

ガイダンス + 作業分解（WBS） ー 作りきたいものを、実現する方法論

ソフトウェア開発（ISC322）／ 2026年度 秋学期

2026年9月16日（水）5-6-7限 15:25-19:30 国際基督教大学 情報科学メジャー

担当: **太田啓路** ohtak@icu.ac.jp

今日のゴールと流れ（15:25-19:30）

先週 9/9 は出張（VR学会）のため休講にしました。ご迷惑をおかけしました。今日はその分、**ガイダンス+作業分解**を続けてやります。

時間	内容
5限 15:25-16:40	ガイダンス — この授業は何を学ぶか／私がいま作っているもの／スケジュール・評価・ルール／使う AI と無料の範囲／名言集／まずやってみる（MVP）
6限 16:50-17:30	作業分解の講義 — プロジェクトとは／WBS の3ルール／ガント・PRP／ 太田の事例：AI で WBS を作る ／参考事例
6限後半～7限 17:30-19:20	実習 — 自分のプロジェクトを決め、手で分解 → AI に分解させ → 自分で検証 → ガント → 3人で見直し → 再計画 → 提出
19:20-19:30	コメントシート（授業時間内に Moodle へ）

今日のゴール

「自分のプロジェクト」の WBS が、AI と他人の目を通して、実行できる形になっている

作るものはアプリでなくていい。卒論・就活・学園祭・展示・事業——「実現したいこと」なら何でも。

自己紹介 ① 研究者として

- **太田 啓路**（おおた けいじ）／ 情報科学メジャー シニアレクチャー。担当: 情報科学概論・コンピュータゲーム・映像情報メディア技術基礎・ソフトウェア開発・インタラクティブコンテンツ制作 など
- **博士（国際情報通信学）** — 早稲田大学 国際情報通信研究科 修了
- 博士論文: 「安全かつ快適な3次元映像表現に関する人間工学的研究」
- 研究テーマ: **3D ゲームにおける映像酔い**（SEGA 共同研究）、立体映像のインタラクティブ応用、文化財の LBVR（ロケーション型 VR）
- 一貫した問い: 「**その映像は、人の身体にとって安全で快適か**」

研究も会社も「作れる」だけでは足りない。**誰のために・何を確かめて・どう届けるか**——この授業で扱う「8割」を、私自身が毎日やっています。

自己紹介 ② 映像・VR の会社をやっています

株式会社リ・インベンション（2013年設立／前身 QXD は 2008年～）代表。秋葉原 UDX の 3D シアター・3D スタジオ担当（2006年～）が出発点。

VR ライブ	HMD を 40～100 台同時制御する没入型ライブ
8K VR 撮影	ライブ・舞台・文化財の 360°／180° 3D 制作
2D→3D 変換	映画・シアター・プロモーション映像の立体化
生成 AI × 映像制作	AI の生成力とプロの審美眼を組み合わせる
3D スキャン	フォトグラメトリ・Gaussian Splatting
映像表示システム開発	ドームシアター・多面スクリーン・同期上映

仕事の大半は「コードを書く」ではなく、**企画・見積・計画・調整・検証・納品**。だからこの授業を担当しています。

➤ [株式会社リ・インベンション 公式サイト](#)

自己紹介 ③ これまでの仕事から



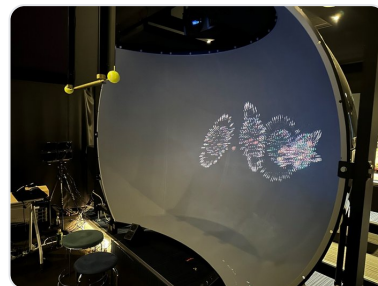
2010 ムーミン
北欧初の 3D 映画



2014 ドラえもん
全編 2D→3D



2020- ドコモ
新体感ライブ 8K VR



2024 NHK
8K3D ドームシアター



2018- 青森
祭り 360°3D シアター

2010	『Moomins and the Comet Chase (ムーミン谷の彗星)』 北欧初の 3D 映画制作
2014	映画『STAND BY ME ドラえもん』 全編 2D→3D 変換
2016	『ONE PIECE FILM GOLD』 一部 2D→3D 変換 / 生中継 180°3D VR
2018	8K VR 撮影 (アイドルライブ) / 青い森ホール「3D 青森祭りの魂」 360°3D シアター
2020-21	NTT ドコモ「新体感ライブ CONNECT」 8K VR 制作

