プロンプトサンプル集

研修でKINTAIを相手に使った文面を、自分の案件で使える形に直しました。どの文面にも、AIに頼む前に自分で書いておくメモがあります。

1、2、4、8は、要件の候補、食い違い、抜け、レビューの指摘を、自分とAIが別々に探す文面です。2通に分け、1通目で資料だけを渡して出力させ、2通目で自分のメモを渡して突き合わせます。先に見せるとAIがメモに引っぱられ、比べる意味がなくなるからです。研修のD1-2、D1-4、D3-1も、この順番で進めました。

ほかの6本は、書きにくい章や項目定義のように、自分で決めたことを前提としてAIに使わせる文面です。研修と同じく、1通目から一緒に渡してください。

使う前の決まり

- お客様の資料を使う前に、社内規程とお客様との取り決めで、生成AIに入れてよい情報かを確かめます。分からないうちは、社名・人名・金額を置き換えたコピーで試します。
- <>で囲んだ箇所を自分の中身に置き換えます。【】の見出しは消さずに残してください。AIが、どこまでが指示でどこからが資料かを取り違えなくなります。
- ClaudeデスクトップアプリのCodeタブなら、資料は貼らずに @ でファイルを指定できます。ファイル名と行番号で根拠を返してもらえるので、確かめるのが速くなります。
- チャットに貼る場合は、行番号が正しく数えられないことがあります。根拠は行番号ではなく、元の文の引用で返してもらってください。
- 1本ごとに新しい会話を始めます。前の資料が残っていると、別の案件の要件が混ざります。2通目と、8の6回だけは、同じ会話で続けて送ってください。

No	文面	使うとき	送り方	研修の演習
1	議事メモからの要件抽出	ヒアリングや打ち合わせのメモが手元にそろったとき	2通	D1-2
2	メモ同士の矛盾探し	相手の違う議事メモが2本以上あるとき	2通	D1-2
3	要件定義書のドラフト	要件一覧が固まり、テンプレートに流し込むとき	1通	D1-3
4	観点を決めた抜け漏れ探し	要件定義書を相手に見せる前	2通	D1-4
5	画面一覧と画面遷移	要件定義書から画面設計に入るとき	1通	D2-1
6	項目定義と入力チェックの書き分け	1画面の項目定義と入力チェックを書くとき	1通	D2-2
7	設計書から画面のモック	画面設計書が一通り書けたとき	1通	D2-3
8	観点を1つに絞った設計レビュー	他の人が書いた設計書をレビューするとき	6通と1通	D3-1
9	要件と設計の対応表	設計書を出す前の最後の確かめ	1通	D3-2
10	お客様にAIの利用範囲を説明する文面	案件の開始時や、お客様に利用を相談するとき	1通	D3-4

要件を集める

1　議事メモからの要件抽出　研修では D1-2

用途

ヒアリングや打ち合わせのメモから、要件の候補を根拠つきで表にするときに使います。1通目を送る前に、メモを読んで自分で拾った候補を10件ほど書いておきます。1通目ではメモだけを渡し、表が出てから、2通目で自分の候補を渡してください。どちらかにしか無い要件が、ここではっきりします。研修のD1-2と同じ順番です。

全文　1通目　資料だけを渡す

> 議事メモから要件を抜き出して、表にしてください。
>
> 【議事メモ】
> <議事メモを貼る。複数あるときは、メモごとにファイル名を見出しにして分ける>
>
> 【出してほしい表】
> | 要件ID | 要件 | 分類（機能・非機能・制約・課題） | 根拠（ファイル名:行番号） | 根拠の文（メモからそのまま引用） | 発言者 | 要確認 |
>
> 【守ること】
> ・メモに書いていないことを要件にしないでください。業界の一般的な要件で補わないでください。
> ・1つの発言に要件が2つ入っていたら、2行に分けてください。
> ・根拠の文は要約せず、メモの文をそのまま写してください。
> ・メモ同士で言っていることが食い違う箇所は、要確認の列に「食い違い」と書き、相手のメモの行も根拠に並べてください。
> ・「できるだけ早く」「使いやすく」「従来どおり」のような言い方は数字に直さず、要確認の列に「数字の確認が要る」と書いてください。

