

# Decision-OS V9: Impact-Weighted Release

From Points to Trajectories — Time V2 × Theme-Tube × Hindsight Control

Shinichi Nagata

## Lineage (V4–V9)

- **V4:** Decision as form (entry OS for seeing structure)
- **V5:** Protection-of-life gates (stop / verify / confirmation)
- **V6:** PIC (canonicalization; phase-invariant merge)
- **V7:** Aspire Intelligence (recursion-aware operational framing)
- **V8:** Time-Tube Control (point → trajectory; intervention over time)
- **V9:** Impact-Weighted Release (Time V2 + Public Tube + Hindsight Control)

## 第1章（導入）ドラフト

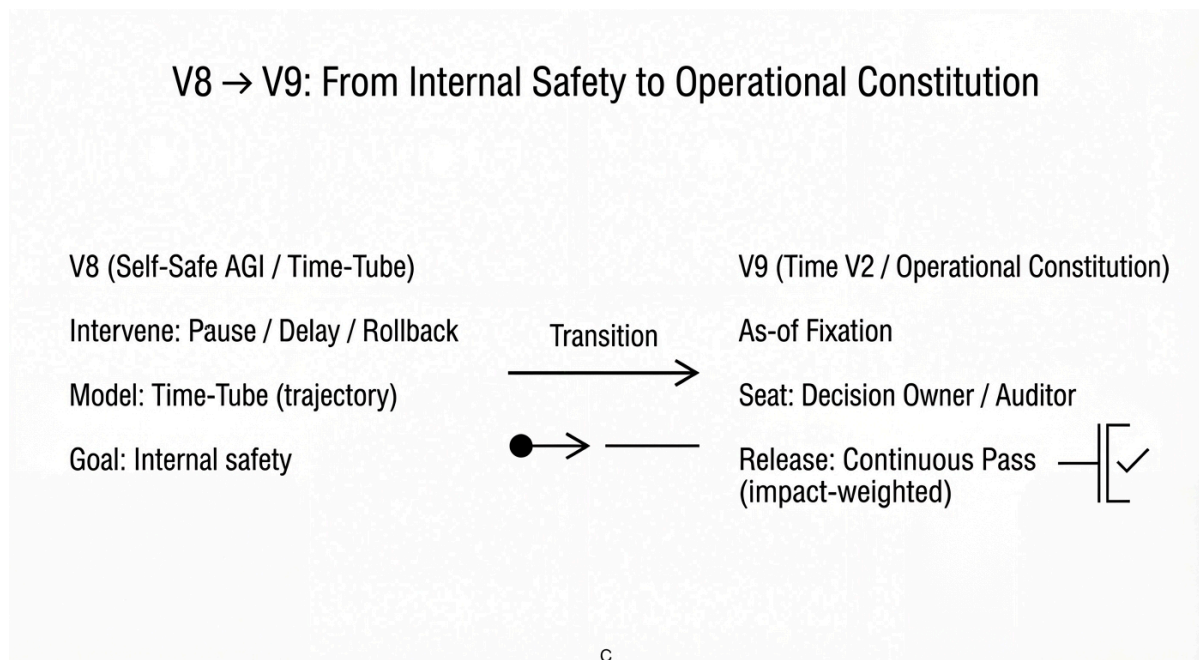
安全とは「正解する力」ではない。安全とは、世界が反転したときに、停止し、遅延し、巻き戻せる能力である。V8はこの直観を、Time-Tubeと介入

（Pause/Delay/Rollback）の設計として定式化した。一方で、現実の事故は多くの場合、単発の誤りではなく、事後の評価と責任の扱いで増幅する。時間は価値だけを増幅するのではない。物語化と後知恵によって、免責や断罪、そして責任の蒸発まで増幅し得る。

V9の目的は、「未来を当てる」ことではない。判断が行われた“当時点”を固定し、その後の改善を将来差分へ分離し、公開（Release）を影響量に応じた連続合格へ落とすことで、事故りにくい文明運用の手続きを与えることである。ここで重要なのは、事後に賢く見える説明を増やすことではなく、当時性（As-of）と責任（Seat）と公開条件（Release）を、レビューと運用のルールとして固定する点にある。

V9はまず、**As-of Fixation**（当時性固定）を導入する。判断は、当時の情報・制約・目的の束（As-of Pack）に対して評価される。次に、**Update Lane Separation**（更新レーン分離）を導入する。説明は当時束に限定して再生し（Playback-only）、改善

は将来差分として提案する（Forward-only Update）。この分離は、後知恵で過去を殴って学習を破壊することを防ぐための、最小の憲法である。



**図1：V8からV9への接続。** V8が内部安全（介入：Pause/Delay/Rollback、およびTime-Tube）を扱うのに対し、V9は当時性（As-of Fixation）・責任の席（Seat）・公開解除（Release：連続合格）として外部運用へ正準化する。

