

設計レビューハーネスの使い方

Day3 のD3-1とD3-2で手でやった設計レビューを、順番と点検を決めてClaudeに回させる道具です。画面設計書を渡すと、あなたが先に指摘を書いたあと、Claudeが6つの観点を1つずつ分けて指摘を出します。付属の点検ツールが、指摘の根拠の文が設計書に本当にあるかを確認め、作り話の根拠を外します。採否を選ぶのはあなたで、その結果をまとめたものがレビュー記録です。要件一覧を渡すと、要件と設計の対応表も作れます。

入れ方

1. 教材サイトの持ち帰り資料から、review-harness-kensyu_Day3.zipをダウンロードします。
2. ZIPを右クリックして、[すべて展開] を選びます。
3. [参照] を押して、配布フォルダのidup-handsonを選び、[展開] を押します。
4. Codeタブでidup-handsonを開き、[新規] で新しい会話を始めます。
5. 入力欄に/review-harness-kensyuと打つと、候補に出ます。

動く順番

段階	Claudeがすること	あなたがすること
1 準備	設計書を最後まで読み、シートと行の指し方を表にする	照らす要件のファイルを選ぶ
2 自分の指摘	自分の指摘.mdを6観点の形で作って止まる	メモ帳で3件を目安に書き、「書けた」と送る
3 観点ごとのレビュー	6つの観点を1つずつ分けて見る。1観点5件まで	待つ
4 根拠の点検	点検ツールで根拠の文を照らし、設計書にない根拠の指摘を外す	外された指摘と件数を見る
5 判定	観点ごとに一番疑わしい指摘を1件ずつ質問にする	設計書の根拠の行を開き、採用、不採用、保留を選んで理由を書く
6 レビュー記録	レビュー記録.mdにまとめ、もう一度点検する	見つけた人の列と件数を見る
7 対応表	要件一覧を選んだときだけ、対応表.mdを作って点検する	対応なしの扱いを5件まで選ぶ
8 結果	ファイルの場所と、判定ごとの件数を返して止まる	保留の指摘を誰に確かめるかを決める

使い方は2つ

使い方	渡すもの	書く場所
研修	配布フォルダの 資料/か 成果物/にある設計書	成果物/ハーネス/<設計書のファイル名>/に、自分の指摘、AIの指摘、レビュー記録、対応表
案件	自分の案件で書いた画面設計書を、mdかxlsxで渡す	Claudeが場所を聞きます。要件化ハーネスで作った案件フォルダなら、成果物/4_レビュー/

練習のしかた

練習は、配布フォルダの架空の設計書で行います。1本目は、D3-1でレビューしたPJ010です。自分で作ったレビュー記録と、件数と判定を見比べられます。

/review-harness-kensyu @資料/day3/レビュー対象_画面設計書_PJ010_収支ダッシュボード.md をレビューしてください。照らす要件は 成果物/day1/要件一覧.md です。

2本目は、Day2で自分が書いたKIN030の設計書です。自分の設計書を、ほかの人の目で見える練習になります。

/review-harness-kensyu @成果物/day2/画面設計書_KIN030_v0.2.md をレビューして、要件と設計の対応表まで作ってください。照らす要件は 成果物/day1/要件一覧.md です。

途中で止まる場所

Claudeが止まるとき	あなたがすること
照らす要件のファイルを聞くと	選択肢から選びます。要件がなければ「要件なしで進める」を選びます
自分の指摘を先に書くよう頼むとき	自分の指摘.mdをメモ帳で開き、3件を目安に書いて保存し、「書けた」と送ります
疑わしい指摘を見せて、判定を聞くと	設計書の根拠の行を自分の目で見てから選びます。保留の理由には、誰に確かめるかを書きます
対応表を作るかを聞くと	作るなら、そのあと対応なしの扱いを、設計に足す、次版、対象外から選びます
最後に結果を返して止まるとき	保留の指摘を確かめる相手と期限を、あなたが決めます

点検ツールが確かめること

点検ツール	確かめること
check_review.py	シートの名前が設計書の見出しにあるか。行のIDかNoがそのシートにあるか。根拠の原文が、そのシートの中に一字ずつあるか。判定が 採用、不採用、保留 のどれかで、理由があるか
check_trace.py	要件一覧の全件が対応表に載っているか。対応先のIDが設計書にあるか。集計の件数が表と合うか。扱いの決まっていない対応なしが何件あるか

点検ツールだけを試す

ハーネスを回す前に、点検ツールだけを試すこともできます。ZIPに入っている見本は、5件のうち2件の根拠を、わざと設計書にない文と別の行の文にしてあります。次の文面をCodeタブの入力欄に送ってください。その2件がNGとして表示されます。

```
python .claude/skills/review-harness-kensyu/scripts/check_review.py サンプル/点検の見本_AIの指摘.md --doc 資料/day3/レビュー対象_画面設計書_PJ010_収支ダッシュボード.md を実行して、結果をそのまま見せてください。
```

守ること

- 案件の設計書は、AIに渡してよい情報かを、社内のルールで先に確かめます。分からないときは、社名と人名を置き換えたコピーで試します。
- 採否、重さの最終判断、対応なしの扱いは、あなたが決めます。Claudeには決めさせません。

- 点検ツールが外した指摘は、AIが根拠を作った指摘です。消さずに残るので、どんな文を作ったかを見ておきます。
- 点検が通っても、指摘が正しいとは限りません。点検ツールが確かめるのは、根拠の文が設計書にあることです。
- 設計書は、あなたが頼んだときだけ直させます。直した設計書は、別の名前で保存させます。

うまくいかないとき

起きたこと	直し方
/review-harness-kensyuが候補に出ない	idup-handsonの中に.claude/skills/review-harness-kensyuがあるかを見ます。あれば、[新規] で新しい会話を始め直します
pythonが見つからないと出る	pyに替えて、もう一度実行します
点検でNGが消えない	NGの行に表示された直し方を、そのままClaudeに送ります。3回直しても消えなければ、その指摘を外させます
設計書がxlsxしかない	そのまま渡します。Claudeがtools/xlsx_to_md.pyでmdにしてから読みます
判定の質問が出ずに先へ進んだ	「段階5の判定から、やり直してください」と送ります
xlsxへの変換が止まる	LibreOfficeで開いていれば閉じて、ファイル名を変えずにやり直します

この道具は、Day3で身に付けた、観点を分けて頼む、根拠に戻って確かめる、採否は人が決める、の3つを手順に固定したものです。要件化ハーネスで作った要件一覧を、このハーネスの対応表に渡すと、要件から設計までを同じIDでたどれます。

