

Day1

上流工程 要件定義・設計 生成AI活用研修

ヒアリングメモから要件定義書へ

株式会社インフォメーション・ディベロプメント様

2026年10月8日（木） オンライン



本資料の二次転載禁止について

本資料の著作権は株式会社ギブリーに帰属します。受講者の学習を目的として提供するものです。

本資料の全部または一部を、当社の許諾なく複製、転載、配布、公衆送信することを禁じます。社内での共有をご希望の場合は、事務局を通じてご相談ください。

個人での録画・録音はご遠慮ください。

講師紹介



人的資本インテリジェンス部門講師・生成AI
エバンジェリスト

安田 光喜

Mitsuki Yasuda

公立はこだて未来大学 大学院（メディアデザイン領域）を卒業し、街づくり会社での複合商業ビルの立ち上げにおける情報インフラ整備やデザイン全般業務の責任として関与。その後デザイン事務所を立ち上げ、飲食店を中心としたデザイン業務や美術館展示のアートコンテンツ開発などを行う。また、地域ブランディングにも注力し、市主催の企画運営業務を行なった。2018年10月『小中学生向けのプログラミング教育』に注目し、函館市で初めての小中学生向けプログラミングスクール

『D-SCHOOL北海道』を立ち上げて運営。その後北海道のドラッグストア『サッポロドラッグストアー』からスクールの買収を受け、教育を専門とするグループ会社『株式会社シーラクス』にて技術開発責任者として運営。スクール設計・運営、20以上の自治体連携（イベントや出張教室運営）、大人向けプログラミングスクールメンターなどさまざまな『次世代IT教育』に取り組む。

Dify公式資料の日本語翻訳や、SecondMeアンバサダーなど生成AI最新トレンドに注力。

得意領域

- ・生成AIをはじめとした世界最先端トレンド
- ・生成AIを用いた新規事業の開発、運営、補助
- ・アプリケーション開発/自動化プログラム環境構築
- ・教育（大学講師、研修、スクール運営など）

実績

- ・生成AI研修
- ・大手企業、外資、教育機関(小中高大)でのITトレンド講演
- ・自治体と連携した地域教育実績
- ・上場企業社員対象のリスキリング支援

メッセージ

みなさんがこれから生成AIをパートナーに新しい時代を生きていけるように、3日間全力でサポートしていきます！なにかご不明点などありましたら、講師陣になんでも気軽にご質問ください。

[#ClaudeCode](#) [#Codex](#) [#Cursor](#) [#Antigravity](#)

[#全国统一生成AI活用技能試験S1](#)

[#生成AIパスポート](#) [#G検定](#)

本日のゴール

ヒアリングメモから要件を根拠付きで抜き出し 要件定義書のドラフトを Word で作る

メモにない要件を AI に足させない頼み方を、4つの演習で身につけます。

今日できあがるもの（成果物/day1/）

ファイル	中身
要件一覧.md	要件が25件以上。全件に根拠（ファイル名と行番号）
要件定義書_v0.2.docx	テンプレートの見出しが全部あり、空欄が無い
確認事項.md	相手・聞くこと・なぜ、が5件以上

3回の見取り図

Day1 の要件定義書が Day2 の画面設計の入力になります

Day1 10月8日（木）

ヒアリングメモから
要件定義書へ

作るもの

要件一覧

要件定義書（Word）



Day2 10月15日（木）

画面設計書とモック

作るもの

画面設計書（Excel）

画面のモック（HTML）



Day3 10月22日（木）

設計レビューと
収支ダッシュボード

作るもの

レビュー記録・対応表（Excel）

収支ダッシュボード（HTML）

前の回が終わらなかった方には、次の回の最初に講師が合流版を配ります。そこから同じ位置で始められます。

本日の進め方

説明は短くし、時間の大半は手を動かします

順	コマ	時間	作るもの
1	導入	[15min]	(配布フォルダを開く)
2	D1-1 文脈の有無による出力の違い	[25min]	D1-1_比較.md
3	D1-2 ヒアリングメモからの要件抽出	[40min]	要件一覧.md
4	休憩	[10min]	
5	D1-3 要件定義書のドラフト作成	[35min]	要件定義書_v0.1.docx
6	D1-4 抜け漏れ探し	[25min]	要件定義書_v0.2.docx
7	まとめ	[15min]	理解度チェック
8	質疑・アンケート	[15min]	

題材 KINTAI

3回とも同じ会社の同じシステムを扱います

項目	内容
会社	サンプル情報サービス株式会社（架空）。独立系の Sler、社員約300名
部署	第1開発部（金融）、第2開発部（公共・運輸）、管理部
システム	勤怠・工数管理システム KINTAI。全社員が使う唯一の Web システム
稼働	2014年から
作り	JSP + Spring MVC + MyBatis。TERASOLUNA 5.x ベース

Java のコードは読みません

3回とも、扱うのは文書と画面です。作りの情報は、設計の前提として知っておくだけで足ります。

依頼内容は3行だけで、詳細はヒアリングで決まります

No	依頼内容	対象画面
1	工数入力画面に予定工数を入力できるようにする	KIN030 工数入力
2	1日の作業時間の合計が24時間を超えたら入力エラーにする	KIN030 工数入力
3	プロジェクトの収支を見る画面を新設する	PJ010 収支ダッシュボード