全文　2通目　1通目の出力を読んでから、同じ会話で送る

> 表を見る前に、私が自分で拾った要件の候補を書いておきました。この候補と、さっきの表を突き合わせてください。表はまだ直さないでください。
>
> 【私が先に書いた要件の候補】
> <1通目を送る前に書いた候補を貼る。1行1件、10件前後>
>
> 【出してほしいもの】
> 次の3つに分けた表にしてください。それぞれ、根拠のメモの行も並べてください。
> 1. 私の候補にだけあるもの
> 2. 表にだけあるもの
> 3. 同じことを指しているが、中身が食い違うもの

差し替える箇所

- 1通目の【議事メモ】に、議事メモの本文。Codeタブなら @ でメモのファイルを指定します
- 2通目の【私が先に書いた要件の候補】に、1通目を送る前に書いた候補。表を読んだあとに思いついたものは入れません
- 分類の列は、自社の要件定義書の分け方に合わせて書き換えて構いません

うまくいかないときの一言

メモに無い要件が混ざったら、「根拠の文を引用できない行を消して、表を出し直してください」と送ります。

出力を使う前の確かめ方

- 根拠の文を3件選び、メモの中を検索して、同じ文が本当にあるかを見ます。見つからない根拠が1件でもあれば、全件を疑います。
- 表にだけある要件は、根拠の行を開いてメモを読み、発言の意味を広げて要件にしていないかを見ます。数が多いときは、自分の候補に無かった理由が気になるものから開きます。
- 要確認が0件なら、あいまいな言い方を拾えていません。メモの中の「できるだけ」「早く」を自分で探して足します。

2　メモ同士の矛盾探し　研修では D1-2

用途

部署や役職の違う相手から聞いたメモが2本以上あるときに使います。納品の期日、承認が何段あるか、帳票をどこへ出すかのような点で、言っていることが食い違っているところを表に出します。どちらが正しいかはAIに決めさせず、誰と誰に何を聞けば決まるかまでを出させてください。自分で気づいた食い違いは1通目を送る前に書いておき、2通目で渡します。

全文　1通目　資料だけを渡す

> 複数の議事メモのあいだで、言っていることが食い違う箇所を探してください。どちらが正しいかの判断と、直し方の案は要りません。
>
> 【議事メモ】
> <メモ1 のファイル名と本文>
> <メモ2 のファイル名と本文>
>
> 【探してほしいもの】
> ・同じことについて、数字、期限、担当、権限が違う
> ・片方が「やりたい」と言い、もう片方が「できない」「規程で禁止」と言っている
> ・同じ言葉を別の意味で使っている。例 納期と出荷日
> ・どちらも、相手が決めると思っていて、誰も決めていないこと
>
> 【出してほしい表】
> | No | 食い違いの中身 | 発言A（ファイル名:行番号と引用） | 発言B（ファイル名:行番号と引用） | 種類 | 誰と誰に何を聞けば決まるか |

【守ること】
・言い方が違うだけで中身が同じものは、表に入れないでください。
・表に入れなかったものを最後に3件まで挙げ、食い違いではないと見た理由を1行ずつ書いてください。

全文　2通目　1通目の出力を読んでから、同じ会話で送る

表を見る前に、私が自分で気づいていた食い違いを書いておきました。さっきの表と突き合わせてください。

【私が気づいていた食い違い】
<1通目を送る前に書いたもの。1行に1つ>

【出してほしいもの】
次の3つに分けてください。
1．私のメモと表の両方にあるもの
2．私のメモにだけあるもの。議事メモに根拠の発言があるかも確かめてください
3．表にだけあるもの

差し替える箇所

- 1通目の【議事メモ】に、相手の違うメモを2本以上。同じ人のメモ2本では食い違いが出にくいです
- 1通目の【探してほしいもの】の例は、自分の案件で揉めそうな言葉に差し替えます
- 2通目の【私が気づいていた食い違い】に、1通目を送る前に書いたもの。1つも気づけなかったときは、2通目は送らずに、表の発言Aと発言Bを自分でメモの中から探します

うまくいかないときの一言

どちらが正しいかをAIが書き始めたら、「判断は要りません。食い違っている2つの発言を並べるだけにしてください」と送ります。

出力を使う前の確かめ方

- 発言Aと発言Bの引用を、両方ともメモの中で検索して見つけます。片方しか見つからない行は捨てます。
- 表に入らなかった3件を読みます。自分には食い違いに見えるものがあれば、自分の判断で表に戻します。
- 聞く相手が「お客様」のような大きい書き方になっていたら、役職か部署まで自分で絞ります。