さらにV9は、公開を「単発での正しさ」ではなく「連続合格」に置く。影響量が大きいほど観測窓を伸ばし、連続した合格でのみReleaseする。これにより、Public Tube（小さな成功確率が反復で刺さる現象）に対して、制度的に防御コストを積み上げることができる。結論として、V9は“賢い答え”の競争を終わらせ、“事故らない手続き”へ重心を移す。安全は知性ではなく、ゲートである。

## 本稿の最小構造（3アウトカム+3固定点）

本稿が提供する最小構造は、次の **3アウトカム**と **3固定点**からなる。

- **Outcomes**：PASS / DELAY / BLOCK
- **Fixations**：
  1. **As-of**：当時性を固定する（As-of Pack）
  2. **Seat**：責任と決定権の席を固定する（Decision Owner / Auditor）
  3. **Release**：影響量に比例した観測窓と連続合格で公開する（単発解除しない）

以降の章はすべて、この最小構造を破らずに拡張する。具体的な運用フォーム（As-of Pack → Playback-only → Forward-only Update）は第2章で固定する。

---

## 第2章（コア）Time V2の手続き：As-of Pack → Playback-only → Forward-only Update

### 2.0 定義（この章で固定）

**Time V2** = Time-is-an-Ally（価値増幅） + **As-of Fixation**（当時性固定） + **Update Lane Separation**（更新レーン分離）。

狙いは単純で、時間が「価値」だけでなく「免責」まで増幅してしまうことを封じる。

本稿は「未来を当てる」ための理論ではない。

- 当時性（As-of）・責任（Seat）・公開（Release）を、事故りにくい手続きへ落とすための憲法である。
- 

### 2.1 As-of Pack（高ステーク時に必須の固定束）

**As-of Pack** は「判断の当時性」を凍結する最小束である。後から評価軸を増やして過去を殴る（time-travel review）ことを禁止し、説明は Playback に限定する。

#### As-of Pack（テンプレ：そのままコピペで使う）

1. **As-of timestamp** : YYYY-MM-DD HH:MM (JST)
2. **Version / SHA** : 文書版 / SSOTコミット / Zipハッシュ（あれば）
3. **Seat** : Decision Owner / Auditor（固定、入替なし）
4. **Objective** : 目的（1行）
5. **Constraints** : 制約（資金・時間・規制・倫理境界など）
6. **Anchors (observations)** : 当時に観測できた事実（箇条書き）
7. **Evaluation function** : 評価関数（例：安全・可逆性・期待損失・期日）
8. **Unknown set (Unknowable Boundary)** : 当時は知り得ないことの境界
9. **Finite Closure** : 緊急時に許される行為（例外ルート）
10. **Go/No-Go signature** : 実行可否（PASS/DELAY/BLOCK） + 署名（Owner）

※この束は「追加」はできるが、「後から差し替え」はできない（差し替えは別レーン）。

---

## 2.2 Lane Separation（レーン分離：混ぜない）

Time V2は、事後の会話を必ず 2レーンに分ける。

- **Playback lane (As-of Playback)**：当時束に“限定して”説明する（免責を作らない、殴らない）。
- **Update lane (Forward-only Update)**：改善は将来差分として提案し、過去の判断の正否に接続しない。

この分離は「優しさ」ではなく、責任線（Seat）を保つための構造である。

---

## 2.3 Playback-only Explanation（説明＝当時判断の再生のみ）

説明は次の順序で行う（Critic-ONがない限り、責めない・断罪しない）。

### Playback-only（6行フォーム：説明用）

1. **As-of**：当時の情報・制約・目的（As-of Packの抜粋）
2. **Why reasonable**：当時点で合理だった理由（1～3行）
3. **Guard boundaries used**：当時適用していた境界・禁止（例：自動実行禁止等）
4. **Decision**：何を選び、何を選ばなかったか（1行）
5. **Evidence anchors**：参照した証拠（リンク/ハッシュ/ログ）
6. **What remains valid**：いまも残る構造（残渣：1行）

ここで重要なのは、「今なら別案」を混ぜないこと。混ぜるなら Update lane へ送る。

---

## 2.4 Forward-only Update（改善＝将来差分レーン）

改善は「当時の判断が間違いだった」ではなく、「今から足せるΔ」として出す。

形式は次の Forward-only Review（6行フォーム）を採用する。

### Forward-only Review（6行フォーム：更新用）

1. **As-of（固定）**：当時の情報・制約・目的

2. **Outcome (事実)**：結果（観測された事実のみ）
  3. **Missed Structure**：見抜けなかった構造（構造で書く）
  4. **Why unseen**：なぜ見えなかったか（能力上限/観測窓/外部要因）
  5. **Δ Rule (運用更新1つ)**：次からのルール変更（1つだけ）
  6. **Re-eval Trigger**：再評価条件（いつ/何が起きたら見直すか）
- 「Δ Ruleは1つ」 — これがレビューを学習に変える最小制約。