希望時期

2027年4月

備考

詳細は各部のヒアリングで確認すること

ヒアリングの相手

第1開発部 田中課長
第2開発部 鈴木主任
管理部 佐藤さん

起票は情報システム課（2026年8月27日）。依頼書は 資料/day1/ に入っています。

現行画面の一覧

改修するのは **KIN030**、新設するのは **PJ010** です

画面ID	画面名	使う人	中身
KIN010	メニュー	全員	各画面へのリンク。ログイン中の氏名と締め状態
KIN020	日次勤怠入力	全員	勤務区分、出勤、退勤、休憩（分）、実働（自動計算）
KIN030	工数入力	全員	日付ごとに PJ・作業区分・作業時間・備考を行で入れる
KIN040	交通費申請	全員	日付、区間、金額、目的
KIN050	月次締め・CSV	PM・管理部	締め状態、CSV 出力、パートナー実績の CSV 取込
PJ010	収支ダッシュボード	（新設）	改修依頼の3番目。Day3 で作ります

KIN010～KIN050 は配布フォルダの 現行画面/ に、ブラウザで開けるモックとして入っています。

KIN030 を開いて、今の入力チェックを確かめてください

[3min]

手順

- 1 現行画面/index.html をダブルクリックします
- 2 メニューの「工数入力（KIN030）」を押します
- 3 「行を追加」を押し、日付と PJ・作業区分を選びます
- 4 作業時間に 25 と入れて「保存」を押します
- 5 画面の上部に出るエラーを読みます

KIN030 の今の入力チェック

No チェック

- 1 作業時間は必須
- 2 0.25時間単位
- 3 1行あたり24時間以内

KINTAI 勤怠・工数管理システム 第1開発部 渡辺さん ログイン中 9月分 未締め(10/2 17:00 ロック予定) 画面ID: KIN030

メニュー > 工数入力

・2026-09-30 の行: 作業時間は1行24時間以内にしてください

工数入力

日付ごとに、プロジェクト・作業区分・作業時間・備考を1行で入力してください。作業時間は0.25時間単位、1行24時間以内です。

日付	プロジェクト	作業区分	作業時間	備考	
09/14/2026 ☐	PJ-COMMON-01 社内業務(会議・教育)	会議	1	部会	削除
09/14/2026 ☐	PJ-2026-011 信託財産管理システム 保守	保守対応	1.75	月次バッチの結果確認	削除
09/14/2026 ☐	PJ-2026-014 企業保険 契約管理 改修	設計	5.25	契約照会画面の詳細設計	削除
09/15/2026 ☐	PJ-2026-011 信託財産管理システム 保守	保守対応	4		削除
09/15/2026 ☐	PJ-2026-014 企業保険 契約管理 改修	製造	4		削除
09/30/2026 ☐	PJ-2026-011 信託財産管理システム 保守	保守対応	25		削除
合計			41.00	実働との比較はしていません(現行仕様)	

サンプル情報サービス株式会社(架空)の現行画面モックです

25 を入れて保存し、1行24時間以内のエラーが出た状態

配布フォルダの中身

資料は読むだけ、作ったものは**成果物/day1** に置きます

idup-handson/	
├─ CLAUDE.md	Claude が最初に読む約束ごと（AGENTS.md を読み込む）
├─ 資料/day1/	改修依頼書とヒアリングメモ3本
├─ 資料/データ/	勤怠工数CSV_202609.csv ほか
├─ 現行画面/	KINTAI の現行画面のモック
├─ テンプレート/	要件定義書テンプレート.md ほか
├─ tools/	md_to_docx.py ほかの変換ツール
└─ 成果物/day1/	今日作るものはすべてここ

架空の題材です

会社・人物・金額はすべて架空です。研修中は、配布した資料だけを AI に渡してください。



最初に**配布フォルダ**を Claude に開かせます

順	操作
1	Claude デスクトップアプリを起動し、Code タブを選びます
2	環境は「ローカル」を選びます
3	「フォルダを選択」を押し、デスクトップの idup-handson を選びます
4	画面の上部にフォルダ名が出れば完了です

事前準備で一度開いたフォルダと同じものです。別の場所に展開した方は、その場所を選んでください。

※ Claude のロゴは Anthropic の商標です。

入力欄で使う操作

今日使う操作はこの4つです

操作	使いどころ
@ を打つ	ファイルを指定できます。依頼書やメモを渡すときに使います
モード名を押す	入力欄の下の表示で切り替えます（自動・手動・編集を受け入れる・プラン）。今日は「手動」で始め、D1-3 から「編集を受け入れる」にします
差分の表示	ファイルを変えると +12 -1 のような表示が出ます。押すと中身を確認られます
許可	コマンドを実行する前に、許可を求められることがあります

許可を求められたら、何を実行するのかを読んでから押してください。

D1-1

文脈の有無による出力の違い

[25min]