3　要件定義書のドラフト　研修では D1-3

用途

固まった要件一覧を、自社の要件定義書のテンプレートに流し込むときに使います。ねらいは、書けないところを埋めずに残すことです。推測で埋まった要件定義書は、読んだ人が決まったことだと受け取ります。先に、一番書きにくい章とその理由を自分で1行書いておきます。

全文

> 要件一覧をもとに、要件定義書の下書きを Markdown で書いてください。書けないところを埋めずに残すのが目的です。
>
> 【要件一覧】
> <確定した要件一覧を貼る。要件ID の列を必ず残す>
>
> 【要件定義書の目次】
> <自社や案件のテンプレートの見出しを、階層ごと貼る>
>
> 【私が一番書きにくいと思っている章と、その理由】
> <1〜2行>
>
> 【書き方】
> ・目次の見出しは1つも消さず、順番も変えないでください。
> ・各文の終わりに、根拠の要件ID を付けてください（例 [R-012]）。
> ・要件一覧に無いことを書いた文には、文末に（要確認 誰に何を聞くか）を付けてください。
> ・書けることが1つも無い章は、見出しの下に（要確認 ...）だけを置いてください。空欄にはしないでください。
> ・応答時間、件数、保存期間などの数字は、要件一覧にあるものだけを使ってください。
>
> 【最後に付けてほしい表】
> | 章 | 要確認の中身 | 聞く相手 |

差し替える箇所

- 【要件一覧】に、確定した一覧。要件IDの無い一覧なら、先に番号を振ります
- 【要件定義書の目次】に、自社のテンプレートの見出し。Wordのテンプレートなら見出しだけを写します
- Wordにしたいときは、研修で使った配布フォルダのtools/md_to_docx.pyで変換できます

うまくいかないときの一言

（要確認）がほとんど付かずに全部埋まったら、「要件IDを付けられない文を全部挙げてください」と送ってから、その文を要確認に変えさせます。

出力を使う前の確かめ方

- 非機能要件の章を最初に読み、応答3秒や同時接続100のような、要件一覧に無い数字が無いかを見ます。あれば消すか要確認にします。
- （要確認）の数を章ごとに数えます。1つの章に集中していたら、その章はAIではなく自分が書き直す章です。
- 書きにくいと書いた章が、本当に要確認として残っているかを見ます。それらしい文で埋まっていたら、根拠の要件IDをたどります。

4　観点を決めた抜け漏れ探し　研修では D1-4

用途

要件定義書をお客様やレビュー担当に見せる前に使う文面です。「抜け漏れを探して」とだけ頼むと、どの案件にも当てはまる一般論が並びます。観点を4つに絞り、手元のデータで起きていることを一緒に渡すと、その案件に固有の抜けが出てきます。1通目を送る前に、自分で見つけた抜けを1件以上書いておき、2通目で渡してください。研修のD1-4と同じ順番です。

全文　1通目　資料だけを渡す

```
要件定義書の抜け漏れを、観点を決めて探してください。

【要件定義書】
<要件定義書を貼る。Code タブなら @ で指定する>

【手元のデータで起きていること】
<実データの例外の例。例 同じ顧客の二重登録、月をまたぐ契約の変更、上限を超える注文数。社名、人名、金額は置き換える>

【観点】
次の4つを1つずつ見てください。
1. 例外 正常の流れから外れたとき。取消、差し戻し、期限切れ、0件、上限を超える入力
2. 運用 誰がいつ何をするか。締め、月次、年度替わり、担当の交代、問い合わせ
3. 権限 誰が見られて誰が直せるか。直した履歴を残すか
4. データ移行 今のデータを新しい形にどう移すか。移せないものは何か

【出してほしい表】
| 観点 | 抜けている可能性のあること | そう考えた根拠（要件定義書の章か、データの例） | 確認する相手 | 聞くこと |

【守ること】
・観点ごとに3件までにしてください。
・根拠が示せないものは表に入れず、最後に「根拠なしの候補」として分けてください。
・要件定義書はまだ書き換えないでください。
```

全文　2通目　1通目の出力を読んでから、同じ会話で送る

> 私が先に見つけた抜けを渡すので、それ以外を探してください。この抜けは、さっきの表を見る前に書いたものです。
>
> 【私が見つけた抜け】
> <1件以上。何を見て気づいたかも書く>
>
> 【出してほしいもの】
> さっきの表を、次の3つに分けてください。
> 1．私の抜けと表の両方にあるもの
> 2．私の抜けにだけあるもの
> 3．表にだけあるもの
> 私の抜けを読んで、表に無い抜けに新しく気づいたら、根拠を付けて最後に足してください。

