

Proof of Growth v1.2

Proof of Growth — v1.2

Complete specification and current reading.

Concode · condaydigital.com v1.2 · 2026-08-29 · supersedes v1.1 of the same date

Changes from v1.1, all three forced by defects found in independent reproduction: gate disclosure must be scoped to the reporting window and is no longer permitted to mix scopes (§6), every row of a gate table must carry its source counts (§6, §7), and §6's cross-producer comparison is now declared as a bounded exemption from §2 rather than left implicit (§6.1). **v1.1's §6 figures were lifetime-scoped and are superseded** — the K-weighted mean, the producer's own acceptance ratio and the gap between them all change. K_G , $|W|$, R_G , the projection and the target ratio are untouched.

Changes from v1.0, all forced by the adversarial review shipped alongside this document: scope narrowed to within-subject (§2), gate disclosure now required (§6), Goodhart added as the primary failure mode (§8), normative core stripped of product names (§4), invalid uses declared (§0.2), validation protocol stated (§9).

PART I — THE SPECIFICATION

0 · What this is

A credential certifies a **level**: verified once, by one institution, on a date now past.

Proof of Growth evidences a **trajectory**: verified continuously, by many parties, over a window the reader picks.

These are different claims. Neither replaces the other. v1.0 framed PoG as a credential substitute; that framing is withdrawn.

PoG is not a score. It is a set of numbers reported together with the evidence they were computed from. Any presentation collapsing it to one number is not PoG.

0.1 · Valid uses

- Self-assessment against your own earlier baseline.
- Voluntary disclosure of a trajectory.
- Answering: *is this producer's verified output rate rising, flat or falling, at gates whose height is disclosed?*

0.2 · Declared invalid uses

These are not cautions. They are declared out of scope, and a figure produced for them is not a PoG figure.

1. **Ranking or comparing different people.** The confounds are between-subject and do not cancel. See §2.1.
2. **Any decision where the producer has an incentive to inflate A** — hiring, funding, grant award, litigation — unless A is omitted entirely and R is reported with gate disclosure.
3. **As a measure of quality.** PoG counts acceptances. It does not inspect artifacts.
4. **As a target.** See §8.1.

A measure that declares its invalid uses cannot be discredited by them. Lines of Code, the h-index and the Journal Impact Factor each failed to do this, and each was discredited by a use its authors had not disclaimed.

1 · The unit

An artifact is a **verified unit** if and only if it carries all three:

field	requirement
locator	An identifier any third party can resolve without your cooperation.
timestamp	A date recorded by a system you do not control.
verifier	A named party, other than you, who accepted it and had standing to reject it.

All three. The verifier condition does the work.

Self-publication fails. A page on your own domain has a locator and a timestamp. Nobody with the power to say no said yes. It is output. It is not verified output. This is not a judgment about worth — it is a statement about who attested.

These three fields are deliberately free of any technology. They would have worked for a guild register in 1400 and a parish record in 1700.

2 · Scope: within-subject, longitudinal

A PoG figure compares one producer against their own earlier baseline. Nothing else.

2.1 · Why

Three confounds sit between any two producers and none of them cancel:

confound	effect
access	Verifier availability differs by field, geography, language, employment. R partly measures who will look at your work.
gate height	Two acceptances at differently-selective gates are not one unit each.
unit size	A one-line correction and a six-month body of work are both 1.

Two producers at the same R have done incomparable things.

Within one producer across adjacent windows, all three are approximately constant — same field, same access, same gate mix, same unit-size distribution. **That is the entire justification for the measure, and it is why it is called Growth and not Level.**

3 · The window

$W = [t1, t2]$. **|W| is elapsed calendar days. Not active days, not working days.**

An active-day denominator requires knowing which days you worked — and across more than one machine, account or context, the **union** of those calendars. That union is reported by no platform and cannot be derived by adding them, because the calendars overlap. The rate becomes uncomputable.

Elapsed days needs two dates and nothing else. It is also the more honest denominator: a rate divided by active days measures intensity while working; divided by elapsed days it measures what arrived in the world between two dates. Rest is part of the rate.

4 · Verifier classes

Classes are defined by **properties of the verification**, never by the platform providing it.

class	property
G — gated	A human or committee exercised discretion, and had an interest in rejecting.
A — automated	Deterministic pass/fail. No discretion, no rejection interest.
P — publication	A registry issued a permanent, publicly resolvable identifier.