作るもの

成果物/day1/D1-1_比較.md

同じ依頼の3通りの送り方

渡すものと禁止事項だけを変えて、出力を比べます

送り方

出力で見るところ

① 依頼文だけ

「ファイルは読まずに」と断って、依頼文だけで送る

→ メモにない要件がいくつあるか

② + 依頼書とメモ3本

①と同じ依頼に、@ で4つのファイルを付けて送る

→ メモの言葉が使われているか

③ + 禁止と根拠

②に「メモにないことは書かない」「根拠の行番号」を足す

→ 根拠の行に本当に書いてあるか

AI は、渡されていない会社の事情を知りません。渡さなければ、一般的な勤怠システムの知識で埋めます。

CLAUDE.md の役目

配布フォルダの約束ごとは**会話の最初**に読み込まれます

CLAUDE.md から読み込まれる AGENTS.md の約束ごと（抜粋）

資料/ の中は書き換えない。作ったものは 成果物/dayN/ に保存する

ヒアリングメモにないことを要件として足さない。推測には（要確認）を付ける

要件や指摘には根拠を付ける。根拠はファイル名と行番号

Office 形式の出力は tools/ の変換ツールを使う

Claude reads them at the start of every session.

Claude treats them as context, not enforced configuration.

強制の設定ではないので、守らせたいことは依頼の文面でももう一度言います。

出典 Anthropic 「How Claude remembers your project」 Claude Code Docs（2026年9月24日閲覧）

<https://code.claude.com/docs/en/memory>

禁じる指示と根拠の求め方

書いてよい範囲を決め、1件ずつ**出どころ**を付けさせます

禁じる

メモにないことは要件として書かない
推測で埋めた箇所には（要確認）を付ける

書いてよい範囲が決まります

根拠を求める

各要件に根拠のファイル名と行番号を付ける
例 ヒアリングメモ_01:L23

1件ずつ、自分で開いて確かめられます

根拠から原文へたどる例

REQ-001
24時間超で入力エラー



改修依頼書:L27



27 | 2 | 1日の作業時間の合計が24時間を
超えたら入力エラーにする | KIN030 |

行を開いて同じことが書いてあれば、その要件は採れます。

Anthropic も出どころを示させる方法を勧めています

手（原文の見出し）	中身	今日の演習での形
Allow Claude to say "I don't know"	分からないと言ってよいと明示する	（要確認）を付けさせる
Use direct quotes for factual grounding	先に原文をそのまま抜き出させる	根拠の行を引用させる
Verify with citations	主張ごとに出どころを示させ、 見つからなければ取り下げさせる	根拠のファイル名と行番号

同じページに、これらの手を使っても誤りはゼロにならないと書かれています。だから根拠の行は自分で開きます。

出典 Anthropic 「Reduce hallucinations」 Claude Docs（2026年9月24日閲覧）

<https://platform.claude.com/docs/en/test-and-evaluate/strengthen-guardrails/reduce-hallucinations>

演習 D1-1 のねらいと目安

ねらい

同じ依頼でも、渡すものと禁止事項で出力がどう変わるかを自分の目で見ます。以後の全演習で使う指示の型を決めます。

できあがりの目安

- ☐ 3回分の出力が D1-1_比較.md に並んでいる
- ☐ 違いを自分の言葉で3点書いてある
- ☐ 指示の型を自分の言葉で1行にしてある

使うファイル・作るファイル

区分	ファイル
使う	資料/day1/改修依頼書_KINTAI-2026-03.md
使う	資料/day1/ヒアリングメモ_01～03（3本）
作る	成果物/day1/D1-1_比較.md

最初の5分は AI を開かずに書きます

■ 自分の手で ■ AIと一緒に

STEP1
[5min]

STEP2
[10min]

STEP3
[5min]

STEP4
[5min]

■ STEP1 自分で考える

改修依頼書を読み、AI に渡すものと禁じるものを2行ずつ書きます

■ STEP2 AI と詰める

①②③を同じ会話で順に送り、比較ファイルに並べさせます

■ STEP3 確かめる

①のメモにない要件を数え、③の根拠の行を2つ開きます

■ STEP4 自分で判定する

違いを3点書き、指示の型を自分の言葉で1行にします

③だけは自分で書いた禁止事項を差し込みます

① 何も渡さない

ファイルは読まずに、あなたの知識だけで答えてください。
勤怠システムの工数入力に予定工数を足す要件を書いて。
成果物/day1/D1-1_比較.md を新しく作り、
「## ① 何も渡さない」の見出しの下に保存してください。

② 依頼書とメモを渡す

〔@ で依頼書とメモ3本を指定〕
この4つを読んで、勤怠システムの工数入力に
予定工数を足す要件を書いて。
D1-1_比較.md の末尾に見出し「## ② 資料を渡す」
を足して、その下に保存してください。

③ 禁止事項と根拠を足す

〔②と同じ4つを指定〕
この4つを読んで、勤怠システムの工数入力に
予定工数を足す要件を書いて。
- メモにないことは書かない
- 各要件に、根拠のファイル名と行番号を付ける
 (例 ヒアリングメモ_01:L23)
- 次の2つも守る
<ここにSTEP1で書いた「禁じること」2行を貼る>
D1-1_比較.md の末尾に見出し「## ③ 禁止と
根拠を足す」を足して、その下に保存してください。

①②③は同じ会話で順に送ります。①で比較ファイルを作り、②と③はその末尾に足していきます。

数えて開き、自分の言葉で書きます

見方	中身
数える	①の出力のうち、依頼書にもメモにも無い要件の数
開く	③の根拠から2つ選び、その行に本当に同じことが書いてあるか
書く	違いを3点と、指示の型「渡す・禁じる・根拠を求める」を自分の言葉で1行

