計測表

自分の数字をその場で書きます。演習の区切りごとに手を止めて記入してください。

演習中に自分で数える欄です。あとから思い出して埋めることはできないので、その場で書いてください。

数字を並べる目的は、レビューの置き方を自分の数字で選べるようにすることです。論文の数字ではなく、自分の MR で何が起きたかを見ます。

Day1 3通りのレビュー

ハンズオン1で、同じ差分に3通りのレビューを当てます。順番は固定です。実装したセッションを最初に、履歴を切った側を後にします。

項目	(1) 実装したセッション	(2) `/clear` 後に同僚のコードとして	(3) 別ツール
指摘の総数			
うち P0 と P1 に相当するもの			
重大な指摘の有無(仕込み欠陥を当てたか)			
所要時間(分)			
使ったモデルと effort			

(3) の別ツールは、Codex を持っている方は `codex review --base main "P0/P1 のみ、3件まで、コード提案不要"`、持っていない方は `git diff main | claude --bare -p --json-schema review.schema.json` です。

Codex CLI は P2 と P3 も出します。件数だけを比べると多く見えるので、2行目の「P0 と P1 に相当するもの」で比べてください。

書き終えてから答える欄

問い	記入欄
3つのうち、仕込み欠陥を当てたのはどれか	
同じ差分なのに指摘が変わった理由として、自分が納得できるもの	
自社で常用するなら、どの置き方にするか	
その置き方を選んだ理由(コストと再現性のどちらを優先したか)	

履歴を捨てた別セッションのほうが自己レビューより当たる、という測定結果があります(Cross-Context Review で F1 28.6 対 24.6、自己レビューの2回目は 21.7)。自分の数字がこれと違う向きに出たら、そのまま書いてください。1 件の MR で逆転することは普通に起きます。数字が一致しなかったことより、「履歴を切るだけで指摘が変わる」ほうが持ち帰る事実です。

Day1 委譲判定の集計

委譲判定シートの集計欄から転記します。

項目	件数
指摘の総数	
即修正	
要調査	
将来課題	
意図した設計	
CI で落ちたもの	
指摘の仕分けにかかった時間(分)	

Day2 検出率

ハンズオン2で、仕込み欠陥がいくつあるかは Issue に書いてあります。レビューがそのうち何件を当てたかを数えます。

項目	Issue #2	Issue #3
仕込み欠陥の件数		
レビューが当てた件数		
指摘の総数		
検出率(当てた件数 ÷ 仕込み欠陥の件数)		
ノイズ率(仕込み欠陥以外の指摘 ÷ 指摘の総数)		
REVIEW.md を書く前の指摘の総数		

項目	Issue #2	Issue #3
REVIEW.md を書いた後の指摘の総数		

下の2行が、この日で一番使う数字です。REVIEW.md を書いて指摘が減り、検出率が落ちていなければ、基準が効いています。両方とも減っていたら、Do not report に入れすぎです。

テスト有無の比較

同じリファクタを、機能テストが無い状態と有る状態の2回頼みます。

項目	テスト無し	テスト有り
CI の結果		
壊れたことに気づいた場所		
気づくまでにかかった時間(分)		

Day2 夜間ループ

ハンズオン3で、1回目がどこで止まったかを記録します。止まらずに完走したら、その旨を書いてください。

項目	記入欄
止まった箇所	
止まった理由	
どの層の設定で直すか	
直した内容(ファイル名とキー)	
再実行の結果	
ジョブの所要時間(分)	
使ったターン数(`--max-turns` の上限に当たったか)	

止まる箇所の候補です。当てはまるものに印を付けてから、上の欄を埋めてください。

- 権限で拒否されて、そこから先へ進まなかった
- PHP 5.4 で動かない構文が混入して、lint で落ちた
- テストが落ちて、そのまま終わった
- レビューの手順を飛ばして MR を作った

- 指摘を全部受け入れて、設計が元に戻った
- MR の作成が権限で通らなかった
- ターン数の上限に当たって、途中で終わった

最後に1枚書く欄

問い	記入欄
今のハーネスで夜間に回せる範囲	
人が残る3点	
年末までに埋める層の順番	