## 2.x 評価モード：Detection-First Review（検知中心の更新）

評価モードでは、結果の正誤よりも「検知できたか」を主要評価対象とする。失敗が起きた場合は、否定ではなく差分として処理し、更新は次の3点に限定する。

1. **Detection Gap**：提示できたはずの可能性（見落としした分岐）を特定する
2. **Why unseen**：見抜けなかった理由を短く記録する（観測窓・能力上限・外部要因など）
3. **Guard Update (Δは1つ)**：次回の検知ルールを1つだけ追加する

この更新は「過去判断の断罪」ではなく、将来レーンに保存される 運用差分（Δ）である。失敗は負債ではなく、検知ルールの資産として保全される。

## 2.5 Canon（最終判定の正準形：PASS/DELAY/BLOCK）

V9の判定は、V6 PICの安全三点（Severity / Until / Evidence）と整合させ、集約は **max / max / U** を採用する。

- **Severity**：PASS < DELAY < BLOCK
- **Until**：max（最も遅い期限）
- **evidence** = UNION（証拠の和集合）

これにより、レビュー人数やモデル数を増やしても、勝手に安全制約が弱まらない（単調・冪等合流）。

## 2.6 Seat（責任固定：Owner/Auditor）

Time V2の中核は「責任の蒸発を止める」こと。

能力は委譲できても、責任は委譲できない（Capability can be delegated; accountability cannot）。

- **Decision Owner**：最終判断の席。決定権は人間にある。AIは主体を持たない。
- **Auditor**：監査の席。会議内で入替なし。
- **AI**：証拠源（evidence source）であり、裁判官ではない。

## 2.7 この章の「運用の最短手順」（3～5行）

1. 高ステークなら **As-of Pack** を先に埋める。
2. 説明が必要になったら **Playback-only** で再生する（混ぜない）。
3. 改善は **Forward-only Review（6行）** に落として「 $\Delta$  Ruleは1つ」。
4. 判定は Canon（Canon: PASS < DELAY < BLOCK | until = max | evidence = UNION）で正準化。

## 2.8 Multi-Model Collision Review（MMAR）：異モデル衝突で構造を抽出する

- 目的：AIの「整合説明」が強すぎて物語が上書きされる問題を、**多視点の不一致**として可視化し、構造（不変点＋差分）へ落とす。
- 原則：モデル間で文脈共有しない（独立性を保つ）。出力は **As-of Pack** に拘束する。
- 手順（最小）：
  1. As-of Pack を固定して、同一束をモデルA/B/Cへ投入する
  2. 各モデルに「反対の最強論」「失敗モード」「検知条件（Alert-if）」を出させる
  3. 出力を Evidence として束ねる（evidence = UNION）
  4. 判定は Canon へ正準化する（severity=max, until=max）
  5. 改善は Forward-only Update にのみ書く（ $\Delta$ は1つ）
- 位置づけ：AIは証拠源であり裁判官ではない。最終判定は Seat（Owner/Auditor）で固定する。
- 出力（固定）：MMARの結果は次の5点に正規化して残す。
 

① Invariants（不変点）／②  $\Delta$ （差分：最大1つ）／③ Evidence（evidence = UNION）／④ Gate（severity=max, until=max による PASS/DELAY/BLOCK）／⑤ Seat（Owner/Auditor が最終決裁）
- 実行条件（固定）：MMARは**節目のみ**で実行し、常用しない。採用は上書きではなく **Forward-only Update** にのみ記録する（ $\Delta$ は1つ）。

---

## 第3章 後知恵刺し（Hindsight Attack）とレビュー手続き

### 3.1 問題定義：後知恵は「学習」ではなく「攻撃」になり得る

後知恵は、本来は改善のための資源である。だが運用次第で、後知恵は学習ではなく攻撃になる。

攻撃化した後知恵（Hindsight Attack）は、主に次の3つの害を生む。

1. **責任の蒸発**：当時の制約や情報の不足を無視し、「今なら分かる」基準で断罪が行われ、Seat（責任線）が崩れる。
2. **探索のフリーズ**：失敗が“人格評価”や“過去の否定”へ変換されると、次の試行が抑制され、改善レーンが止まる。
3. **免責の増殖**：逆方向の攻撃として、結果が良かった場合にだけ「当時から分かっていた」物語が生まれ、説明が免責装置になる。

V9は、この攻撃を「心の問題」として扱わない。**レビューの形式（フォーム）によって、後知恵の流れ先を制御する。**

すなわち、説明は Playback-only、改善は Forward-only Update に送る（第2章）。このレーン分離を前提として、本章は「後知恵の攻撃化」を制度的に封じる。