よくある落とし穴

③の出力が整って見えるので、根拠の行を開かずに次へ進んでしまうこと。行番号が1～2行ずれているだけでも、別の発言を指していることがあります。

D1-1 早く終わった方へ

追加と考察

③の出力で、根拠の行番号がずれている要件はありましたか。
なぜずれるのかを考えて、比較ファイルに1行書き足します。

発展課題

③の依頼に「分からないところは質問を返して」と足して送り、
AI が返した質問を読みます。自分ならどれを聞くかを選びます。

時間が余った方だけ取り組みます。答え合わせは講師が口頭で行います。

D1-2

ヒアリングメモからの要件抽出

[40min]

作るもの

成果物/day1/自分の要件候補.md、要件一覧.md

要件の4分類

依頼書の中には**非機能**の要件が1件也没有せん

分類	意味	改修依頼書の中の例
機能	システムがすること	1日の作業時間の合計が24時間を超えたら入力エラー（L27）
非機能	速さ・権限・履歴・運用などの質	依頼書には無い。メモから探す
制約	決まっていて動かさせない条件	希望時期 2027年4月（L32）
課題	今困っていること。まだ要件の形ではない	月末の集計に時間がかかっている（L20）

課題は、そのままでは作れません。誰の、どの作業が、どう変わればよいかまで書けたら要件にします。

要件一覧の列

根拠と要確認の列は空けないのが決まりです

要件ID	要件	分類	根拠	発言者	要確認
REQ-001	1日の作業時間の合計が24時間超で入力エラー	機能	改修依頼書:L27	情報システム課 森	—
REQ-002	稼働は2027年4月	制約	改修依頼書:L32	情報システム課 森	—

列	書き方
根拠	ファイル名と行番号。1つの要件に根拠が2つあれば両方書く
発言者	その行で話している人。聞き手の発言は根拠にしない
要確認	食い違いや推測がある要件に、誰に何を聞くかを書く

ヒアリングメモの書き方

1行に1発言なので、**行番号**で出どころを指せます

資料/day1/ヒアリングメモ_01_第1開発部_田中課長.md

24 ■ 売上実績表の作り方

25 聞き手：手順を順番に教えてもらえますか。

26 田中：まずKINTAIの月次締め画面、KIN050か、あそこから勤怠のCSVを出します。

27 田中：それをExcelに貼る。貼る用のシートがあって、そこにそのまま貼り付け。

28 田中：で、ピボットでPJコード別に作業時間を足す。社員の分はこれで出ます。

根拠の書き方

意味

ヒアリングメモ_01:L26

メモ01の26行目。田中課長の発言

メモは1本 120～180 行です。発言者・日時・出席者つきの議事メモの形で書いてあります。

矛盾の見つけ方

同じ話題を別の人が別の条件で話している所を探します

同じ話題を2本以上のメモで見比べます。数字・期限・誰がやるかは、ずれやすい所です（例は別の題材の会議室予約）

探し方	例
数字が違う	「予約は1日に20件ほど」と「多い日は100件を超える」
期限が違う	「キャンセルは前日まで」と「当日の1時間前まで受けている」
誰がやるかが違う	「予約は総務がまとめて入れる」と「各自が入れている」

見つけたら

片方に決めずに、両方の行番号を並べて要確認にします。どちらが正しいかは聞いて決めます。

あいまいな言葉の見つけ方

数字や条件に言い換えられない言葉は要確認に回します

別の題材での例

言葉	数字にする問い	聞く相手
「すぐ反映してほしい」	何分以内なら困らないか	使う人
「大量に登録する」	1回に何件、月に何件か	登録する人
「使いやすく」	誰が、どの操作を、何回で終わりたいか	使う人

AI は、あいまいな言葉を一般的な数字で埋めてしまうことがあります。要件一覧では数字を足さず、問いの形で要確認の列に書きます。

演習 D1-2 のねらいと目安

ねらい

3本のメモから要件を根拠付きで抜き出し、メモ同士の矛盾を表に出します。

できあがりの目安

- ☐ 要件一覧.md に要件が25件以上
- ☐ 全件に根拠（ファイル名:行番号）がある
- ☐ 「要確認」が3件以上ある
- ☐ メモ同士の矛盾が3つ以上載っている

使うファイル・作るファイル

区分	ファイル
使う	資料/day1/ヒアリングメモ_01～03（3本）、改修依頼書_KINTAI-2026-03.md
作る	成果物/day1/自分の要件候補.md
作る	成果物/day1/要件一覧.md

最初の10分で、自分の候補を10件以上書きます

■ 自分の手で ■ AIと一緒に

STEP1

[10min]

STEP2

[10min]

STEP3

[10min]

STEP4

[10min]

■ **STEP1** **自分で考える**

メモ3本を読み、要件の候補を10件以上書きます

■ **STEP2** **AI と詰める**

列を指定して要件を抽出させ、要件一覧.md に保存させます

■ **STEP3** **突き合わせる**

自分の候補と比べ、3つに分けます。食い違いは要確認に

■ **STEP4** **自分で判定する**

要件一覧を確定し、根拠の行を3件開いて確かめます

自分の候補は、**突き合わせの段**で初めて渡します

STEP2 抽出

〔@ でメモ3本と改修依頼書を指定〕

この4つから要件を抜き出して、表にしてください。

列は 要件ID / 要件 / 分類 / 根拠 / 発言者 / 要確認

- 要件IDは REQ-001 から順に振る
- 分類は 機能・非機能・制約・課題 のどれか
- 根拠は「ヒアリングメモ_01:L23」の形で必ず付ける
- 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る
- 1件に1つのことだけを書く
- メモにないことは書かない

