持ち帰り演習集

研修のあと、自分の案件の資料で回す演習が3本あります。研修でKINTAIを相手にやったことを、自分の議事メモ、自分の画面、自分のチームの設計書に置き換えます。1回で終わらせず、次の打ち合わせ、次の設計書と、案件の節目ごとにv1、v2、v3と回してください。

演習	所要	1回目の時期	使う文面
A　議事メモからの要件一覧	[45min]	研修後、最初の打ち合わせのあと	サンプル集の1、2、4
B　既存画面1つの改修設計とモック	[60min]	改修の依頼が来た画面があるとき	サンプル集の6、7
C　チームの設計書の観点別レビュー	[50min]	チームの設計書がレビューに回ってきたとき	サンプル集の8、9

道具と記録の残し方

3本とも、研修で使った配布フォルダidup-handsonのtools/とテンプレート/を、自分の作業フォルダにコピーして使います。MarkdownからWordやExcelにする変換は研修と同じ手順です。

記録は作業フォルダに置く 持ち帰り演習メモ.txtに書き足します。1回ごとに日付と「演習A v1」のような見出しを付け、人の名前は書かずに役割で書きます。

記入例（架空の案件）

> 10月29日　演習A v1
> 使ったメモ：保守案件の定例の議事メモ2本（社名と人名は置き換えたコピー）
> 自分の候補：12件　AI の表：21件
> 自分にだけあった：2件（帳票の出力時期、月末の問い合わせ窓口）
> AI にだけあった：9件。うち5件は採用、4件はメモの意味を広げていたので捨てた
> 要確認：4件。聞く相手は先方の業務担当2件、自社のPL2件
> 根拠の抜き取り：3件中3件一致

演習A　議事メモからの要件一覧

所要	[45min]
1回目の時期	研修後、最初のヒアリングか打ち合わせのあと。終わった案件の議事メモでも構いません
使うもの	自分の案件の議事メモ2本以上、テンプレート/要件定義書テンプレート.md、 プロンプトサンプル集の1、2、4

目的

研修のD1-2とD1-4を、自分の案件のメモで回します。自分の候補は先に手で書き、AIの表が出てから渡します。順番も研修と同じです。実際の議事メモの食い違いは、誰にも気づかれないまま次の工程へ流れがちです。どちらかにしか無い要件と、メモ同士の食い違いを、この演習で表に出します。

手順

1. AIを開かずに、使う議事メモが社内規程とお客様との取り決めでAIに入れてよいものかを確かめる。分からなければ、社名、人名、金額を置き換えたコピーを作る。元のメモは書き換えない。 [5min]
2. メモを読み、要件の候補を10件以上、自分の手で書く。あいまいな言い方と、メモ同士で食い違っていそうなところに印を付ける。書いたものは、手順4までAIに見せない。 [10min]
3. サンプル集の1の1通目で、メモだけを渡して要件の表を出させる。新しい会話を開き、サンプル集の2の1通目でメモ同士の食い違いを出させる。 [10min]
4. それぞれの会話で2通目を送り、自分の候補と、印を付けた食い違いを渡して突き合わせる。自分にだけある、AIにだけある、食い違う、の3つに分かれたら、採るか捨てるかを1件ずつ自分で決める。食い違いは要確認にし、誰に何を聞くかを書く。 [10min]
5. 根拠を抜き取りで3件開いて確かめ、要件一覧を確定する。時間が残れば、自分で抜けを1つ探して書いてから、サンプル集の4で抜け漏れを探す。 [10min]

自分で考えるところ

手順2の候補出しと、手順4で採るか捨てるかの判断は、自分でやります。AIにだけある要件は、メモの発言の意味を広げていないかを見て決めます。

食い違いのどちらが正しいかは、この演習では決めません。決めるのはお客様です。聞く相手と聞くことが書けていれば、この演習は終わりです。

できあがるものと置き場所

- 作業フォルダの 要件一覧.md。全件に根拠（ファイル名と行、または引用）が付いている
- 同じフォルダの 確認事項.md。要確認の項目ごとに、聞く相手と聞くこと
- 持ち帰り演習メモ.txtの「演習A v1」

完成の条件

- 要件一覧の全件に根拠がある
- 自分の候補とAIの表を比べた結果が、3つに分けてメモに書いてある
- 要確認が3件以上あり、どれにも聞く相手と聞くことが書いてある
- 根拠を抜き取りで3件確かめ、3件とも元の文と一致した