空飛ぶ一畳敷 — 学内文化財「泰山荘・一畳敷」のロケーション型 VR (LBVR)



一畳敷の室内（写真：湯浅八郎記念館 図録）



3D Gaussian Splatting による再構築



複数人での LBVR 体験（湯浅八郎記念館）

学会論文（添付）

論文 PDF: 学内文化財における 3D Gaussian Splatting を活用した Location-Based Virtual Reality の試作 (VRSJ 2026)

ポスター（画像）

制作ダッシュボード（受講期間中の限定公開）

ダッシュボード URL: <https://ri-projects.pages.dev/soratobu/>

今日の授業との関係

制作期間は **10 日**。作業を 12 に分解し、PC 3 台と AI コーディングエージェントで並列に作った。

その「分解の仕方」を、6限で**実物**として見せます。

論文の共著: 学内の共同研究者・会社の共同開発者・湯浅八郎記念館。文化財の撮影許可と権利確認も「作業」の一部でした。

- 松浦武四郎（「北海道」の名付け親・1818-1888）が 1886 年に建てた**畳一枚の書斎**。全国の古社寺の古材 90 点。1930 年代に三鷹へ移築、1999 年 国の登録有形文化財。**このキャンパスの泰山荘に現存**
- 表門・一畳敷・敷地を 3DGS で再構築し、Meta Quest 3 を **2 人同時に** LAN で同期。畳の上に座って体験する「空飛ぶ畳」の演出、武四郎が語り部
- 先週 **日本バーチャリアリティ学会 第31回大会（VRSJ 2026・富山）** で発表・デモ → 10 月 泰山荘特別公開 → 11 月 一般公開予定（湯浅八郎記念館と連携）

NO！トレ — 消費者トラブルを疑似体験する AI 教材（青森県の消費者教育事業）



トップ（呼び名・シナリオ
選択）

体験する内容を選ぶ

「非通知」からの着信——
ここから AI と会話

- 詐欺の電話・勧誘を **AI が相手役**になって再現し、「見抜く力・断る勇氣」を体験で学ぶスマホ／タブレット教材。体験の最後は消費者ホットライン 188 へ
- 県内 7 市の商業施設を巡回して啓発活動。契約から翌年 2 月末までの**長期プロジェクト**（企画会社＋技術会社＋県の連携）。私の会社は AI 教材の設計・開発・端末運用を担当
- いまは「県の指摘 7 件を 9/28 までに直す」改善計画の真っ最中。方言（津軽弁）対応の go/no-go を 9/19 に判断

今日の授業との関係

一畳敷が「10 日の短距離走」なら、こちらは「9 か月のマラソン」。
フェーズ表と「合格の門」（何ができたら完了か）の書き方を、参考事例で見せます。

計画ダッシュボード（受講期間中の限定公開）

➤ NO！トレ 計画ボード（WBS・ガント・決定事項）

※ 授業で使うのはブラウザの Claude と Gemini だけです。

真葛焼 プロジェクションマッピング — 実物の花瓶に映像を投影する

- 宮川香山 真葛ミュージアム（横浜）の代表作「崖二鷹大花瓶」を 3D プリントした原寸大の花瓶に、映像を投影する常設展示
- 約 2 分・60fps・縦 4K（プロジェクタを 90°回転）。演出家（伊藤有壱氏）の絵コンテに基づく 23 カット
- 制作完了は今月末、展示は 10 月 3 日（土）から。いまが追い込み
- 工程: Blender で 3D の奥行き・マスクを出す → 画像生成 AI でキーフレーム → 花瓶だけは実物を固定座標に合成（AI に描かせない）→ 動画化 → 4K 拡大 → 組み立て → 現場でマッピング

今日の授業との関係

絵コンテが 15 カット → 23 カットに増えたとき、作り直さずに済んだのは「カット構成を 1 つの一覧（manifest）で管理していた」から。
作業の一覧を 1 か所に持つと、計画の変更に強くなる——これが今日の WBS の効き目です。後で分解の例も見せます。