差し替える箇所

- 1通目の【手元のデータで起きていること】に、実データを見て分かった例外。データそのものは貼らず、起きていることを文で書きます。自分で見つけた抜けは、ここには書かずに2通目で渡します
- 1通目の【観点】は、案件の種類に合わせて1つ入れ替えて構いません。帳票が多い案件なら「帳票」、外部連携が多い案件なら「連携」です
- 2通目の【私が見つけた抜け】に、1通目を送る前に書いた抜けと、気づいたきっかけ

うまくいかないときの一言

どの案件にも当てはまる一般論ばかり出たら、「要件定義書の章かデータの例を根拠に書けるものだけに絞ってください」と送ります。

出力を使う前の確かめ方

- 2通目の3つの分け方を読み、自分にだけある抜けを数えます。自分にだけあるものは、データや現場を見たから拾えた抜けです。
- 根拠なしの候補は、すぐに要件に足しません。確認事項リストに入れ、相手に聞いてから決めます。
- 確認する相手と聞くことを、そのまま質問票に写せるかを見ます。「仕様を確認」のような聞き方は、相手が答えられないので書き直します。

画面を設計する

5　画面一覧と画面遷移　研修では D2-1

用途

要件定義書から画面設計に入るときに使います。改修の案件では、新しく作る画面より、影響を受ける既存の画面を拾い漏らすほうが後で困ります。今の画面を影響なしのものまで全部渡し、先に自分の見立てを書いておいてください。

全文

> 要件定義書から、改修で手を入れる画面の一覧と画面遷移を作ってください。
>
> 【要件定義書】
> <要件定義書を貼る。Code タブなら @ で指定する>
>
> 【今の画面の一覧】
> <画面ID、画面名、使う人を1行ずつ。画面の HTML があれば @ で渡す>
>
> 【私の見立て】
> <変わると思う画面と、その理由を1行ずつ>
>
> 【出してほしいもの】
> 1. 画面一覧の表
> | 画面ID | 画面名 | 区分（改修・新設・影響あり・影響なし） | 対応する要件ID | 使う人 | 区分の理由 |
> 今の画面は、影響なしのものも含めて全部載せてください。
> 2. 画面遷移を Mermaid の flowchart で。新設の画面は名前の頭に「新」を付けてください。
> 3. 次の2つの表
> ・要件ID が1つも付かない画面
> ・どの画面にも付かない要件ID
> 4. 私の見立てと区分が違う画面について、両方の理由を並べた表

差し替える箇所

- 【今の画面の一覧】に、今の画面をすべて。影響なしの画面を省くと、影響ありの判断ができません
- 【私の見立て】に、自分で考えた区分と理由
- 使う人の列は、自社の権限の呼び方（一般、承認者、管理者など）に合わせます

うまくいかないときの一言

今の画面の一部しか表に出てこなかったら、「今の画面一覧の行数と、表の行数を合わせてください」と送ります。

出力を使う前の確かめ方

- どの画面にも付かない要件IDを1件ずつ読みます。夜間の処理のように画面が要らない要件なのか、画面の拾い漏れなのかを自分で決めます。
- 影響ありの画面は、データの取込や出力の画面を見落としやすいです。CSVの取込、帳票、一覧のダウンロードがある画面を自分で見直します。
- 遷移図は、図にして開いて矢印の抜けを見ます。Codeタブなら「このMermaidをHTMLにして開けるようにして」と頼めます。

6　項目定義と入力チェックの書き分け　研修では D2-2

用途

1つの画面の項目定義と入力チェックを書くときに使います。入力チェックは、1項目で決まる単項目チェック、同じ画面の項目を見比べる相関チェック、画面の外のデータや状態を見る業務ロジックチェックの3つに分けて書きます。分けておかないと、製造の段階でどこに実装するかが決まりません。自社の設計標準に書き分けがあれば、そちらを貼って差し替えます。