Classes are reported side by side and never summed. Adding a gated acceptance to an automated completion produces a number whose meaning depends entirely on the mix.

Weights are forbidden. A weight is a hidden argument a reader can only accept or reject, and one contested weight discredits the whole figure. Stratify instead — it puts the argument on the page.

5 • The two terms

- $K(w)$ — verified units per \$1, timestamped inside w
- $A(w)$ — **all** artifacts produced in w : shipped, half-finished, abandoned, unverified

$R(w) = K(w) / w $	verified units per elapsed day
$V(w) = K(w) / A(w) $	verified share, 0..1

R alone is a cherry-pick: it shows what got through and is silent on what did not. V alone is a snapshot with no time in it.

A is self-reported and unauditable. Nobody else can count what you started and abandoned. This is why \$0.2 excludes adversarial use: a self-reported term with an incentive to misreport will be misreported, and PoG would degrade to R alone — which this section declares invalid on its own.

A low V with a high A is the signature of someone attempting more than they finish. That is a different thing from producing little, and making it visible is part of why the second term exists.

6 • Gate disclosure — required

When reporting R_G , you must report the height of the gates that produced it.

All figures in this section are scoped to $[t1, t2]$, the same window as R . A gate table weighted by lifetime acceptances does not describe the gates that produced K_G ; it describes a different, larger set whose repository mix is not the same. Mixing the two scopes in one statement is a defect, not a presentation choice.

Let $M(v)$ = the producer's units accepted by verifier v with acceptance timestamp inside $[t1, t2]$. Sum of $M(v)$ = K_G , necessarily.

For each verifier v with $M(v) > 0$:

$a(v)$ = accepted submissions by ALL producers / total submissions by ALL producers
Report the M -weighted mean of $a(v)$, and the range.

This is a disclosure, not a weight — every term is a count, and nothing is scaled by it.

Every row must carry its own source counts. A row printing a percentage with its numerator and denominator withheld is a figure without a locator, and §7 excludes it from the statement. A table whose unsourced rows carry weight in the headline mean is worse than one that omits them, because the defect is invisible in the summary figure. Either source every row, or drop the unsourced rows, recompute the mean over what remains, and state how many units and what share of K were excluded.

Second required disclosure: the producer's own acceptance ratio, also scoped to the window — their submissions *created* inside $[t1, t2]$ that were accepted, over all their submissions created inside $[t1, t2]$. A producer whose personal rate sits far above the gates' general rate is selecting easy work; far below, they are attempting above the median. R shows neither and hides that there is anything to see.

Censoring must be stated. A short window right-censors the producer's own ratio: submissions made near $t2$ have not had time to be decided. The gates' general rate is computed over a pool that is almost entirely decided. The two are therefore not cleanly comparable over a short window, and any statement reporting them together must say so and give the undecided count. **Report the raw ratio as primary** — it shares its construction with the gate rate, undecided items in the denominator of both. A decided-only ratio may be given alongside; it may not replace the raw figure, and choosing between them by which is more flattering is a violation of §7.

Without §6, R_G is Lines of Code with a better vocabulary.

6.1 · §6 is a bounded exemption from §2 — declared

§2 scopes PoG to within-subject comparison and lists three confounds that do not cancel between producers. §6 then compares one producer's acceptance ratio against **all** producers at the same gates. That is a between-subject comparison, and §2 forbids exactly this. The tension is real and is resolved here explicitly, because a reviewer who finds it undeclared will use §2 to discredit §6.

The exemption. A gate's general acceptance rate is reported as **context for a single producer's own figure**, never as a ranking of producers. The comparison is *producer-against-gate*, not *producer-against-producer*. The gate rate is a property of the verifier, not of any person, and it is the only way to distinguish 19 acceptances at gates that admit four in five from 19 at gates that admit one in two — a distinction R_G cannot make and without which §8.1 has no mitigation at all.

The bound. No cross-producer ordering may be derived from §6, and doing so is an invalid use under §0.2.1. Specifically:

- The gap between a producer's ratio and a gate's general rate **is not a quality signal, in either direction.**
- **§2.1's unit-size confound bites here directly.** A ratio below the gate's general rate is exactly what you would observe from a producer submitting larger changes than the median contributor — and also exactly what you would observe from one

submitting work that needs more revision. **§6 cannot distinguish these and does not attempt to.** §9.3.2 is the study that would.