➤ 宮川香山 真葛ミュージアム

この授業は「何かを実現するための方法論」を学ぶ授業

ソフトウェア開発を題材にするが、
身につけるのは「つくりたいものを、他の人と協力して実現する」ための手順。

この授業で扱うこと

- 目的を決め、作業に分解し、計画を立てる（PM・WBS・ガント）
- 失敗に備える（リスク管理）／小さく始めて確かめる（MVP・KPI）
- 使う人の目で見える（UX/UI）／正しく動くか確かめる（テスト）
- 何を作るかを言葉と図で決める（仕様・モデリング）
- チームで作る、発表し、フィードバックで直す（WS・評価）

過年度の受講生の「プロジェクト」

卒論の執筆／就活（For 内定）／公務員試験合格／TOEIC 800 点／学内新聞づくり
／カフェ開発／ICU スケジュールアプリ／速読アプリ／Unreal Engine のゲーム／
進路決定

つまり

半分はソフトウェアではない。でも手順は同じ。
今日の実習でも、**自分が本当に実現したいこと**を題材にしてください。

ソフトウェア開発の流れ — 「プログラミングは2割」

営業 → ニーズのヒアリング → 企画提案（見積） → 契約 → 要件定義 → 設計 → プログラム開発 → バグ修正 → テスト → 運用テスト → 納品 → 保守（+マニュアル作成・使い方の指導）

私が毎年シラバスに書いている一文

「実際のシステム開発において、プログラミングが占める作業は2割程度に過ぎない」

残り8割は、何を・誰のために・どう作り、どう確かめ、どう届けるか。PM の考え方は、あらゆる仕事で必要になる。

フェーズ	主な作業	典型的な失敗
要件定義	関係者への聞き取り・整理	「作ったけど使われない」
設計・計画	構造の選択・作業分解・日程	「後から直せない」「終わらない」
実装	コードを書く（← ここが2割。AI が最も得意）	バグ・技術的負債
テスト	動作・品質・安全性の確認	スキップして後から炎上
運用	監視・障害対応・機能追加	「誰も全体を知らない」

AI で実装が速くなった分、「何を作るか」「正しいか」「誰が責任を持つか」の重みが増した——これは断定ではなく、今日のデータと一緒に確かめます。

スケジュール（9/16～11/4・毎週水曜 5-6-7限・I-212）

日付	テーマ	実習（6限後半+7限）	課題
9/16	ガイダンス+作業分解（WBS）	自分のプロジェクトの WBS を AI と作り、検証する	課題1 出題
9/23	祝日だが実施 リスク管理	自分の WBS にリスク登録票・プランB → 計画を直す	
9/30	MVP・KGI/KPI・再計画 — 小さく素早く始める	自分のプロジェクトの MVP と KPI ツリー	課題2 出題
10/7	UX/UI	紙スケッチ → AI で画面案 → 相互批評	課題1 締切 10/7
10/14	テスト・デバッグ	受入条件を書き、AI が作ったテストをレビュー	課題2 締切 10/13
10/21	モデリング（仕様＝完成の条件）+ チーム結成	3人チームで課題→CANVAS→仕様 v0	WS 準備
10/28	ワークショップ：制作・発表	3限まるごと制作、最後に発表（1チーム6分）	課題3 出題
11/4	成果物の評価・フィードバック	相互試用 → 改善 → 再検証	最終レポート 締切 11/18

日程は変わることがあります。確定版は毎回、授業内と受講生ページで案内します。

評価と課題

区分	割合	中身
出席	50%	毎回の Moodle 出席 + コメントシート（欠席レポートの扱いは次ページ）
課題	50%	課題1 計画書（15%）／課題2 MVP・KPI・受入条件（15%）／課題3 WS 成果物（8%）+ 最終レポート（12%）

課題	出題	提出物	見るところ
課題1	今日（9/16）→ 10/7 締切	自分のプロジェクトの計画書: WBS+ガント+リスク登録票（9/23）+ 最初の一步の記録 （担当 = 人 / AI の列つき）	100% ルール・依存・人と AI の分担の根拠・ 検証の記録
課題2	9/30 → 10/13 締切	MVP 定義 + KPI ツリー + 受入条件 2 本	小さく始める設計・測り方
課題3	10/28 → 11/4	チーム制作物 + 仕様 + AI/人間の分担 + 検証ログ	仕様どおりか・検証したか
最終レポート	11/4 → 11/18 締切	A4 2枚: 初期計画 → 受けたフィードバック → 採否の理由 → 直した結果 → 卒論・就活への 応用	判断と証拠。改善量ではない