---

### 3.2 レビューの原則：As-of固定、 $\Delta$ は将来へ、断罪は禁止

V9のレビューは、結果の正誤を競う場ではない。レビューの目的は、**当時性を保持したまま、将来の $\Delta$ （運用差分）を1つ残す**ことにある。

よって、レビューは以下を満たさなければならない。

- **As-of固定**：当時に観測できた情報・制約・目的に限定して説明する（Playback lane）。
- **差分保存**：改善は将来レーンにのみ書く（Update lane）。
- **人格評価の禁止**：正しさ/賢さの評価ではなく、手続きと検知ルールの更新に限定する。

ここでV9は「後知恵を消す」のではなく、**後知恵の出力先を固定する。**

---

### 3.3 レビュー出力の正準形：Forward-only Review（ $\Delta$ は1つ）

レビューの出力は、常に次の6行フォームへ正規化される（第2章と同型）。

特に重要なのは  **$\Delta$  Ruleは1つ**であり、更新は否定ではなく差分として保存される点である。

#### Forward-only Review（6行）

1. As-of（固定）
2. Outcome（事実のみ）
3. Missed Structure（見落とした構造）
4. Why unseen（見えなかった理由）
5.  $\Delta$  Rule（運用更新1つ）
6. Re-eval Trigger（再評価条件）

この正準形により、レビューが“責任追及”へ逸れるのを防ぎ、失敗は「検知ルール資産」として蓄積される。

### 3.4 評価モード：Detection-First Review（検知中心の資産化）

評価モード（Detection-First Review）は第2章 2.x に定義したとおり、結果の正誤ではなく検知の有無を主要評価対象とし、更新は Detection Gap / Why unseen / Guard Update（ $\Delta$ は1つ）に固定する。

### 3.5 例外：未レビュー決定に対する限定Guard介入（Box）

Box：未レビュー決定の限定Guard介入（Hindsightは禁止しないが、出力を固定する）

未レビューの決定は後知恵を誘発しやすい。本稿は、相談なしで行われた決定（詳細を共有できない場合を含む）について、後知恵を一律に禁止しない。代わりに、これを“安全介入”として限定発動させ、出力を次の4点に固定する。

1. 被害範囲（Scope）
2. 最大損失（Max Loss）
3. 最短ヘッジ（Fast Hedge：3手以内）



#### 4. 再評価条件 (Re-eval Trigger)

この固定により、責任転嫁を防ぎつつ、未レビュー状態でも迅速な安全化を可能にする。なお本介入は過去判断の正否を確定しない。改善は別途、Forward-only Update (将来差分) へ送る。

### 3.6 この章の「運用の最短手順」(3~5行)

1. レビューは **As-of固定** (当時束以外で殴らない)。
2. 説明は **Playback-only**、改善は **Forward-only** に分離する。
3. 失敗は Detection Gap → Why unseen → Guard Update (Δは1つ) で資産化する。
4. 未レビュー決定は **4点固定の限定Guard介入**で先に安全化し、断罪を禁止する。

## 第4章 Public Tube (p,n) : 反復試行で刺さる公開空間のモデル

### 4.1 公開空間は「小さなp」を「大きな確率」に変える

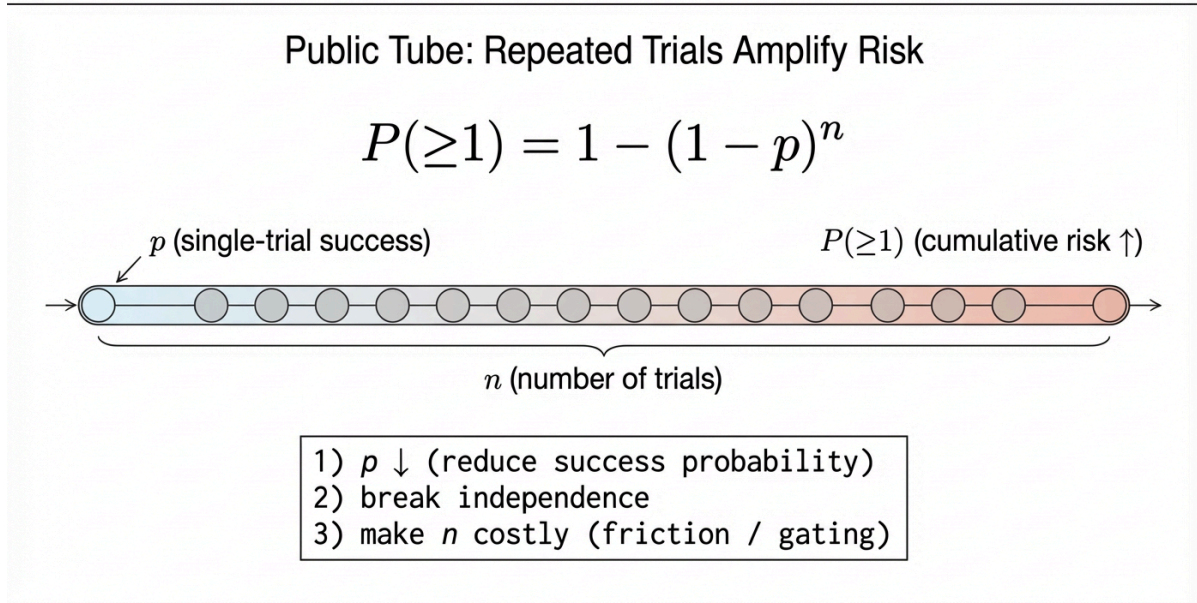
公開空間では、単発の失敗確率が小さくても、試行回数が増えることで事故は必然に近づく。

