ループを書くだけで満足するな。

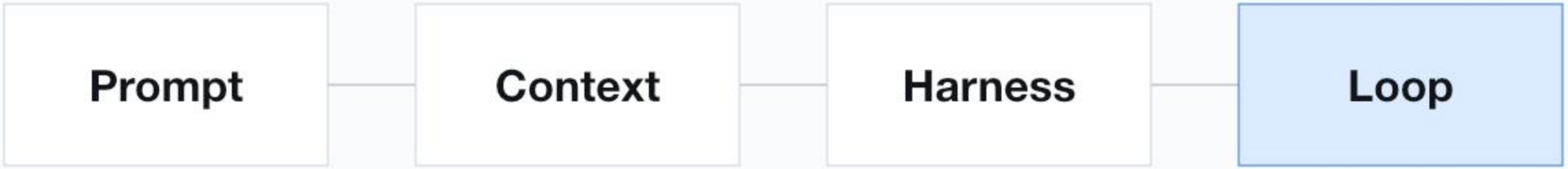
意味と意図を監査せよ。

そして、**Chronicle**を持つループを書け

AUDIT
TRACE
STOP

ループエンジニアリングは、プロンプトの外側を設計した

人間が一回ずつ指示する状態から、AIが反復できる構造を設計する状態へ。



第1世代の成果

AIに安全な試行錯誤をさせるため、実行環境・テスト・権限・停止条件を設計対象にした。

第1世代ループは「実行の成功」を最適化する

テストが通る、エラーが消える、作業が完了する。
しかし、それは「意味が守られた」ことを保証しない。

問題は、評価関数に含まれていないものが削られること。

テストが通っても、意味はずれる

AIの反復は、元の意図・責任・不確実性を少しずつ書き換えることがある。

元の意図	高リスク操作は人間が明示承認する
ループ中の補完	confidence が高ければ自動実行してよい
見かけの成功	テストは通る。UIも自然に見える
意味の逸脱	責任境界がAI判断へ移っている

壊れていないのに危ない、という状態がここで起きる。

Chronicleがないループは、短期視点に閉じる

直近のログ、直近の失敗、直近の指示だけで推論すると、過去の判断理由が消える。

通常ログ

何が起きたか
どのテストが通ったか
どのファイルを変えたか

欠けるもの

なぜ変えたか
何を却下したか
何を保留したか

Chronicle

判断の来歴
意味変化の記録
次ループへの申し送り

ループには「記憶」ではなく、監査可能な来歴が必要になる。

第2世代ループは、意味・意図監査と来歴保存を内蔵する

第1世代	実行が成功したかを見る
第2世代	意味と意図が保たれ、判断の来歴が残っているかを見る

速く回す設計から、保ったまま回す設計へ。

Meaning / Intent Auditは、意味と意図の差分を見る

意図・目的

何を実現したかったのかがすり替わっていないか

不確実性

未確定の前提を確定扱いしていないか

責任主体

人間判断をAI判断へ移していないか

価値構造

安全より速度を暗黙に優先していないか

非目標

やらないと決めたことを混ぜていないか

制度的含意

法務・労務・セキュリティの意味が増えていないか

テストは振る舞いを見る。Auditは、振る舞いの意味と意図が変わっていないかを見る。

Chronicleは、次のループが読む判断の来歴である

```
loop_id: 2026-08-02-agent-loop
baseline:
  purpose: 意味を保って実装する
  non_goals:
    - 承認なしの自動実行
actions:
  - 実装
  - テスト
semantic_audit:
  responsibility_shift: false
  uncertainty_erasure: false
decision: human_review
```

保存すべきもの

- 何を目的に始めたか
- どの判断で変えたか
- 何を却下したか
- 何を保留したか
- 誰が承認したか
- 次に何を読むべきか

ChronicleはAIのメモリではなく、チームの監査証跡である。

Human on the Loopは、Chronicleの読者になる

人間は現在の出力だけでなく、「なぜそうなったか」を読む。

人間の価値は、AIより遅いことではない。
AIが持たない文脈と責任を持っていることにある。

実装は、PR・CI・運用ログから始められる

PR	Semantic / Intent Diffを必須項目にする
CI	高リスク変更に監査抜けを警告する
Agent	Loop Reportを毎回出させる
Docs	Chronicle.mdへ判断理由を追記する
Review	approve / revise / stop を明示する

最初から巨大な基盤はいらない。
高リスクなループだけ、意味・意図監査を厚くする。

これから重要なのは、速さではなく保存される意味である

何を保ったまま回るのか。

何が変わったら止まるのか。

なぜその判断をしたのかを、
未来のチームが読めるのか。

Chronicleを持つループを書け。