テスト・期末試験はなし。授業外学習の目安は週 225 分。課題は日本語・英語どちらでも可。

受講のルール — 出席・コメントシート・欠席レポート

出席とコメントシート

- 出席は **Moodle の出席機能** で、授業時間内に各自登録
- コメントシートは毎回、**授業時間内に Moodle へ**（19:20 に時間を取ります）。やむを得ず退出する場合は当日 24:00 まで
- 「授業を受けたことが分かる」内容を書く。コメントは次回シェアし、質問には次回答えます
- 共有したくない部分は `//共有なし 文章 //` で囲む

今日のコメントシート（初回）

- 自己紹介（メジャー・学年・プログラミング/AI の経験）
- つくりたいもの・プロジェクトにしたいもの（1つ以上。なぜそれか）
- 今日、AI が見落としたこと／自分が直したこと
- 質問・要望

欠席した場合（欠席レポート）

資料（この HTML+補講ビデオ）を読んで欠席レポートを出すと、その回の出席点の**8割**で出席扱い。忌引き・教育実習はレポートを出せば**10割**。

分量	締切	提出先
A4 2枚（3コマ分）	授業日の1週間後 24:00	Moodle「欠席レポート」（コメントシートと別）

書くこと: ①日付 ②要点の要約（自分の言葉で） ③印象に残った事例・名言と理由 ④自分のプロジェクトへの適用 ⑤質問 + **その回の実習の成果物**（今日なら WBS）

欠席の連絡はメール（ohtak@icu.ac.jp）へ。事前でも事後でも構いません。

使う AI は2つ、どちらも無料 — 「無料はどこまで？」

サービス	無料でできること（2026-09-16 確認）	上限・注意
Claude claude.ai	チャット・コード生成・表や図の生成（Artifacts）・Projects 5 個まで・ファイル添付（PDF/xlsx など）。モデルは Sonnet/Haiku	5 時間ごと にリセットされる回数上限（数は非公表）。Artifacts を使うには Settings > Capabilities > 「Code execution and file creation」を ON
Gemini gemini.google.com	チャット・Canvas（文書／コードの横並び編集）・画像生成・Deep Research	5 時間ごと+週の上限。添付は 32k トークンまで。混雑時に一部機能が使えないことあり
ChatGPT （参考）	文章・表・Mermaid のテキストは無料で出せる。データ分析（Python 実行）・画像生成・Canvas は無料で使えるが 別枠の上限	ガントを「画像」や「xlsx」で出させると、途中で止まる人が出る。今日使わない

今日の運用ルール

- ① 標準は **claude.ai**、詰まったら **gemini.google.com** に同じ文を貼って続行（だから2つ）
- ② 成果物は **Markdown の表**と **Mermaid のテキスト**に固定 — どのサービスでも無料で出せる形
- ③ 両方止まったら、配布の「保存済み AI 初稿」で続ける。AI を使うこと自体は必須ではない

課金は求めません

自分で課金しているサブスク（ChatGPT Plus・Claude Pro・Gemini など）があれば、そちらを使ってよいです。

Copilot・Cursor・Claude Code などは「世の中にはこういう道具もある」と紹介はしますが、提出物の前提にはしません。

上の表は公式ヘルプで確認した範囲。数値は変わるので、迷ったら各社の料金ページを見てください。

AI 利用の約束（シラバス §13）

1. 提出物は、自分で説明できるように理解する努力をする（分からない部分は調べる・AI に聞き直す・人に聞く）
2. AI に**任せた部分**と、自分が**判断した部分**を提出時に明示する
3. AI 生成物は出発点。**レビュー・検証・改善は提出者の責任**
4. 個人情報・機密・第三者の著作物を**プロンプトに入れない**

なぜ「理解しようとする」ことが大事か

理解していないものは、**直せない**（壊すしかない）・**間違いに気づけない**
（動いてる＝正しい、と思い込む）・**人に説明できない**。全部を理解できなくても、「どこが分かっていないか」を言えるようにしておく。

AI の利用は推奨

禁止ではない。「使いこなして、責任を持つ」を 8 週間で身につける。
今日の実習は、まさにその練習。

伝説的な開発者の言葉 ①（出典が確かなものだけ）

"Adding manpower to a late software project makes it later."

遅れているソフトウェア開発に人を足すと、さらに遅れる。

Fred Brooks, *The Mythical Man-Month* (1975)

→ AI エージェントを増やしても、レビューする人の時間は増えない。今日の6限で。

"There was no choice but to be pioneers; no time to be beginners."