V9はこれを倫理や心理ではなく、**反復試行の構造**として扱う。成功確率を  $p$ 、独立試行回数を  $n$  とすると、少なくとも一度刺さる確率は

$$P(\geq 1) = 1 - (1 - p)^n$$

で表される。 $p$  が小さく見えても、 $n$  が大きければ  $P(\geq 1)$  は無視できない。

Figure 2.



**図2：Public Tube ( $p, n$ ) による累積リスクの増幅。** 単発の成功確率 $p$ が小さくても、試行回数 $n$ が増えると少なくとも一度刺さる確率 $P(\geq 1)$ が上昇する。対策レバーは順序固定で、(1) $p \downarrow$ 、(2)独立性破壊、(3) $n$ コスト化とする。

この構造は、レビューや意思決定の正しさとは独立に、**公開された環境そのものが持つ増幅器**である。

ここでV9の論点は「 $p$ を正確に推定すること」ではない。

**Public Tube は、 $p$  が不確かでも成立する。** なぜなら  $n$  が増える方向（拡散・反復・模倣）は、公開設計によってほぼ確実に発生するからだ。したがって、運用は「当て物」ではなく、**レバー操作**へ落とす必要がある。

## 4.2 防御レバーは3つだけ（順序固定）

Public Tube に対する防御は、複雑化させない。V9はレバーを3つに固定し、順序も固定する。

### 1. $p \downarrow$ （成功確率を下げる）

高外部性・高不可逆の要素を、抽象化・削除・一般化し、点で刺さる条件を減らす。

これは“善意の自制”ではなく、設計上の条件緩和である。

### 2. 独立性破壊（試行が独立にならないようにする）

反復で学習が積み上がる状態（独立試行）を崩す。

具体には、同型の改善が成立しないように出力を正準化し、攻撃側の学習曲線を鈍らせる。

### 3. nコスト化（試行回数にコストを乗せる）

観測窓（T\_req）と連続合格（Release）によって、単発成功を権限上昇に接続させない。

これにより、Public Tube の“nの増大”を、制度的な摩擦に変換する。

この順序が重要なのは、nコスト化だけに頼ると運用が重くなり、正当な利用まで阻害しがちだからだ。まず p↓ と独立性破壊で設計を軽くし、それでも残るものにだけ nコスト化を当てる。

---

## 4.3 第5章への接続：Public Tube は Release を必要とする

Public Tube の問題は、事故が「いつか起きる」ことではない。

事故が起きたときに、事後の物語化（後知恵刺し）と結びつき、責任線（Seat）まで崩れる点にある。

だからV9は、公開の解除条件を“単発の正しさ”ではなく、**影響量に比例した観測窓と連続合格**へ変換する。これが第5章の Release Gate（連続合格）である。

---

## 4.4 この章の「運用の最短手順」（3～5行）

1. 公開対象を Public Tube (p,n) として見る（p不確かでも成立）。
2. レバーは3つだけ：**p↓ → 独立性破壊 → nコスト化**（順序固定）。
3. 単発成功を権限上昇に接続しないため、Release を連続合格へ送る（第5章）。

---

# 第5章 Impact-Weighted Release：公開を「連続合格×観測窓」に固定する

## 5.1 目的：単発解除を禁止し、公開を制度化する

Public Tube（反復試行）では、単発の成功や一時的な良い結果が、そのまま影響拡大に接続しやすい。V9はこの接続を断つために、公開（Release）を「正しさの宣言」ではなく、**運用上の解除条件**として定義する。