表は 成果物/day1/要件一覧.md に保存してください。

STEP3 突き合わせ

@成果物/day1/要件一覧.md

@成果物/day1/自分の要件候補.md

自分の要件候補と要件一覧を
突き合わせて、次の3つに分けた
表を返してください。

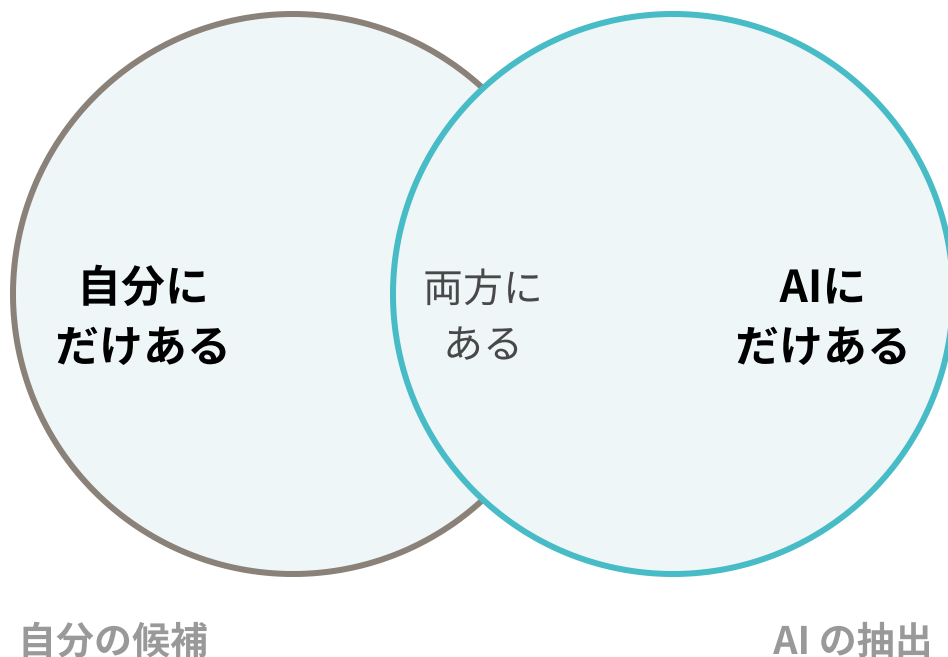
要件一覧はまだ書き換えしないで。

1. 自分の候補にだけあるもの
 2. 要件一覧にだけあるもの
 3. 同じことを指しているが、
中身が食い違うもの
- それぞれ根拠のメモの行も並べて。

STEP2 では自分の候補をまだ見せません。先に見せると、Claude がそれに引っぱられて比べられなくなります。

突き合わせの3分類

食い違いは**要確認**にして、誰に何を聞くかを書きます



自分にだけある

根拠の行を付けて足します

AIにだけある

根拠の行にあれば採ります

食い違う

要確認にし、聞く相手を書きます

重なっていても言い方が違うものは、同じ要件かどうかを根拠の行で見分けます。

根拠が付いていても、**その行を読むまでは信じません**

落とし穴	見分け方
根拠が聞き手の発言や見出しの行を指している	行を開き、発言者が誰かを見る
食い違う2つの発言のうち、片方だけが要件になっている	同じ話題の要件で、根拠が1つしか無いものを探す
1つの発言から要件が3つも4つも作られている	同じ根拠が並んでいる行を数える
要確認が0件のまま確定している	推測で埋めた要件が無いかをもう一度見る

STEP4 では、根拠の行を3件開いて確かめます。抜き取りの3件は、自分で選んでください。

要件一覧の根拠として、確かめやすい書き方はどれですか

- A ヒアリングで聞いた
- B 田中課長の発言
- C ヒアリングメモ_01:L23
- D 複数のメモから

答えはこの組の問題がそろってから確かめます。チャットに記号だけ書いてください。

2本のメモで同じ条件の話が食い違うときの扱いはどれですか

- A 日付の新しいメモに合わせる
- B 役職の高い人の発言に合わせる
- C AI にどちらか決めさせる
- D 両方の根拠を書いて要確認にし、聞く相手を書く

答えはこの組の問題がそろってから確かめます。チャットに記号だけ書いてください。

答えは **C・D** の順です

Q1

C**ヒアリングメモ_01:L23**

ファイル名と行番号があれば、誰でもその行を開いて確かめられます。

A・B・D は、どこを読めばよいかが決まりません。

Q2

D**両方の根拠を書いて要確認にし、聞く相手を書く**

どちらが正しいかは、要件を書く側では決められません。

A～C は、根拠の無い判断で片方を消してしまいます。

D1-2 早く終わった方へ

追加と考察

AI が拾わなかった要件は、メモのどんな書き方の中にありましたか。
行番号を添えて、要件一覧の下に書き足します。

発展課題

要件に優先度（初期リリース必須・次版）を付けさせ、
判断の根拠をメモの発言の行番号で示させます。

時間が余った方だけ取り組みます。答え合わせは講師が口頭で行います。

休憩

[10min]