先駆者になるしかなかった。初心者でいる時間はなかった。

Margaret Hamilton（アポロ誘導ソフト責任者）, WIRED 2015

→ 前例のないプロジェクトの始め方。まず動かす。

"It's easier to ask forgiveness than it is to get permission."

許可を取るより、あとで許してもらう方が簡単だ。

Grace Hopper（COBOL の母）, *Chips Ahoy* 1986

→ まず試作して見せる（MVP）。ただし「承認前に告知」とは違う——実習の練習問題で。

伝説的な開発者の言葉 ②

"Program testing can be used to show the presence of bugs, but never to show their absence!"

テストでバグの存在は示せるが、不在は示せない。

Edsger Dijkstra, *Notes on Structured Programming* (EWD249, 1969)

→ AI が作ったテストが全部通っても「正しい」証明ではない (10/14)。

"You can't trust code that you did not totally create yourself."

自分で全部作ったのではないコードは、信用できない。

Ken Thompson (UNIX の作者), Turing Award Lecture 1984

→ 全自作せよ、ではない。「信用の根拠をどう作るか」を問う言葉 (9/23 リスク)。

"Beware of bugs in the above code; I have only proved it correct, not tried it."

上のコードのバグに注意。私は正しいと証明しただけで、試してはいない。

Donald Knuth (『The Art of Computer Programming』著者), 1977 年の書簡

→ 「正しいはず」と「動かして確かめた」は別。AI の出力にもそのまま当てはまる (10/14)。

残り (Brooks「銀の弾丸はない」/Hoare/Fowler/Torvalds/Beck ほか) は、各回の冒頭で1つずつ。受講生ページに一覧を置きます。

AI 時代一枚 — 使う人は増えた。「信頼」「安全」は伸びていない

以下はいずれも公開されている調査レポートです（発行元・対象・時期を併記）。数字は各社の発表値。原文リンクは次ページ。

調査（発行元・種類）	対象・時期	分かったこと
DORA 2025「State of AI-assisted Software Development」 Google Cloud の研究チーム DORA（DevOps Research and Assessment）が毎年出す業界調査。開発組織の生産性研究として最も引用される	世界の技術者 約 5,000 人への調査（2025年6～7月）+100 時間超の聞き取り。2025-09-24 公開	開発者の 90% が仕事で AI を利用（前年比 +14pt）。AI は「増幅器」——チームの強みも弱みも増幅する。作業の速さ（スループット）は上がったが、 リリース後の安定性は下がる 傾向が続く。30% は AI が書いたコードを「ほとんど／全く信頼していない」
Stack Overflow Developer Survey 2025 開発者 Q&A サイト Stack Overflow の年次アンケート。世界最大級の開発者調査	177 か国 49,000 人超 の開発者（2025年）	AI ツールの利用は 84% 。だが出力を「正確だと信頼」は 29%、「高く信頼」は 3% 。 66% が「惜しいが、ピットリではない（almost right, but not quite）」と回答。45% が AI コードのデバッグに時間を取られる
Veracode「2026 GenAI Code Security Report」 アプリケーション・セキュリティ企業 Veracode の年次レポート	100 超の AI モデルに同じコーディング課題を解かせ、セキュリティ検査にかけた（2026-07 公開）	AI が書いたコードのセキュリティ合格率は平均 56% ——4 年間ほぼ横ばい。モデルが賢くなくても「安全なコード」は増えていない。XSS（クロスサイトスクリプティング）対策の合格率は 15%
DORA 2026「ROI of AI-Assisted Software Development」 同じ DORA チームの追加レポート（2026-04）。※本資料は InfoQ の要約記事経由	導入企業のコスト・効果分析	導入初年度は Jカーブ ——検証とワークフローの適応で、一時的に生産性が下がってから上がる
Faros AI / LinearB（2026） 開発分析ツール企業の利用データ集計。※二	各社の顧客データ（数万人規模）	AI が手伝った変更提案（PR）はサイズ約 2.5 倍・レビュー待ち約 5 倍。コードレビューに使う時間が大きく増加