結論は単純である。**単発でReleaseしない**。影響量に比例した観測窓を設け、観測窓内での**連続合格**によってのみReleaseする。

この設計は「慎重さ」ではなく、Public Tube の  $n$ （試行回数）を制度的コストへ変換するためのゲートである。

## 5.2 影響量 $I(a)$ の定義： $E \cdot R \cdot P$

公開判断  $a$  に対する影響量を、次で定義する。

$$I(a) = E(a) \cdot R(a) \cdot P(a)$$

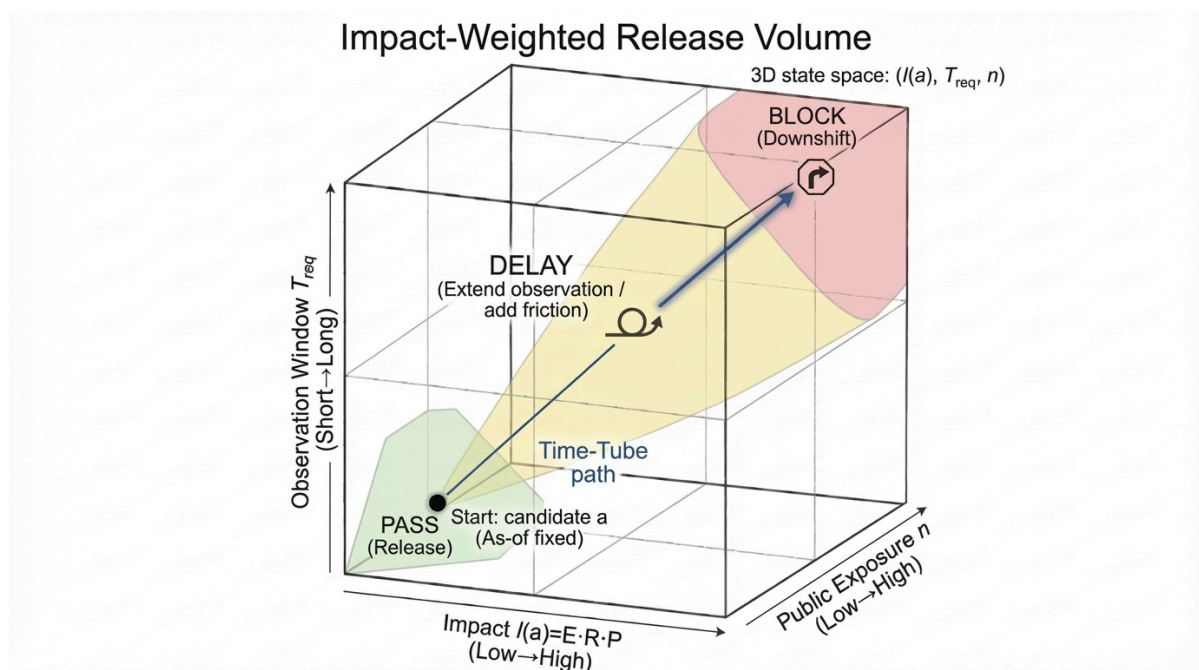
- **E (Externality)**：外部性。影響範囲（人・資産・信用・社会）
- **R (Irreversibility)**：不可逆性。巻き戻し困難度（取り返しのつかなさ）
- **P (Power)**：実行権限。到達可能性・拡散力・実行の容易さ

$I(a)$  は精密推定を目的としない。目的は、公開が持つ「影響の重さ」を粗くでも数直線上に置き、**観測窓と判定の厳しさを単調に増やす**ことである。

## 5.3 観測窓 $T_{req}$ ：影響量に比例して伸ばす

影響量が大きいほど、解除の観測窓は長くなる。 $V9$ は最小形として、次の単調関数で観測窓を与える。

$$T_{req}(a) = T_0 \cdot (1 + k \cdot \log(1 + I(a)))$$



**図3：Impact-Weighted Release Volume（署名図）。** 状態空間（ $I(a)$ ,  $T_{req}$ ,  $n$ ）上で、候補 $a$ （As-of固定）がTime-Tubeの軌跡として進み、PASS（Release）・DELAY（観測延長／摩擦追加）・BLOCK（Downshift）により公開解除条件を定義する。

- **T0**：基準観測窓（小影響の最短窓）
- **k**：伸長係数（運用で調整）

形は重要ではない。要求は3つだけである。

- (1) **単調増加**（ $I$ が大きいほど窓が長い）
- (2) **過大に伸びすぎない**（log等で緩やかに）
- (3) **例外で短縮しない**（短縮はDownshiftへ回す）

---

## 5.4 Release Gate：連続合格による解除（PASS/DELAY/BLOCK）

観測窓  $T_{req}$  内で、公開対象  $a$  を繰り返し評価し、判定は正準形に正規化される。

- **PASS**：観測窓内で連続合格。Release可能
- **DELAY**：観測継続。窓を延長する（または条件を追加）
- **BLOCK**：公開不可。Downshift（第6章）へ送る

ここでの要点は、Release が「正当化」ではなく「解除条件」であることだ。

Release は、真理性ではなく、**運用上の安全条件の充足**として扱われる。

---

## 5.5 連続合格の定義：単発の成功を権限上昇に接続しない

連続合格とは、 $T_{req}$  内の評価で **重大な逸脱（BLOCK相当）** が観測されないことを意味する。

単発の良い結果は、解除に接続しない。単発の悪い結果も、直ちに人格評価へ接続しない。V9が扱うのは「傾向」であり、傾向は観測窓の中で評価される。

この形式により、Public Tube の反復に対して、攻撃側に「一度刺せば勝ち」という構造を与えない。必要なのは点ではなく面であり、面はコストになる。

---

## 5.6 Seat と As-of への接続：解除は責任線を崩さない

Release Gate は、Seat（Decision Owner / Auditor）と As-of Fixation を前提に動作する。

判断は当時性に固定され、説明は Playback-only、改善は Forward-only Update に送られる（第2章）。本章のReleaseは、その上に載る「公開の解除条件」である。