- Two producers' gaps may not be compared with each other. The gap is interpretable only against the same producer's gap in an adjacent window, which is §2's rule reasserted.

The exemption is granted to the gate figure alone. It does not extend to R, V, or any other quantity in this document.

7 • Reporting format

A PoG statement is complete only with all of:

window	t1 -> t2, and W in elapsed days
per class	K , R, and the verifier(s) named
gate	weighted mean acceptance and range, per §6
own ratio	the producer's acceptance ratio at those gates
A	self-declared total with its counting rule – or explicitly omitted
V	per class, marked self-declared
baseline	the same figures at t1
locators	queries anyone can run to reproduce K
losses	known unrecoverable records, per §8.6
prediction	a dated future figure, per §9.2

Any figure without a reproducible locator is not part of the statement. **Do not round.**

8 • Failure modes

8.1 • Goodhart — the primary one

When a measure becomes a target, it ceases to be a good measure.

This is not theoretical for R_G. Hacktoberfest incentivised merged pull requests and produced a flood of whitespace edits and one-line changes aimed at projects that had not asked for them; maintainers were swamped and the programme was restructured to opt-in. That is R_G optimisation run as a natural experiment, and it failed.

Class stratification alone does not defeat it, because gated verifiers are not uniform — farming low-gate verifiers raises R_G as easily as farming badges raises R_A. §6 is the mitigation: gate-farming raises the disclosed acceptance rate, visible in the same table as the rate it inflates.

§0.2.4 states the rest of the mitigation: PoG is not to be used as a target.

8.2 · Access asymmetry

R partly measures access to verifiers. Not corrected for. §2 confines the damage.

8.3 · Unit heterogeneity

Stratification does not fix it. Report unit counts and never claim they are size-adjusted.

8.4 · A is unauditale

§5. Marked, not solved; §0.2.2 is the containment.

8.5 · Window shopping

A flattering window can be chosen. Report lifetime-to-date from a stated origin alongside.

8.6 · Survivorship

Lost or deleted work vanishes from A and inflates V. Record known losses explicitly.

9 · Validation

9.1 · Status

PoG is an unvalidated instrument. It has no test-retest data, no convergent validity against any independent measure, and no cohort test of its predictive claim. Until §9.3 runs, no decision should rest on a PoG figure.

The predictive claim, stated so it can be attacked:

Verified output rate over a recent window predicts future verified output rate better than credential status does.

Falsified by a cohort study in which credential status predicts subsequent verified output at or above the predictive power of a prior-window R, controlling for domain and access.

9.2 · Every statement carries a dated prediction

Project R forward, name the date, let the record settle it. A measure that never commits to a future number cannot be wrong, and a measure that cannot be wrong is not evidence.

9.3 · The protocol that would need to run

1. **Test-retest** — stability of R across adjacent equal windows, same producer.
2. **Convergent validity** — R against an independent competence measure in the same domain.

- 3. **Predictive** — prior-window R vs credential status as predictors of next-window verified output, in a cohort, controlling for domain and access.
- 4. **Gaming** — does R rise without a corresponding rise in independently-judged quality when producers know they are measured?

PART II — CURRENT READING

Producer: **Concode**. Origin **2026-04-22** (GitHub account created 2026-04-20 — the origin is itself externally attested).

Window: t1 2026-07-13T18:27:05Z (prior published pog.json) -> t2 2026-08-29. |w| = **46.2312 elapsed days**. Lifetime-to-date from origin: **129 days**.

Class G — gated

	baseline t1	reading t2	delta
merged pull requests	27	46	+19
distinct repositories (lifetime)	—	17	—
submissions opened, all states (lifetime)	—	112	—

$K_G = 19$ $R_G = 19 / 46.2312 = 0.4110$ verified units/day

Gate disclosure — §6, window-scoped

Primary table. Scoped to [t1, t2], per §6. The 19 units of K_G, across the 10 verifiers that produced them. Every row carries its source counts. Read 2026-08-29.

repository	window merges	accepted (all)	submitted (all)	gate
joke2k/faker	7	1,290	1,566	82.3755%
mne-tools/mne-python	3	7,727	9,012	85.7412%
nilearn/nilearn	2	3,455	3,943	87.6236%
15five/django-scim2	1	149	184	80.9783%
aeon-toolkit/aeon	1	2,032	2,551	79.6550%
jaegertracing/jaeger	1	4,497	6,252	71.9290%
mailerlite/mailerlite-nodejs	1	55	81	67.9012%
mdomke/schwifty	1	144	187	77.0053%
octokit/request.js	1	594	661	89.8638%
zeroclaw-labs/zeroclaw	1	4,278	6,846	62.4890%

Column 2 sums to 19 = K_G, as §6 requires.