AI 時代一枚 — 原文リンクと、授業での読み方

原文（すべて無料で読める）

- DORA 2025 発表記事（Google Cloud Blog）
- DORA 2025 レポート本体（dora.dev・PDF）
- Stack Overflow Developer Survey 2025 — AI の章
- 同 プレスリリース（49,000 人・信頼 3%）
- Veracode 2026 GenAI Code Security Report
- DORA 2026 ROI レポートの要約（InfoQ）
- Faros AI / LinearB の集計記事（二次情報）

確認日 2026-09-16。DORA 2026 と Faros/LinearB は原典 PDF を直接確認しておらず、数値は「傾向」として扱ってください。

3 つの調査に共通するパターン

1. 使う人は増えている（90%・84%）
2. 速くはなる（DORA: スループット向上）
3. だが信頼・安定性・安全性は伸びていない（信頼 3%、安定性は低下、安全合格 56% 横ばい）

今日の授業に効く読み方

速く作れるようになった分、「確かめる」「決める」の仕事が増えた。

だから WBS には、AI に任せる作業と一緒に「人が確認する作業」と「その時間」を書く。

注意: 調査はいずれも「回答者の自己申告」か「特定ツールの利用データ」。AI の効果は課題や使い方で大きく変わるので、「AI で必ず速くなる」とも「危ないから使わない」とも言えない。

まずやってみる — 小さく、素早く始める (MVP)

MVP (Minimum Viable Product) = 仮説を確かめるための、最小の試作品。Eric Ries『リーン・スタートアップ』(2011)。

- **Zappos** (靴の EC) : 在庫を持たず、靴屋で撮った写真だけを載せて「売れるか」を確かめた
- **Dropbox**: 製品の前に 3 分の説明動画を公開し、登録待ちが一晩で 5 千→7.5 万人
- **Facebook**: 1 つの大学だけで始めた

私が事業計画を書くとき何度も参考にした本が『起業の科学』(田所雅之)。「作り込む前に、欲しがる人がいるか確かめる」。

MVP の本格解説は別の回 (9/30)

MVP の作り方・KGI/KPI・リーンスタートアップは **9/30 の授業で扱います**。今日は「小さく始める」という姿勢だけ持ち帰ってください。

今日から

**WBS は「完璧な計画」を作る道具ではない。
「最初の一步」を決めて、走り出すための道具。**

実習の最初に「**最初の 1 週間でやる最小の一步**」を 1 行書いてもらいます。

卒論なら「章立て 1 枚を先生に見せる」。アプリなら「画面 1 枚を紙に描いて友達に見せる」。

休憩中に考えておくこと: 「自分のプロジェクトにするなら、何か」(1 行)。

Hopper の「許可より許し」も同じ精神。ただし練習問題で見るように、**承認が要る作業を飛ばす**のとは違う。

プロジェクトとは — 「開始と終了があり、成果物・期限・制約がある仕事」

定義	開始と終了のある一連の作業。時間・資源・目標によって制約され、 具体的な成果物・期限・予算 がある。予算で投入できる資源（人・モノ・カネ）が決まる
定常業務との違い	プロジェクトには明確な目標と開始・終了がある／定常業務（経理・人事・サポート…）は何度も繰り返す／プロジェクトには目的を達成するための組織（チーム）ができる
マネージャーの課題	時間・コスト・品質のバランスを取り、円滑に進める。不意打ちは日常茶飯事——「曖昧さへの耐性」「変更管理の能力」「顧客第一の思考」が要る
大きなプロジェクト	マンハッタン計画（1942・「プロジェクト」という言葉が初めて使われた）／コロンブスの航海／忠臣蔵（討ち入りまでの 1 年 9 か月）／ピラミッド建造／オリンピック・万博／新幹線・ダム／映画・大規模ゲームの制作／新製品の開発／会社の設立
身近なプロジェクト	卒業論文／就活（For 内定）／公務員試験・TOEIC・資格試験／学園祭の出し物・サークルの公演／留学の準備／引っ越し／旅行の計画／同人誌・ZINE の制作／YouTube チャンネルの立ち上げ／小さなアプリや Web サイトの開発
太田の今のプロジェクト	空飛ぶ一畳敷 VR（10 日）／NO！トレ（9 か月）／真葛焼のプロジェクションマッピング（展示は 10/3 から）／相馬野馬追 VR
過年度の受講生	速読アプリ／カジュアルゲーム／ICU スケジュールアプリ／カフェ開発／学内新聞づくり／進路決定／Unreal Engine のゲーム／教育ミニゲーム集