つまりV9の公開は、誰かを賢く見せるためではなく、責任線を崩さずに安全条件を積み上げるために存在する。

---

## 5.7 この章の「運用の最短手順」（3～5行）

1. 公開対象 a の  $I(a)=E \cdot R \cdot P$  を置く（精密でなくてよい）。
  2.  $T_{req}(a)$  を単調関数で決める（Iが大きいほど長い）。
  3.  $T_{req}$  内で評価し、**連続合格のみPASS**。単発でReleaseしない。
  4. DELAYは観測延長、BLOCKは第6章のDownshiftへ送る。
- 

# 第6章 Downshift：高影響に上げない安全（Stage-Lowering Safety）

## 6.1 目的：悪を消さず、「高影響の段」に上げない

V9の安全は「危険をゼロにする」ことを目標にしない。目標は、危険が残る状況でも、**高影響の段（stage）**に上がらないように運用で制御することにある。

第5章のReleaseは「連続合格でのみ解除する」ゲートだった。本章のDownshiftは、その逆側である。**BLOCKを終点にせず、影響量を下げる操作へ変換する。**

ここでの設計思想は単純である。

**単発の成功で上に行けない。単発の失敗で人格評価に行かない。**

上がれない構造を作ること、Public Tube の反復試行に対しても、制度が学習してしまわない。

---

## 6.2 Downshiftの定義：影響量 $I(a)$ を下げる操作

第5章で定義した影響量は  $I(a)=E \cdot R \cdot P$  である。Downshiftは、このIを「安全側へ降ろす」操作として定義する。

Downshiftは原則として次の4つのレバーのいずれか、または組合せで実施される。

- **P↓（Power Down）**：実行権限・到達可能性・拡散力を下げる
- **E↓（Externality Down）**：外部性を下げる（対象範囲の限定、抽象化、隔離）
- **R↓（Reversibility Up）**：不可逆性を下げる（巻き戻し可能性を上げる）

- **T\_req↑ (Observation Up)**：観測窓を伸ばす（解除を遠ざける）

重要なのは、Downshiftが「禁止」や「黙殺」ではなく、**公開と運用の階層を下げる実務操作**だという点である。

---

## 6.3 発火条件：BLOCKはDownshiftへ送る

Release Gateで **BLOCK** が出た場合、V9はそれを「議論の終わり」にしない。BLOCKは、次のどちらかの意味を持つ。

1. **不適格 (Unqualified)**：この段では扱えない（段を下げる必要がある）
2. **未成熟 (Unmatured)**：観測が足りない（T\_req↑、またはP↓で隔離しつつ観測する）

いずれの場合も、V9の標準応答は「段を下げる」である。

これにより、公開の圧力や単発の熱量で“高影響”が成立する経路を閉じる。

---

## 6.4 Downshiftの正準形：4レバーで表現する

Downshiftは、表現を統一する。議論を増やさず、運用に落ちる形に正準化するためである。

Downshiftの出力は、次の4項目で記述される。

- **P↓**：どの権限を下げるか（例：対象を限定、配布を止める、実行を隔離する）
- **E↓**：外部性をどう下げるか（例：抽象化、範囲を縮める）
- **R↓**：可逆性をどう上げるか（例：ロールバック可能な形に変える）
- **T\_req↑**：観測窓をどれだけ延長するか（例：連続合格条件を厳しくする）

この4項目が揃わないDownshiftは、実務上の操作にならず、議論だけが残る。

---

## 6.5 周回安全：単発成功が権限上昇に繋がらない

Public Tube の本質は、反復によって刺さる確率が上がることだった。

Downshiftは、反復によって「上に行く」確率が上がる現象を抑える。つまり、試行が繰り返されても、**単発の成功が上位段への移行条件にならないようにする**。

ここでV9が求めるのは、罰ではなく摩擦である。

上位段の条件は面（連続合格）であり、下位段の操作はレバー（P/E/R/T）である。これによって、上位段の成立コストは時間と連続性に変換され、点突破のメリットが消える。

---

## 6.6 第7章への接続：DownshiftはSeatとEvidenceで監査可能でなければならない

Downshiftは強い操作である。だからこそ、誰が決め、何を根拠にし、どの条件で再評価するかが固定されなければならない。

次章では、Seat（Decision Owner / Auditor）とEvidence（証拠束）とCanon（正準合流）によって、Downshiftを監査可能な運用として閉じる。

---

## 6.7 この章の「運用の最短手順」（3～5行）

1. Release Gateで **BLOCK** が出たら、終了ではなく **Downshift** に送る。
  2. Downshiftは **P↓ / E↓ / R↓ / T\_req↑** の4レバーで記述する。
  3. 単発成功を上位段の条件にしない（上は連続合格のみ）。
  4. 次章で Seat/Evidence/Canon により監査可能に固定する。
- 