全文

```
画面の項目定義と入力チェックを、3種類に書き分けて作ってください。

【対象の画面と改修の中身】
<画面名と、何を足すか、何を変えるか>

【関係する要件】
<要件ID と本文>

【私が先に書いた項目定義】
<5項目ほど。項目名、型、桁、必須、初期値、誰が入れるか>

【チェックの書き分け】
・単項目チェック 1つの項目だけで判定できるもの。必須、型、桁、範囲、形式
・相関チェック 同じ画面の複数の項目を見比べて判定するもの。開始日と終了日、明細の合計と合計欄
・業務ロジックチェック 画面の外のデータや状態を見ないと判定できないもの。締めの状態、マスタにあるか、他の人の入力との重複

【出してほしい表】
表1 項目定義
| 項目ID | 項目名 | 型 | 桁 | 必須 | 初期値 | 入力者 | 根拠の要件ID |
表2 入力チェック
| チェックID | 種類（単項目・相関・業務ロジック） | 対象の項目ID | 条件 | メッセージID | メッセージの文 |
表3 判断に迷ったもの
| チェック | 迷った理由 | どちらにした場合に何が変わるか |

【守ること】
・私の項目定義と違う値にしたところは、表1の下に理由を書いてください。
```

> ・1つのチェックに条件を2つ入れないでください。
> ・どの種類か決めきれないチェックは、表2に入れず表3に入れてください。

差し替える箇所

- 【対象の画面と改修の中身】に、画面名と改修の中身。画面は1つに絞ります
- 【私が先に書いた項目定義】に、自分で書いた5項目。AIの表と比べる元になります
- 【チェックの書き分け】は、自社の設計標準に書き分けがあれば、その文に差し替えます

うまくいかないときの一言

3種類の区別があいまいなまま表が出たら、「業務ロジックチェックだけを先に挙げ、なぜ画面の中だけでは判定できないかを1行ずつ書いてください」と送ります。

出力を使う前の確かめ方

- 表3を先に読みます。迷ったチェックの種類は、AIではなく自分が決めます。決めた理由は、設計書の備考に残してください。
- メッセージIDが重なっていないか、同じIDに違う文が付いていないかを見ます。
- 業務ロジックチェックが0件なら、締めの状態やマスタとの照合を見落としています。画面の外で決まることを自分で1つ足します。

7　設計書から画面のモック　研修では D2-3

用途

画面設計書が一通り書けたところで、画面として成り立つかを確かめるために使います。モックを作ると、初期値、並び順、エラーの出る位置のように、設計書に書いていないのに決めないと画面にならないことが見えてきます。それを設計書に戻すのが目的です。モックで確かめたいことを、先に3つ書いておきます。

全文

> 画面設計書をもとに、画面のモックを HTML で作ってください。設計書の抜けを見つけるためのモックです。
>
> 【画面設計書】
> <項目定義、入力チェック、メッセージの表を貼る。Code タブなら @ で指定する>
>
> 【モックで確かめたいこと】
> <自分で決めた3つ。例 予定と実績を同じ行で見比べられるか、エラーがどこに出るか、締めたあとに入力できないか>
>
> 【作り方】
> ・HTML と JavaScript だけで、1つのファイルにしてください。外部のライブラリ、CDN、Web サーバは使わないでください。ダブルクリックで開けるようにします。
> ・入力欄には、設計書の項目ID を id 属性に付け、画面上にも小さく表示してください。
> ・入力チェックは、設計書のチェックID とメッセージID のとおりに出してください。
> ・データは画面の中に数件だけ埋め込み、実在の会社名や人名は使わないでください。
> ・設計書に書いていないのに決めないと作れなかったこと（初期値、並び順、エラーの出る位置、ボタンの名前など）は、画面の一番下に「設計書に無いので仮に決めたこと」として一覧で出してください。

差し替える箇所

- 【画面設計書】に、設計書の表。Excelなら、研修の配布フォルダの tools/xlsx_to_md.pyでMarkdownにしてから渡します
- 【モックで確かめたいこと】に、自分で決めた3つ
- チャットで使うときは、出てきたHTMLをメモ帳に貼り、拡張子を.htmlにして保存してから開きます

うまくいかないときの一言

外部のライブラリを読み込むモックが出てきたら、「外部の読み込みを全部外し、1ファイルで動くように書き直してください」と送ります。

出力を使う前の確かめ方

- 確かめたいこと3つを、ブラウザで1つずつ操作して確かめます。3つとも設計書の記述で説明できるかを見ます。
- 「設計書に無いので仮に決めたこと」を1件ずつ読み、設計書に書き足すか、仮のままにしてお客様に聞くかを決めます。
- エラーを出してみて、メッセージIDと文が設計書と一致しているかを見ます。モックが勝手に文を変えていたら、どちらが正しいかを決めます。