再開は講師の合図でお知らせします

D1-3

要件定義書のドラフト作成

[35min]

作るもの

成果物/day1/要件定義書_v0.1.md と docx

要件定義書テンプレートの章立て

要件一覧は7章に入り、聞くことは12章に集まります

1 文書情報

2 改訂履歴

3 背景と目的

4 対象範囲

5 関係者

6 現状の業務と課題

7 要件一覧

8 機能要件の詳細

9 非機能要件

10 制約と前提

11 用語

12 確認事項

テンプレートは 要件定義書テンプレート.md です。先頭の「書き方」の章は、書き終えたら消します。

空欄の扱い

書けない欄は埋めずに、要確認と相手を書きます

推測で埋めた欄

区分	要件
性能	画面は3秒以内に表示する

根拠が無いのに、決まったことに見えます

要確認で残した欄

区分	要件
性能	要確認（聞く相手を書く）

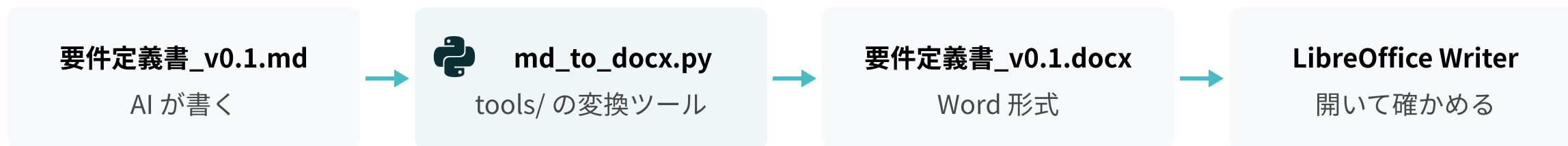
12章の確認事項に、聞くことが載ります

テンプレートの書き方の決まり

- 空欄は残さない。書けない欄には「要確認」と確認する相手を書く
- 推測で埋めた文には（要確認）を付け、確認事項の章に相手と聞くことを書く
- 表の列は消さない。行は足してよい

Markdown から Word への変換

AI には **Markdown** で書かせ、変換ツールで Word にします



実行されるコマンドの形

```
python tools/md_to_docx.py <入力.md> <出力.docx>
```

変換も Claude に頼みます。python が見つからないと言われたら、py で実行させます。変換ツールは見出し・段落・箇条書き・表・コードブロックを Word にし、書体は Meiryo にそろえます。

※ Python のロゴは Python Software Foundation の商標です。

演習 D1-3 のねらいと目安

ねらい

要件一覧をテンプレートに流し込み、Word の要件定義書にします。分からない箇所は埋めずに「要確認」で残します。

できあがりの目安

- ☐ 要件定義書_v0.1.docx が LibreOffice Writer で開ける
- ☐ テンプレートの見出しが全部ある
- ☐ 空欄が無い（書けない箇所は「要確認」と相手）

使うファイル・作るファイル

区分	ファイル
使う	成果物/day1/要件一覧.md、テンプレート/要件定義書テンプレート.md
作る	成果物/day1/要件定義書_v0.1.md
作る	成果物/day1/要件定義書_v0.1.docx

書かせる前に、一番書きにくい章を自分で決めます

■ 自分の手で ■ AIと一緒に

STEP1

[5min]

STEP2

[15min]

STEP3

[10min]

STEP4

[5min]

■ **STEP1** **自分で考える**

目次を見て、一番書きにくい章と、誰に確認するかを書きます

■ **STEP2** **AI と詰める**

要件一覧とテンプレートを渡し、Markdown で書かせます

■ **STEP3** **確かめる**

md_to_docx.py で Word にし、LibreOffice で開きます

■ **STEP4** **自分で判定する**

要確認の数を数え、多すぎる章は自分で書き直します

書きにくい章と確認相手を、自分のメモで差し込みます

〔@ でテンプレートと 成果物/day1/要件一覧.md を指定〕

テンプレートの見出しと順番をそのまま使い、要件一覧の中身を流し込んで、要件定義書を Markdown で書いてください。保存先は 成果物/day1/要件定義書_v0.1.md です。

- 「書き方」の章は守る決まりとして読み、書き終えたら消す
- ほかの見出しは消さない。書くことがない章も見出しを残し、「要確認」と書く
- 推測で書いた文には、文末に「(要確認: 確認する相手)」を付ける
- 各要件には、要件一覧の要件IDを残す
- 「食い違い」の要件は、どちらかに決めずに両方の言い分を並べる

私が一番書きにくいと思った章と、確認する相手は次のとおりです。

<ここにSTEP1のメモ2行を貼る>

Word への変換は STEP3 で別に頼みます。

ここからモードを「編集を受け入れる」にすると、保存のたびに止まらずに進みます。

開けたら、見出しと表の崩れを先に見ます

見るところ	見方
見出し	テンプレートの章が全部あり、番号と階層がそろっているか
表	罫線と見出し行が付いているか。テンプレートの列が消えていないか
空欄	空いたままの欄が無いか。書けない欄に「要確認」と相手があるか
要確認の数	章ごとに数える。多すぎる章は、自分で書き直す
文書情報	作成日・作成者・版が埋まっているか

