism. Distance between the surfacing and the locus is a design property worth examining in any framework — short distance means fast revision but limited authority; long distance means broader authority but slower response.

The surfacing mechanism is the hardest condition to satisfy

Most frameworks that fail to instantiate the pattern fail at condition three. The first-order apparatus is usually present; the construct usually exists somewhere (even if not formally); the locus of revision exists in some form (even if rarely used). What is missing is the mechanism that makes gaps between construct and operation continuously visible. Building the mechanism is the work that distinguishes a framework that performs first-order observation only from one that genuinely operates the second-order pattern. In each of the four instances, examining how the surfacing mechanism actually works is the most informative part of the analysis.

The construct is often implicit, and making it explicit is its own work

Condition two — an explicit construct of what the apparatus is supposed to do — is almost satisfied in most frameworks. The construct exists; it is in the heads of designers and senior practitioners; it can be reconstructed by anyone who knows the framework well. But it is not retrievable as a separate artefact that the surfacing mechanism can compare operation against. Making the construct explicit is one of the most underrated framework-design moves available, because it converts tacit knowledge into something a surfacing mechanism can use. The SLO pattern's success in software engineering is in part because SLOs make explicit what was previously implicit; the construct's externalisation is what enables the surfacing mechanism to do its work.

> **WHAT TO REMEMBER FROM FOUR INSTANCES**
>
> *The four conditions are timescale-independent. Locus of revision can be local or distributed. The surfacing mechanism is the hardest to build. The explicit construct is often nearly present but rarely retrievable — and making it retrievable is one of the highest-leverage framework-design moves available.*

A diagnostic checklist

The pattern's main practical value is as a diagnostic question: does this framework I am working with — designing, operating, evaluating — have it? The four conditions give the checklist.

- **Does the framework do first-order observation?** What apparatus does it use to observe the world it operates in? Is the apparatus mature enough that you can describe what it actually does, not just what it is supposed to do?
- **Is there an explicit construct of what the apparatus is supposed to do?** Can you point to a document, specification, or recoverable artefact that describes the apparatus's intended function, distinct from the apparatus itself? If the construct exists only in people's heads, condition 2 is not satisfied.
- **Is there a surfacing mechanism?** When the apparatus and the construct diverge, who notices, and through what mechanism? If the answer is "experienced operators notice

when something feels off," the mechanism is not structural and condition 3 is not satisfied.

- **Is there a locus of revision, and is it reachable from the surfacing mechanism?** When a divergence is surfaced, who can revise the construct or the apparatus? How long does that take? Is the path from surfacing to revision reliable, or does it depend on relationships and timing?

A framework that answers all four affirmatively has the pattern. A framework that fails any one of them does not, and the failure points to where to intervene. The intervention is different for each condition, and the difference matters for cost and effort.

Adding first-order apparatus where it is missing is the largest investment — building the observation infrastructure from scratch. Making an implicit construct explicit is the smallest — usually a documentation exercise. Building a surfacing mechanism is the most variable: it can be light (a quarterly review with the right scope) or heavy (continuous instrumentation across many systems). Establishing a locus of revision is mostly governance work and is bounded by how willing the framework's owners are to relocate revision authority closer to where surfacing happens.

Practitioners encountering a framework that does first-order observation only — which is most of them — can use the checklist to identify which of the three remaining conditions is most actionable, and to recommend the corresponding intervention. The diagnostic vocabulary makes the conversation precise where it would otherwise be impressionistic.

Limits of the pattern

The pattern is useful, and it generalises, but it does not solve every framework problem and pretending otherwise would weaken the cases where it genuinely helps. Three limits are worth naming.

First, second-order observation does not eliminate the need for occasional first-principles redesign. Even a framework that operates the pattern well will, over long enough time, accumulate drift the pattern cannot fully address — because the construct itself was based on assumptions that have shifted, and revising the construct requires going outside the framework's normal revision authority. The pattern stretches the interval between full redesigns; it does not eliminate it. Frameworks that treat the pattern as a substitute for occasional foundational rethinking will eventually be wrong-footed.

Second, the pattern can recurse — a framework can do second-order observation on its second-order observation, and so on — but recursion has costs and provides diminishing returns. Each additional level requires its own apparatus, construct, surfacing mechanism, and locus of revision. In most domains, two levels (first-order and second-order) is the practical maximum; third-order observation usually collapses back into one of the lower levels under normal operation. The pattern's value is mostly at the move from first-order-only to first-and-second-order; further recursion is theoretically interesting but rarely operationally useful.

Third, the pattern assumes that the world the framework observes is at least partially stable — that the construct's intended function remains a meaningful target across the timescale at which the framework operates. In domains where the operating environment is itself chaotically restructuring, the pattern provides less value because the construct cannot remain stable long enough to be a useful benchmark. This is rare in practice; most domains are stable enough that the pattern helps. But it is worth knowing that the pattern is not universally applicable, and that the question "is this domain stable enough for the pattern to operate?" is itself a useful design question.

Closing

The cybernetics companion ended by noting that PDCA+ instantiates several structural ideas from a long intellectual tradition without naming them, and that naming them makes the ideas more available for further development. This piece is one such development — taking second-order observation from a brief mention in the companion to a worked design pattern with four conditions, four instances, and a diagnostic checklist.

The pattern is not new. Von Foerster articulated it in the 1970s; Beer's audit channel (System 3*) is a closely related construction in the VSM tradition; meta-research has been instantiating it in scientific practice for two decades. What is potentially new is treating it explicitly as a design pattern available to anyone building a framework that needs to remain adequate to a changing world — which is most frameworks, including most GRC frameworks. The four conditions give the pattern a shape that can be checked, designed for, and recognised when present. That recognition is, in this author's view, the move that converts an abstract idea into a working diagnostic tool.

PDCA+ is one instance of the pattern. The framework's value comes in part from the pattern being designed into it from the start. Other frameworks can be evaluated against the same conditions, and the evaluation usually points to specific, actionable interventions. The pattern is portable. What this piece has done is make the portability legible.

References

- von Foerster, H. (2003). Understanding Understanding: Essays on Cybernetics and Cognition. New York: Springer.
- Beer, S. (1979). The Heart of Enterprise. Chichester: John Wiley & Sons.
- Ioannidis, J. P. A. (2005). "Why Most Published Research Findings Are False." PLoS Medicine 2(8), e124.
- Open Science Collaboration (2015). "Estimating the Reproducibility of Psychological Science." Science 349(6251), aac4716.
- Bank for International Settlements (2011). Basel III: A Global Regulatory Framework for More Resilient Banks and Banking Systems.
- Financial Stability Board (2009 onwards). Annual reports on the implementation and effects of regulatory reforms.
- Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (eds.) (2016). Site Reliability Engineering: How Google Runs Production Systems. O'Reilly.
- Maturana, H. R., & Varela, F. J. (1980). Autopoiesis and Cognition: The Realization of the Living. Dordrecht: D. Reidel.