M-weighted mean gate acceptance : 80.995846% -> 81.00%
range : 62.4890% (zeroclaw) – 89.8638% (octokit/request.js)

Producer's own acceptance ratio, window-scoped. Submissions created inside [t1, t2]:

disposition	count
accepted	3
closed, not accepted	4
still undecided at t2	7
total created in window	14

raw (primary, per §6) : 3 / 14 = 21.428571% -> 21.43%
gap vs 80.995846% : 59.567275 points -> 59.57
decided-only (secondary): 3 / 7 = 42.857143% -> 42.86%
gap vs 80.995846% : 38.138703 points -> 38.14

Censoring, stated per §6. Half the window's submissions — 7 of 14 — were undecided at t2. The gates' general rate is computed over pools that are almost entirely decided. The two constructions are therefore not cleanly comparable over a 46-day window, and neither ratio above is a settled figure. The raw ratio is primary because it shares its construction with the gate rate: undecided items sit in the denominator of both. The decided-only ratio is reported because withholding it would also be a choice. **Neither is selected for being the more favourable, and the raw figure is the worse of the two.**

Reading. These are moderately open gates — they accept roughly four submissions in five, which is neither trivial nor selective. The producer's window ratio sits far below the gates' general rate on either construction. That gap is the most informative quantity on this page and it is invisible in R_G. Per §6.1 it is not a quality signal in either direction: it is consistent with attempting larger changes than the median contributor, and equally consistent with submitting work that needs more revision. **PoG cannot distinguish those.** §9.3.2 is the study that would.

A second fact the window scoping exposes: only 3 of the 19 units in K_G were *submitted* inside the window. Sixteen were submitted earlier and accepted during it. Acceptance lags submission, so R_G over a short window is substantially a measure of work done before the window opened. Any single-window R inherits this lag, and comparing adjacent windows — §2's prescribed use — is the only reading that partly cancels it.

Lifetime figures — labelled, not primary

Reported so the change from v1.1 is visible. **These are lifetime-scoped and are not the §6 figures.** All 17 repositories, all 46 merges, every row sourced.

repository	lifetime merges	accepted (all)	submitted (all)	gate
joke2k/faker	8	1,290	1,566	82.3755%
mne-tools/mne-python	6	7,727	9,012	85.7412%
aeon-toolkit/aeon	4	2,032	2,551	79.6550%
mdomke/schwifty	4	144	187	77.0053%
deepmodeling/dpdata	3	654	744	87.9032%
15five/django-scim2	2	149	184	80.9783%
Adyen/adyen-node-api-library	2	1,337	1,624	82.3276%
jaegertracing/jaeger	2	4,497	6,252	71.9290%
js-joda/js-joda	2	379	539	70.3154%
krkn-chaos/krkn-ai	2	163	304	53.6184%
nilearn/nilearn	2	3,455	3,943	87.6236%
nipy/nibabel	2	799	970	82.3711%
octokit/request.js	2	594	661	89.8638%
zeroclaw-labs/zeroclaw	2	4,278	6,846	62.4890%
duckdb/duckdb-web	1	5,410	5,610	96.4349%
growthbook/growthbook	1	4,225	5,234	80.7222%
mailerlite/mailerlite-nodejs	1	55	81	67.9012%

K-weighted mean (lifetime) : 79.823770% -> 79.82%
 range : 53.6184% (krkn-ai) – 96.4349% (duckdb-web)
 own acceptance ratio (lifetime) : 46 / 112 = 41.071429% -> 41.07%
 gap : 38.752341 points -> 38.75

These were v1.1's headline \$6 figures. They are superseded as the \$6 statement and retained only as lifetime context.

Class A — automated

	reading t2
rooms completed	579
global rank	1103 (top 1%)
badges	33

R_A is not computed. No baseline exists at t1. A stock is reported; a rate is not claimed.