崩れていたら、Markdown 側を直してから変換し直します。docx を直接直すと、次の変換で消えます。

D1-3 早く終わった方へ

追加と考察

非機能要件の章に、メモに無い数字（応答3秒など）が入っていないかを探します。
見つけたら根拠をたどり、無ければ要確認にします。

発展課題

用語の章を足し、ヒアリングメモで言い方が揺れている語
（工数・作業時間・稼働）を1つにそろえさせます。

時間が余った方だけ取り組みます。答え合わせは講師が口頭で行います。

D1-4

抜け漏れ探し

[25min]

作るもの

成果物/day1/要件定義書_v0.2.md と docx、要件一覧.md への追記、確認事項.md

抜け漏れを探す4つの観点

書いてあることより、**書いていないこと**を探します

観点	自分に問うこと
例外	途中で人や状態が変わったとき、どう扱うか。 いつもと違う日や入力をどう扱うか
運用	誰が、いつ、どの画面で作業するか。 間違いに気づいたとき、誰が直すか
権限	見てよい人と、見てはいけない人は誰か
データ移行	今あるデータは、新しい項目の上でどう見えるか

速さや止めてよい時間のような非機能は、次のページの6大項目で見ます。

非機能は IPA の6大項目を目次代わりに使えます

大項目	要求例（IPA 資料より）
可用性	運用スケジュール（稼働時間・停止予定など）、障害・災害時における稼働目標
性能・拡張性	業務量および今後の増加見込み、システム化対象業務の特性
運用・保守性	運用中に求められるシステム稼働レベル、問題発生時の対応レベル
移行性	新システムへの移行期間および移行方法、移行対象資産の種類および移行量
セキュリティ	利用制限、不正アクセスの防止
システム環境・エコロジー	耐震／免震、温度／湿度などのシステム環境、CO2排出量や消費エネルギー

全体は6大項目・34中項目・116小項目・238メトリクスに体系化されています。今日は大項目で足ります。

出典 IPA「非機能要求グレード」実践セミナー 講義資料（2019年3月4日、p.18-19）
<https://www.ipa.go.jp/archive/files/000072715.pdf>

データから抜けを探す

メモに無いことが、**データの中**では起きています

資料/データ/勤怠工数CSV_202609.csv（12列のうち8列を表示）

社員ID	表示名	日付	実働分	PJコード	作業区分	作業分	備考
S1001	田中	2026/09/01	480	PJ-2026-011	保守対応	360	
S1001	田中	2026/09/01	480	PJ-2026-014	製造	120	設計レビューの指摘対応

STEP1 のヒント

9/22 の行と、備考列

LibreOffice Calc で開くとき

「テキストのインポート」の画面で、文字エンコーディングが Unicode (UTF-8) になっていることを確かめてから OK を押します。1日に複数の PJ があると、同じ日付が複数行になります。

演習 D1-4 のねらいと目安

ねらい

AI に頼む前に自分で抜けを探します。AI の抜け漏れ探しと比べ、どちらが何を拾うかを知ります。

できあがりの目安

- ☐ 自分で見つけた抜けが1件以上、要件定義書 v0.2 に反映してある
- ☐ 同じ抜けが 要件一覧.md にも、要件IDと根拠つきで入っている
- ☐ 確認事項リスト（相手・聞くこと・なぜ）が5件以上ある

使うファイル・作るファイル

区分	ファイル
使う	資料/データ/勤怠工数CSV_202609.csv、成果物/day1/要件定義書_v0.1.md
作る	成果物/day1/要件定義書_v0.2.md と 要件定義書_v0.2.docx
追記	成果物/day1/要件一覧.md（続きの要件IDで足す）
作る	成果物/day1/確認事項.md

AI に頼む前に、データを8分見ます

■ 自分の手で ■ AIと一緒に

STEP1
[8min]

STEP2
[7min]

STEP3
[5min]

STEP4
[5min]

■ **STEP1** **自分で探す**

CSV を Calc で開き、要件定義書に書いていないことが起きている行を1つ探します

■ **STEP2** **AI と詰める**

例外・運用・権限・データ移行の観点で、抜け漏れを探させます

■ **STEP3** **並べる**

自分の発見と AI の指摘を並べ、重なりと違いを見ます

■ **STEP4** **自分で判定する**

v0.2 と要件一覧に反映し、確認事項リストを作ります

自分の発見は見せずに頼み、反映は**自分の言葉**で書きます

STEP2 抜け漏れ探し

〔@ で 成果物/day1/要件定義書_v0.1.md と 資料/データ/勤怠工数CSV_202609.csv を指定〕
要件定義書の抜け漏れを、例外・運用・権限・データ移行の4つの観点で探してください。

- 観点ごとに表にし、列は 観点 / 抜けていること / 根拠 / 誰に確認するか
- 根拠は、CSV なら社員ID と日付、ヒアリングメモならファイル名と行番号
- 要件定義書はまだ書き換えしないでください

STEP4 反映

〔@ で 成果物/day1/要件定義書_v0.1.md と 成果物/day1/要件一覧.md を指定〕
次の抜けを反映して、成果物/day1/要件定義書_v0.2.md として保存してください。v0.1 は残してください。