今日決めてほしいこと

「自分のプロジェクト」を 1 つ。成果物と期限があって、作業が 30 個は出そうなもの。

ソフトウェアでなくてよい。本当に実現したいことを選ぶほど、実習が自分の役に立つ。

プロジェクトマネジメントに共通する 12 の作業 — 今日は「分解」「順序」「スケジュール」

#	作業	中身	この授業で扱う回
1	目標の設定	何を、いつまでに、誰のために	今日 (Step 0) ・ 9/30 KPI
2	作業の分解	WBS—大きな仕事を管理できる単位に砕く	今日
3	作業順序の図示	依存関係・ネットワーク図・クリティカルパス	今日
4	スケジュールの作成	ガントチャート・マイルストーン・バッファ	今日
5	予算の作成	人件費・外注・道具	10/21 (WS 準備)
6	チーム内の調整	役割・会議・規約	9/30 ・ 10/21
7	報告	関係者への進捗報告	毎回のコメントシート
8	コミュニケーション	相談・交渉・合意	9/30
9	情報提供	資料・記録の共有	10/21 (仕様)
10	意思決定	go/no-go ・ 優先順位 ・ やめる判断	9/23 リスク
11	進捗の計測	実績を入れて計画と比べる (PRP)	今日 Step 7 ・ 毎週
12	テーラリング	型を自分の案件に合わせて削る・足す	今日 (3 ルールの当てはめ)

私の現場の道具 (実話)

仕事では長く **Backlog** (課題管理ツール) を使ってきた。タスクを切って、担当と期限を付けて、進捗を見る—これを**手で**やっていた。いまは **GitHub Issues + AI エージェント**。作業の切り方は同じで、切った 1 つ 1 つを AI に渡せるようになった (後で実物)。道具は毎年変わる。変わらないのは「分

WBS（Work Breakdown Structure・作業分解図）とは

- プロジェクトの作業を、ある切り口で**階層構造**に示す（木の図、または箇条書きの表）
- **ワークパッケージ**= WBS の最下位の作業。プロジェクトを大きな単位（マイルストーン）に分け → ワークパッケージを洗い出す

WBS の役割

- 分かる範囲と分からない範囲を把握し、**何をすべきか**を明確にする
- メンバーで「成功したイメージ」を共有し、必要な作業を洗い出す
- 作業の順序を論理的に決め、担当と資源（予算）の基礎になる
- 実際は、進めながら条件が明確になった時点で**計画を更新する**

卒業論文プロジェクト

- └ プロジェクト管理（週1回の進捗確認）
- └ 研究計画
 - └ 章立て1枚を作る（3h）★最初の一步
 - └ 教員に見せて直す（2h+待ち3~5日）
 - └ 先行研究15本を要約（7h）
- └ 調査・実験
- └ 執筆
- └ 提出・発表

「作る」をそのまま管理すると大きすぎる。「見える・測れる・担当できる」単位まで砕く。

WBS をつくる5つのステップ

1. **プロジェクトをワークパッケージまで分解する。**順序づけ・資源配分・スケジュール作成・監視ができることを目安に。
2. **各ワークパッケージを定義する。**細かさは、プロジェクトの期間・複雑さに見合うものにする。
3. **引き受ける人が分かりやすい書式で示す。**成果物が明確で、それを成し遂げるのに時間がかかるものであること。
4. **すべてを完了すれば目標が達成されることを確認する（=100% ルール）。**
5. **プロジェクト・マネジメントの作業も忘れずに入れる。**報告書・会議・関係者の承認取り付けを「プロジェクト管理」というまとめタスクで持つ。

2026 年の追加

各ワークパッケージに「**誰がやるか=人/AI 支援/確認者**」と「**完了条件**」を書く。AI に渡せるのは、完了条件が言葉になっている作業だけ。

WBS の3つのルール（+今回の当てはめ）

ルール	内容	今日の実習での当てはめ
100% ルール	子要素をすべて足すと親要素と等しくなる。漏れなく・重複なく。最も重要	「プロジェクト管理」「レビュー」「引継ぎ」も入れる。範囲外（やらないこと）は書いて外す
80 時間ルール	1 日（8h）もかからない作業は小さすぎ、2 週間（80h）以上は大きすぎるので細分化する。報告は 0/50/100% で	学生の案件は「次の確認までに完了を判定できる単位」（0.5～8h 程度）に調整
7×7 ルール	1 つの親にぶら下がる子は 7 程度まで、階層は 7 レベル程度まで	今日は 2～3 層、各親 3～7 項目で十分。7 層まで掘らない