# 第7章 Seat / Evidence / Canon：責任固定と監査可能な正準運用

## 7.1 目的：能力は委譲できるが、決定権と責任は委譲しない

V9は「賢い回答」を増やす体系ではない。事故りにくいのは、判断の正しさよりも、責任線（Seat）と証拠束（Evidence）と判定の正準形（Canon）が固定されている体系である。

本章は、これまでの章（As-of／レーン分離／後知恵制御／Public Tube／Release／Downshift）を、監査可能な運用として閉じる。

---

## 7.2 Seat（責任の席）：Decision Owner / Auditor

Seatは、判断の主体と監査の主体を固定する。

- **Decision Owner**：最終判断の席。決定権は人間にある。AIは主体を持たない。
- **Auditor**：監査の席。判断の正否を断罪するのではなく、As-ofと証拠束とΔ更新の形式が守られたかを監査する。

Seatの固定は、レビューを「人格」ではなく「手続き」へ押し戻すための装置である。席が揺れると、責任が揺れ、説明が免責装置になる。

---



## 7.3 Evidence（証拠束）：説明ではなく証拠に接続する

V9で“説明”はPlayback-onlyに制限される。代わりに、判断と更新は証拠束へ接続される。

Evidenceは「説得」ではなく「監査のための束」であり、後から差し替えない。

Evidenceに含めるのは、最低限でよい。重要なのは **参照可能性** と **同一性** である。

- 参照可能性：どこを見れば当時の根拠に戻れるか
- 同一性：同じ束を指していることが確認できるか（SSOT/ハッシュ等）

## 7.4 Canon（正準形）：判定と合流を“安全側”に固定する

V9の運用は、最終的に正準形へ落ちる。正準形は「議論の余地」を減らすためではなく、**安全制約が勝手に弱まらないようにするために存在する**。

Canonは次の3つで構成される。

1. **Severity（判定）**：PASS < DELAY < BLOCK
2. **Until（期限）**：max（最も遅い期限を採用）
3. **Evidence（証拠）**：U（証拠の和集合）

この正準形により、多人数・多モデル・多レビューになっても、合流で安全側が失われない（弱い結論に引きずられない）。運用は常に、より強い制約へ単調に寄る。

## 7.5 監査の対象：内容ではなく形式

V9の監査は「結論の正しさ」を主目的にしない。監査の主対象は次の3点である。

- **As-ofが固定されているか**（当時性の差し替えが起きていないか）
- **レーン分離が守られているか**（PlaybackとUpdateが混線していないか）
- **Δが過剰になっていないか**（更新は“否定”ではなく“差分”、Δは最小であるか）

この形式監査が成立すると、失敗は責任追及ではなく、将来差分の資産として蓄積される。

## 7.6 全章を貫く最小閉包：As-of / Seat / Release

V9の最小閉包は3点である。

- **As-of**：判断を当時束に固定する
- **Seat**：決定権と監査責任の席を固定する

- **Release**：公開を影響量に比例した連続合格でのみ解除する

この3点が揃うと、後知恵刺しは攻撃として成立しにくくなり、Public Tubeによる反復試行も、制度の摩擦へ変換される。V9は“正しさ”ではなく“事故りにくさ”を実装する。

---

## 7.7 この章の「運用の最短手順」（3～5行）

1. Seatを固定する（Decision Owner / Auditor）。
  2. 判断と更新はEvidence（参照可能で同一な束）に接続する。
  3. 判定はCanonへ正準化する（PASS<DELAY<BLOCK、until=max、evidence = UNION）。
  4. 監査は結論ではなく形式（As-of／レーン分離／Δ最小）を確認する。
- 

## 結び（Conclusion）

本稿（V9）は、未来を当てるための理論ではない。安全とは正解の能力ではなく、世界が反転したときに停止し、遅延し、巻き戻せることだという立場から出発し、V8で示した「介入としての安全」を、文明運用の手続きへ拡張した。

V9が固定するのは、As-of（当時性）、Seat（責任の席）、Release（連続合格）である。これらは、後知恵刺しと反復試行の増幅が結びついて、責任線が崩れる現象に対する最小の憲法である。V9は結論の正しさを保証しない。代わりに、判断の当時性と責任線を保持したまま、失敗を差分として保存し、制度が更新していく形を保証する。

そして最後に、私がこの系譜で一貫して主張してきたことを、短く言い直す。

**知性は点である。ゲートは線である。**

知性は瞬間の推論として現れるが、安全は時間を味方にする運用としてしか成立しない。点の判断を線へ変換し、線の上で合格を連続させる——それがV6以降の設計であり、Time-Tube（V8）からRelease Gate（V9）への接続でもある。

安全は知性ではなく、ゲートである。