<ここに、反映すると決めた抜けを自分の言葉で書いて貼る>

同じ抜けを 成果物/day1/要件一覧.md にも、続きの要件IDで足してください。

根拠は、CSV なら社員ID と日付、メモならファイル名と行番号にしてください。

続けて、tools/md_to_docx.py で 成果物/day1/要件定義書_v0.2.docx も作ってください。

最後に v0.2 の（要確認）を全部拾い、成果物/day1/確認事項.md に表でまとめてください。

列は 相手 / 聞くこと / なぜ聞くか の3つです。

確認事項リストの形

聞く相手が決まらない確認事項は、まだ書けていません

No	相手	聞くこと	なぜ聞くか	要件ID	期限
1	(名前)	(はい・いいえか 数字で答える問い)	(決まらな いと書け ない章)	REQ-xxx	(次回の ヒアリング)
列					
相手		書き方			
相手		1人に決める。「関係者」「現場」とは書かない			
聞くこと		答えが1つに決まる形で書く。選択肢があれば並べる			
なぜ聞くか		答えが無いと、要件定義書のどの章が書けないか			

要件として扱うか、運用で済ませるかも、ここで一緒に聞きます。

AI の指摘は、**根拠のあるもの**と一般論に分けます

自分の発見

データの中で実際に起きている行が根拠
件数を数えられる
見つけるのに時間がかかる

AI の指摘

観点を広く並べるのが速い
どのシステムにも当てはまる一般論が混ざる
根拠の行が付いていないものがある

指摘の種類	扱い
根拠のある指摘	要件定義書 v0.2 に反映する
一般論	確認事項に回すか、捨てる。捨てた理由を1行残す

D1-4 早く終わった方へ

追加と考察

CSV で見つけた気になる行を、要件として扱うべきものと、運用で済むものに分けます。分けた理由を1行ずつ書きます。

発展課題

確認事項を、ヒアリングの相手ごとの質問票にさせます。
1人あたりの質問が多すぎないかを自分で見ます。

時間が余った方だけ取り組みます。答え合わせは講師が口頭で行います。

指示の型3つ

今日の4つの演習は、すべて**この3つ**で送りました

渡す

依頼書・メモ・テンプレートを @ で指定する

D1-1 ②

禁じる

メモにないことは書かない。推測には（要確認）

D1-1 ③

根拠を求める

ファイル名と行番号を付けさせ、自分で開く

D1-2～D1-4

Day2 の画面設計も、Day3 のレビューも、同じ3つで頼みます。

要件定義書で疑う場所

もっともらしく埋まっている欄から疑います

- ☐ 根拠の無い要件
- ☐ 根拠はあるが、その行には書いていない要件
- ☐ （要確認）の付いていない推測
- ☐ メモに無い数字（非機能の章に出やすい）
- ☐ 同じものを別の語で書いている箇所

今日の演習で使った指示の型はどれですか

- A 短く・丁寧に・何度も送る
- B 渡す・禁じる・根拠を求める
- C AI に任せる・待つ・そのまま採る
- D 全部渡して、全部採る

答えはこの組の問題がそろってから確かめます。チャットに記号だけ書いてください。

非機能要件にメモに無い「応答3秒以内」がありました。どうしますか

A 要確認を付け、確認事項に聞く相手を書く

B よくある値なのでそのまま残す

C 数字を消して「速やかに」と書く

D 余裕を見て「5秒以内」に直す

答えはこの組の問題がそろってから確かめます。チャットに記号だけ書いてください。

抜け漏れ探して、AI に頼む前に自分でやることはどれですか

- A テンプレートの見出しを消しておく
- B AI に観点を考えさせる
- C データを開き、書いていないことが起きている行を探す
- D 要確認の欄をすべて埋めておく

答えはこの組の問題がそろってから確かめます。チャットに記号だけ書いてください。

答えは **B・A・C** の順です

Q1 **B** 渡す・禁じる・根拠を求める

今日の4つの演習は、すべてこの型で送りました。

Q2 **A** 要確認を付け、確認事項に聞く相手を書く

メモに無い数字は、残しても言い換えても根拠が無いままです。

Q3 **C** データを開き、書いていないことが起きている行を探す

自分で探してから頼むと、AI の指摘と比べられます。

Day2 に持っていくもの

今日の要件定義書が **Day2 の画面設計**の入力になります

- ☐ 成果物/day1/D1-1_比較.md
- ☐ 成果物/day1/自分の要件候補.md と 要件一覧.md
- ☐ 成果物/day1/要件定義書_v0.2.md と 要件定義書_v0.2.docx
- ☐ 成果物/day1/確認事項.md

Day2 10月15日（木）画面設計書とモック

終わらなかった方には、Day2 の最初に 成果物/合流用/へ合流版を配ります。
中身は 要件一覧.md と 要件定義書_v0.2.md・docx です。

質疑とアンケート

ご質問はチャットか口頭でどうぞ

[15min]

アンケートへのご回答をお願いします

URL はチャットでお送りします

教材サイト（ハンズオンガイドとスライドを読み返せます）

<https://idup09.give-app.net>



株式会社ギブリー

〒150-0036 東京都渋谷区南平台町15-13 帝都渋谷ビル8F

<https://givery.co.jp/>

10月15日（木）の Day2 でお会いしましょう。