ヒント

- メモが1本しか無いときは、同じ案件の要件定義書や改修依頼書を2本目として使います。書いた人が違えば、食い違いは出ます。
- 要確認が0件のときは、「できるだけ」「早く」「従来どおり」をメモの中で検索します。数字の無い要望は、ほぼ要確認です。
- AIにだけある要件が自分の候補の倍を超えたら、メモの一般論を拾っていることが多いです。根拠の文を読み、発言そのものが要件になっているかを見ます。

安全の注意

- お客様の議事メモをAIに入れる前に、社内規程と契約で許されているかを確かめます。規程が分からないときは、上長か情報セキュリティの担当に聞きます。
- 置き換えたコピーでも、案件が特定できる固有の名前（システム名、拠点名）が残っていないかを見ます。
- 作った要件一覧をお客様に出すときは、AIの出力をそのまま出さず、自分で確かめた版を出します。

繰り返し方

回	時期	やること
v1	研修後、最初の打ち合わせのあと	議事メモ2本で要件一覧を作る
v2	次の打ち合わせのあと	確認事項を持って打ち合わせに出て、返ってきた答えで要確認が何件減ったかを数える。新しいメモで食い違いを探し直す
v3	要件定義の終わり	要件一覧を要件定義書のドラフトにする（サンプル集の3）。要確認が残った章を、次の打ち合わせの議題にする

発展課題

自分の候補にだけあった要件を集めて、なぜAIが拾えなかったのかを1行ずつ書きます。メモに書いていない前提を自分が知っていた、という理由が多ければ、その前提を議事メモに書き残す習慣を付けると、次の担当者に渡しやすくなります。

演習B　既存画面1つの改修設計とモック

所要	[60min]
1回目の時期	改修の依頼が来た画面があるとき。無ければ、過去に改修した画面で練習します
使うもの	改修する画面の項目の一覧、改修依頼か要件、 テンプレート/画面設計書テンプレート.md、プロンプトサンプル集の6と7

目的

研修のD2-2とD2-3を、自分が保守している画面1つで回します。項目定義と入力チェックを3種類に書き分け、モックを作って、設計書に書いていなかったことを見つけます。モックはブラウザで開くためのHTMLで、本番のコードではありません。

手順

1. AIを開かずに、改修する画面の項目を手で書き出す。画面の実物のソースや画面写真をAIに渡してよいかを確かめ、だめなら項目名と型だけを書く。 [10min]
2. 改修で足す項目か変える項目を5つ、項目定義（型、桁、必須、初期値、誰が入れるか）まで手で書く。 [10min]
3. サンプル集の6で項目定義と入力チェックを作る。表3の迷ったものは自分で種類を決める。tools/md_to_xlsx.pyでExcelにして開く。 [15min]
4. サンプル集の7でモックを作り、ブラウザで開く。先に書いた確かめたいこと3つを操作して確かめる。 [10min]
5. 「設計書に無いので仮に決めたこと」を1件ずつ見て、設計書に書き足すか、お客様に聞くかを決める。設計書をv0.2にする。 [15min]

自分で考えるところ

手順2の項目定義と、表3の迷ったチェックの種類は、自分で決めます。どこに実装するかが変わるので、チームの設計標準と照らします。

モックの見た目の良し悪しは見ません。設計書に書いていないのにAIが決めたことを探すのが、この演習の中心です。

できあがるものと置き場所

- 作業フォルダの 画面設計書_<画面ID>_v0.2.xlsx。項目定義、入力チェック、メッセージのシートがある
- 同じフォルダのmock_<画面ID>.html。ダブルクリックで開ける
- 持ち帰り演習メモ.txtの「演習B v1」。設計書とモックの差と、その扱い

完成の条件

- 項目定義の全項目に、型、桁、必須、初期値、入力者が入っている
- 入力チェックが、単項目3件以上、相関1件以上、業務ロジック1件以上あり、全チェックにメッセージIDがある
- モックでエラーを出すと、設計書のメッセージIDと同じ文が出る
- 設計書とモックの差が2件以上見つかり、設計書v0.2に反映してある

ヒント

- 業務ロジックチェックが思いつかないときは、締めの状態、マスタにあるか、他の人の入力と重なるか、の3つを当てはめます。
- 既存の画面がJSPなら、画面の見た目をHTMLに写す必要はありません。項目の並びと名前が合っていれば、設計の確かめには足ります。
- モックが重くなったら、確かめたいこと3つに関係する項目だけに絞って作り直させます。

安全の注意

- 本番の画面のソースや画面写真には、お客様のデータや社内のURLが写っていることがあります。渡す前に、規程と契約で許されているかを確かめます。
- モックに入れるデータは架空のものにします。本番のデータを数件だけ使う、ということもしません。
- モックをそのまま本番のコードの元にしないでください。製造は、チームの設計標準とフレームワークに沿って行います。