設計を確かめる

8　観点を1つに絞った設計レビュー　研修では D3-1

用途

設計書のレビューで使います。「レビューして」とだけ頼むと、表記の揺れや体裁の指摘が並び、計算式の誤りや例外の抜けのような深い指摘が埋もれます。1通目は観点を1つずつ変えて、観点の数だけ6回送ってください。自分の指摘は6回を送る前に書いておき、6回分の表が出そろってから2通目で渡します。研修のD3-1と同じ順番です。

全文　1通目　観点を差し替えて、同じ会話で6回送る

> 設計書をレビューしてください。今回見る観点は1つだけです。ほかの観点で気づいたことは書かないでください。
>
> 【今回の観点】
> <次の6つから1つだけ残す 要件漏れ、矛盾、曖昧表現、例外系、非機能、項目定義・形式>
>
> 【観点の中身】
> ・要件漏れ 要件にあるのに、設計書のどこにも出てこないもの
> ・矛盾 シート同士、節同士で食い違う記述。計算式と説明の食い違いも含む
> ・曖昧表現 読む人によって意味が変わる言葉。柔軟に、十分な、必要に応じて など
> ・例外系 検索結果が0件、通信の途中切れ、二重送信、取消や差し戻し、月末や年度替わりの日付の境目
> ・非機能 性能の数字、権限、ログ
> ・項目定義・形式 項目の型、桁、必須と、ID やメッセージの書き方がシートの間でそろっているか
>
> 【設計書】
> <設計書を貼る。Excel なら Markdown にしてから。Code タブなら @ で指定する>
>
> 【要件一覧】
> <観点が要件漏れのときだけ貼る。ほかの観点では消す>
>
> 【出してほしい表】
> | No | 指摘 | 根拠（シート名か節と、行） | 根拠の記述（設計書からそのまま引用） | 重さ（直さないと作れない・直したほうがよい・好み） | 直し方の方向 |
>
> 【守ること】
> ・根拠の記述を引用できない指摘は出さないでください。

全文　2通目　6回分の表が出そろってから、同じ会話で送る

> 6つの観点の指摘を、1つの表にまとめてください。そのうえで、私が表を見る前に書いた指摘と突き合わせてください。
>
> 【私が先に見つけた指摘】
> <6回を送る前に書いた指摘。観点、場所、中身を1行に1件>
>
> 【出してほしい表】
> | No | 観点 | 指摘 | 根拠（シート名か節と、行） | 見つけた人（私・AI・両方） |
>
> 【守ること】

> ・私の指摘のうち表に入っていないものについて、見落としたのか、指摘として弱いと見たのかを1行ずつ書いてください。

差し替える箇所

- 1通目の【今回の観点】に、6つのうち1つ。観点だけを差し替えて、同じ会話で6回送ります。2回目からは、【設計書】と【要件一覧】を「1回目と同じ」と書き換えて構いません
- 2通目の【私が先に見つけた指摘】に、6回を送る前に観点ごとに書いた指摘
- 観点の中身は、自社のレビュー観点表があれば差し替えます

うまくいかないときの一言

ほかの観点の指摘が混ざったら、「今回の観点に当てはまらない行を消して、表を出し直してください」と送ります。

出力を使う前の確かめ方

- 根拠の記述を設計書の中で検索し、書いてあるとおりかを見ます。計算式の指摘は、自分で1件だけ手計算します。
- 指摘ごとに採用、不採用、保留を決めます。根拠を開いて確かめていない指摘は、保留にしてください。不採用にしたものは理由を1行残すと、次に同じ指摘が出たときの判断が速くなります。
- パスワードの桁数の決まりが無い、のように、共通の仕組みで決まっていることを指摘していないかを見ます。対象外の指摘は不採用にします。

9　要件と設計の対応表 研修では D3-2

用途

設計書を出す前に、要件が設計のどこで実現されるかを表にするときに使います。見つけたいのは、対応先の無い要件と、根拠になる要件の無い設計項目の2つです。重要な要件を3つ選び、自分で設計書をたどって対応先を書いておきます。