Class P — publication

$K_P = 0$.

artifact	why it fails §1
<i>Eternal</i> (live, condaydigital.com)	self-published — verifier is the producer
<i>Terrarium</i> (HTTP 200, 1,476 LOC)	self-hosted — no verifier
<i>Return Address — Not Mine</i> (34,278 words)	complete; published without an ISBN, by decision
<i>The Structural Tail</i>	written, held
<i>No Ground Truth</i>	written, not yet deposited
<i>Resilienz und Ausdauer</i>	published 2025 under a pen name; outside window, locator does not resolve to this producer

K_P is expected to remain 0, and that is a decision rather than an omission. The producer has elected to self-host rather than deposit with third-party registries. §1's verifier condition is unambiguous about the consequence: self-publication supplies a locator and a timestamp but no party with standing to reject, so it produces output and not verified output. The class is therefore reported as structurally zero for as long as that policy holds.

This is stated rather than quietly dropped because a reader is entitled to know that a class reading zero is a choice about distribution and not a shortfall in production — the artifacts listed above exist and are readable. It also means the instrument is, for this producer, carried entirely by class G. Any future non-zero K_P would represent a reversal of the policy, and should be reported as such.

v — not reported

The catalogue totals 315 artifacts (71 SHIPPED · 167 HALF · 69 IDEA-ONLY · 8 UNSHIPPED) but is cumulative from ~2026-05-05, not scoped to [t1, t2]. Computing v requires date-filtering transcripts.tsv to the window. One pass.

SHIPPED is not K. The two are orthogonal axes:

	verified (§1)	not verified
shipped	merged PRs	<i>Eternal</i> , <i>Terrarium</i> , /app/
not shipped	(empty)	167 HALF, 69 IDEA-ONLY

Everything in the top-right cell is real output no third party attested. **The verified share is smaller than the shipped share.**

Known losses — §8.6

1,164 sessions with no surviving record; the assistant side of 450 further sessions; 53 catalogued paths whose files no longer exist. Excluded from any future A; their exclusion

will inflate any V computed later.

Prediction — §9.2

At $R_G = 0.4110/\text{day}$ sustained over 153 days to **2027-01-29**:

```
projected merged PRs = 46 + (0.4110 x 153) = 108.9 -> 109
```

Producer's stated target: **100 merges by 2026-10-01** — 54 in 33 days, **1.6364/day, 3.98x the measured rate.**

Both are on the record with dates. Either outcome is informative.

PART III — RE-RUN PROTOCOL

For the next reading. Everything below is mechanical.

What moves which number

action	moves
Submitting to other people's gates and getting accepted	K_G, R_G
Submitting more and being rejected more	lowers own acceptance ratio; R_G unchanged
Submitting to more selective projects	lowers disclosed gate %, own ratio may fall
DOI, ISBN or registry identifier issued	K_P — currently zero
Publishing to your own site or repos	nothing. Fails §1 verifier condition
Date-filtering transcripts.tsv to the window	unlocks A, therefore V
Recording a t_1 baseline for class A	unlocks R_A

The producer's window ratio of **21.428571%** (raw, 3/14) is a ratio of accepted to submitted at external gates, scoped to $[t_1, t_2]$. It rises only when a higher share of external submissions is accepted. Volume of self-published output does not enter it. The lifetime figure of 41.071429% (46/112) is context only and is not the §6 statement.

Commands

```
python3 pog.py config.conday.json > pog.json    # R, V, classes, prediction
python3 gates.py 20                             # gate disclosure, resumable
```

`gates.py` is resumable by design — unauthenticated search is rate-limited to roughly 10 requests per minute, and 17 repositories need 34 calls. Delete `gates.json` before a fresh run.

Setting github_token in the config raises the limit.

Three things to fix before the next reading

1. **Set a class-A baseline now** so the next window can report R_A instead of a bare stock.
 2. **Date-filter the catalogue** to $[t_1, t_2]$ so V becomes computable.
 3. **Move K_P off zero.** One DOI does it.
-

Reproduce

```
window merges by repo    is:pr author:CedricConday is:merged merged:2026-07-
13T18:27:05Z..2026-08-29T00:00:00Z
submissions in window    is:pr author:CedricConday created:2026-07-13T18:27:05Z..2026-08-
29T00:00:00Z
  accepted                + is:merged
  undecided               + is:open
  closed unaccepted       + is:closed is:unmerged
per-gate rates            repo:<r> is:pr   and   repo:<r> is:pr is:merged
```

Reproduce every figure before citing it. The adversarial review ships with this document, not after it — §0.2 and §9.1 are the reasons.