繰り返し方

回	時期	やること
v1	改修の依頼が来た画面	項目定義、入力チェック、モックを作る
v2	お客様との仕様確認の前	モックを見せる相手を決め、その人が何と言いそうかを先に書いてから見せる。言われたことを設計書に戻す
v3	次の改修	v1とv2でAIが仮に決めたことの一覧を並べ、毎回出てくるもの（初期値、並び順など）を、チームの設計書テンプレートの項目として足す

発展課題

モックに、締め前と締め後のような状態を切り替えるボタンを付けさせ、状態ごとに入力できる項目が設計書どおりに変わるかを確かめます。状態による違いは、設計書で一番書き漏れやすいところです。

演習C　チームの設計書の観点別レビュー

所要	[50min]
1回目の時期	チームの設計書がレビューに回ってきたとき
使うもの	自分のチームの設計書1本、その要件一覧、 テンプレート/レビュー記録テンプレート.md、tools/xlsx_to_md.py、プロンプトサンプル集の8と9

目的

研修のD3-1とD3-2を、チームの設計書で回します。観点を6つに分けて1回ずつ頼み、出そろってから自分の指摘を渡して突き合わせ、採用、不採用、保留を決めます。自分の指摘をAIの出力のあとで見せる順番も、研修と同じです。回を重ねると、どの観点でAIの指摘が当たりやすく、どの観点で外れやすいかが、自分のチームの設計書で分かってきます。

手順

1. AIを開かずに、設計書をAIに入れてよいかを確かめる。Excelの設計書はtools/xlsx_to_md.pyでMarkdownにしておく。 [5min]
2. 設計書を読み、要件漏れ、矛盾、曖昧表現、例外系、非機能、項目定義・形式の観点ごとに、自分で指摘を書く。合わせて3件を目標にする。書いた指摘は、手順4までAIに見せない。 [10min]
3. サンプル集の8の1通目を、観点を差し替えながら同じ会話で6回送る。6つの観点をまとめて1回で頼まない。 [15min]
4. 同じ会話でサンプル集の8の2通目を送り、自分の指摘と突き合わせる。採用、不採用、保留を決め、不採用には理由を書く。根拠を開いて確かめていない指摘は保留にする。 [10min]
5. レビュー記録テンプレートにまとめ、tools/md_to_xlsx.pyでExcelにする。時間が残ればサンプル集の9で対応表を作る。 [10min]

自分で考えるところ

採用、不採用、保留は自分で決めます。AIの指摘には、既存の仕組みで済んでいることや、その案件では対象外のことが混ざります。

設計書を書いた人に指摘を返すのは自分です。AIが出した指摘でも、自分が根拠を確かめたものだけを返します。

できあがるものと置き場所

- 作業フォルダの レビュー記録.xlsx。指摘ごとに観点、根拠（シート名と行）、採否、理由がある
- 持ち帰り演習メモ.txtの「演習C v1」。観点ごとのAIの指摘数と採用数

完成の条件

- 6つの観点すべてで、1回ずつAIに頼んだ
- 全指摘に根拠（シート名か節と、行）が付いている
- 自分で見つけた指摘が3件以上ある
- AIの指摘のうち不採用が1件以上あり、理由が書いてある

ヒント

- 計算式の指摘は、1件だけでも自分で手計算します。式の読み違いは、数字を入れてみると一度で分かります。
- 指摘が多すぎるときは、重さの列で「直さないと作れない」だけを先に採否判断します。
- 要件一覧が無い設計書では、要件漏れの観点は飛ばし、その代わりに対応表（サンプル集の9）を作れるかを先に確かめます。

安全の注意

- チームの設計書でも、お客様の業務やデータが書かれていれば、お客様の情報として扱います。規程と契約で許されているかを確かめてから入れます。
- レビュー記録に、AIが出した指摘だと分かる印を残すかどうかは、チームで決めます。書いた人が受け取り方を変えることがあるためです。
- AIの指摘を、根拠を確かめないまま書いた人に返さないでください。

繰り返し方

回	時期	やること
v1	設計書がレビューに回ってきたとき	設計書1本を6つの観点でレビューする
v2	次の設計書	v1の記録から、AIの採用率が高かった観点と低かった観点を見て、低かった観点は自分の指摘を先に厚く書く
v3	チームのレビューの手順を見直すとき	採用率の高かった観点を、チームのレビュー観点表にAIへの依頼の文面つきで足す案を作る

発展課題

採用した指摘ごとに修正案を書かせ、その修正案が別の不備を生んでいないかを、同じ観点でもう一度レビューさせます。修正で新しい矛盾が入るのは、人が直したときにもよく起きます。

 Copyright © Givery, Inc.