全文

> 要件と設計の対応表を作ってください。対応先の無い要件と、根拠の無い設計項目を見つけるのが目的です。
>
> 【要件一覧】
> <要件ID と本文>
>
> 【設計書】
> <画面設計書、項目定義など。複数あるときはファイル名を見出しにして分ける>
>
> 【私が先にたどった対応】
> <重要な要件3つと、自分で見つけた対応先>
>
> 【出してほしいもの】
> 1. 要件から見た対応表
> | 要件ID | 要件 | 対応先（設計書名、シート、項目ID） | 対応の程度（そのまま実現・一部・対応なし） | メモ |

> 全部の要件を載せてください。対応先が無い要件は「対応なし」と書き、空欄にしないでください。
> 2. 設計から見た逆引き
> | 設計書と項目ID | 根拠になる要件ID（無ければ「根拠なし」） |
> 3. 私の3件と食い違ったものだけの表
>
> 【守ること】
> ・要件と設計で言葉が違っても、中身が同じなら対応ありにしてください。そう判断した行は、メモの列に言い換えを書いてください。

差し替える箇所

- 【要件一覧】に、要件IDつきの一覧。IDが無いと対応表が作れません
- 【設計書】に、対応を見たい設計書をすべて
- 【私が先にたどった対応】に、自分でたどった3件

うまくいかないときの一言

要件の一部しか表に出なかったら、「要件一覧の件数と表の行数を数えて、足りない要件を追加してください」と送ります。

出力を使う前の確かめ方

- 対応なしの要件の数を数え、1件ずつ扱いを決めます。次の版に回す、対象外にする、設計に足す、のどれかです。
- 根拠なしの設計項目は、要件の書き漏れか、設計の勝手な追加かのどちらかです。要件の書き漏れなら要件定義書に戻します。
- 言い換えで対応ありにした行を読みます。言葉が似ているだけで中身が違うものが混ざりやすいところです。

お客様に説明する

10　お客様にAIの利用範囲を説明する文面　研修では D3-4

用途

案件で生成AIを使いたいとき、お客様に利用の範囲を説明する文面の下書きに使います。何に使い、何に使わず、何を入れないかは自分とお客様が決めることで、AIに決めさせるものではありません。自分で決めた中身を箇条書きにしてから渡し、AIには言葉を整えてもらいます。

全文

> お客様に、この案件で生成AIをどう使うかを説明する文面の下書きを作ってください。中身は私が決めたので、言葉を整えてください。
>
> 【私が決めた利用の範囲】
> ・使う作業 <例 議事メモから要件の候補を出す、設計書のレビュー観点を出す>
> ・使わない作業 <例 お客様への回答文の送付、最終の判断>
> ・AI に入れない情報 <例 お客様の実データ、個人情報、公開前の料金>
> ・入れる前に置き換える情報 <例 会社名、担当者名>
> ・最後に確かめる人 <役割で書く。例 PL がレビューしてから提出する>
>
> 【使うサービス】
> <サービス名と契約の種類。入力が学習に使われない設定かどうか、自社で確かめた内容>
>
> 【お客様の状況】
> <AI の利用についてお客様から言われていること。無ければ「特になし」>
>
> 【書き方】
> ・A4 で1枚に収まる長さにしてください。
> ・見出しは、使う作業、使わない作業、入れない情報、確認の手順、問い合わせ先の5つにしてください。
> ・私が書いていない安全対策や認証を足さないでください。足したほうがよいと思うものは、文面の外に「足すなら確かめが要ること」として並べてください。
> ・「安全です」「情報は漏れません」のような保証の言い方は使わないでください。
> ・お客様に決めてもらいたいことを、最後に質問の形で3つまで置いてください。

差し替える箇所

- 【私が決めた利用の範囲】の5行すべて。ここが空のままだと、AIがそれらしい範囲を作ってしまいます
- 【使うサービス】に、実際に使うサービスと契約の種類。学習に使われるかどうかは、契約と設定で変わるので自分で確かめて書きます
- 【お客様の状況】に、お客様から言われていること

うまくいかないときの一言

お客様が安心しそうな言葉ばかりの文になったら、「保証する言い方を消し、決めたことだけを書いた文にしてください」と送ります。

出力を使う前の確かめ方

- 暗号化、認証、ログの保存期間のように、自分が書いていない約束が文面に入っていないかを見ます。入っていたら消すか、社内で確かめてから残します。
- 入れない情報の一覧が、社内規程の禁止事項と食い違っていないかを見ます。
- お客様に出す前に、上長か情報セキュリティの担当に読んでもらいます。文面の責任は出す側にあります。

 Copyright © Givery, Inc.