検証ハーネス

観測モデルの記録を適合条件に照らして採否を判断する

実装修正

二重処理と余分なI/Oを再検証

実行時のAI構成

生成品質・性能・AIコストで比較

記録が変更の採否を支える

観測モデル

入力・負荷、結果、呼び出し経路を
一つの実行に結び付ける

適合条件

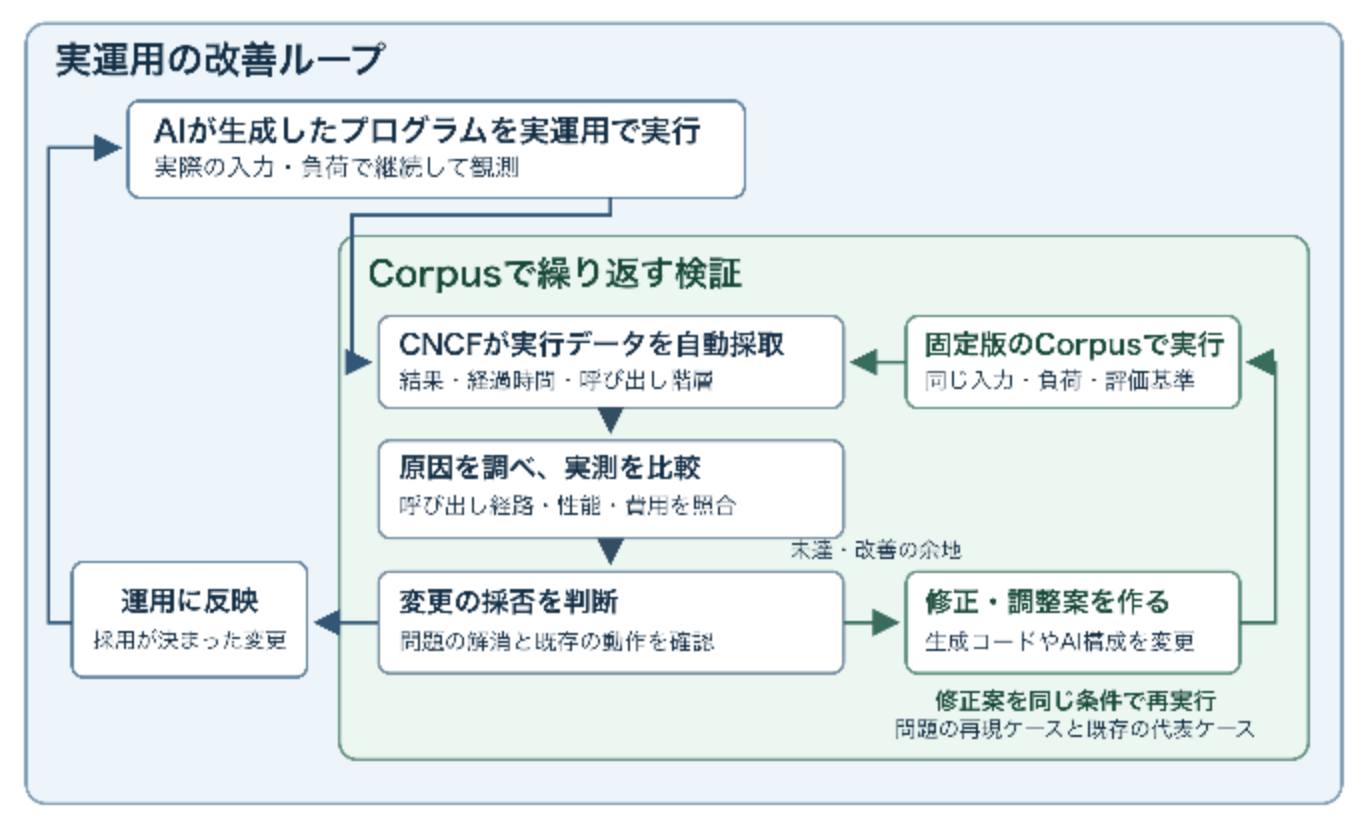
要求と設計から期待結果、契約、
性能目標を定める

検証ハーネス

記録を条件と照合し、採否を判断

同じRunの出力と内部動作を条件に照らす。

実運用とCorpusの二つのループ



同じ条件で再検証し、適合した変更を運用へ戻す。

1,000明細のI/Oを一括化する

発見

CallTreeで
1,000回のI/Oを確認

→

改修

一括照会へ変更し
取得結果と副作用を保つ

→

採否の判断

結果、レイテンシ、
スループットを照合

同じCorpusとリクエストミックスで改修前後を比較。

実行時のAI構成を比較する

構成	計測結果（100件）
構成A	100件正解、費用指数100。費用と処理量が未達。
構成B	100件正解、費用指数45、p95は6秒、毎分45件。
構成C	98件正解、費用指数30。生成品質が未達。

同じ100件を比較した仮想例。全条件に適合した構成Bを採用。

検証ハーネス

実行の事実を、変更の採否につなぐ

観測モデル

実行条件と入出力・処理経路を
同じ実行の記録として結び付ける

実行記録 →

検証ハーネス

期待結果・契約・性能目標に照らし
実測から変更の採否を判断する

実運用とCorpusの二つのループ

実運用

実際の入力と負荷で観測
適合した変更を戻し、効果を追う

⇢ 同じ入力・負荷・評価基準で比較
⇠ 障害再現と修正後の再検証

Corpus再実行

同じ入力・負荷・評価基準で比較
障害再現と修正後の再検証

実装の修正とAI構成の比較

明細I/Oの一括化　取得結果と性能を確認

実行時のAI構成　生成品質・性能・AIコストで比較