80 時間と 7×7 は目安のルール。自分のプロジェクトで適正な粒度を考えるのが課題。

AI 時代の補足

AI に渡す作業は、小さくて、完了条件が明確なもの。

粒度が細かいほど委譲しやすく、確かめやすい。一畳敷では「1 つの Issue = 1 つの AI エージェント」だった。

WBS 分解の例 — 5 つ

卒業論文／花見／一畳敷 VR／NO！トレ／真葛焼。ソフトウェアでないものも、太田の実案件も、同じ形に分解できる。

見るところ: ①マイルストーンは 4～7 個 ②末端の作業は「終わったかどうか」が自分で判断できる ③待ち時間と確認が入っている

分解の例 ①

卒業論文の執筆（9/16 → 12/15・自分 1 人・週 12 時間）

ID	マイルストーン	作業（ワークパッケージ）	完了条件	工数	待ち	担当	先行
A01	A テーマと章立て ～9/26	先行研究を 5 本ざっと読み、テーマ候補を 3 つ書く	候補 3 つが A4 1 枚に	4h	—	人	—
A02		章立て（目次）を 1 枚にする ★最初の一步	A4 1 枚の目次	3h	—	協働	A01
A03		指導教員に見せて直す	「この方向で」の返事	2h	5日	人	A02
B01	B 先行研究 ～10/10	文献を 15 本集めてリスト化（著者・年・主張）	表が 15 行	6h	—	協働	A03
B02		各文献を要約し、自分の立場との差を 1 行書く	15 本すべてに 1 行	9h	—	人	B01
C01	C 調査・実験 ～11/7	調査の設計（質問項目・手順）を書き、教員に確認	教員 OK	6h	3日	人	B02
C02		協力者 20 名を募る（依頼文・SNS・授業で声かけ）	20 名確保	4h	7日	協働	C01
C03		実施 → 集計 → 図表を 5 点作る	図表 5 点	24h	—	協働	C02
D01	D 執筆 ～12/5	第 1～2 章（序論・先行研究）6,000 字	6,000 字	18h	—	協働	B02
D02		第 3～4 章（方法・結果）8,000 字	8,000 字	22h	—	協働	C03
D03		教員レビュー 2 回と修正	「提出可」の返事	6h	各7日	人	D02
E01	E 提出・発表 12/15	体裁チェック（書式・引用・ページ番号）	チェックリスト全 OK	4h	—	AI 支援	D03
E02		提出 ◆12/15 → 発表スライドと練習	受領メール／10 分で話せる	11h	—	人	E01
Z01	Z 管理	週 1 回、実績を入れて計画を引き直す	13 回分の記録	13h	—	人	—

ここが肝

作業の合計は約 132 時間。だが待ち時間の合計は 29 日——教員の返事と協力者集め。自分が速くなくても縮まない日数を先に置くと、締切に間に合うかが分かる。

分解の例 ②

花見を企画する（サークル 30 人・4 月上旬・幹事 3 人）－ ソフトウェアでない例

ID	マイルストーン	作業	完了条件	工数	待ち	担当	先行
A01	A 企画 3/1 まで	目的と予算を決める（1 人いくらかまでか）	幹事 3 人が合意	1h	—	人	—
A02		候補日を 3 つ出し、参加者に希望を聞く	回答 20 名以上	1h	5日	協働	A01
B01	B 場所 3/10 まで	候補地を 3 か所調べる（桜の見頃・トイレ・アクセス）	比較表 1 枚	2h	—	AI 支援	A01
B02		公園の使用許可を申請する	許可証が手元に	1h	14日	人	B01
B03		下見（場所取りの範囲・電源・ゴミ捨て場を確認）	写真とメモ	3h	—	人	B01
C01	C 人集め 3/25 まで	告知（日時・場所・会費・持ち物） ※ B02 の許可が出てから	SNS と掲示で告知済み	2h	—	協働	B02
C02		参加者リストと会費の集金	30 名分の入金確認	3h	10日	人	C01
D01	D 買い出し 前日	買い物リストを作る（食べ物・飲み物・ゴミ袋・レジャーシート）	リストと概算金額	2h	—	AI 支援	C02
D02		買い出し・運搬の分担を決めて買う	レシートと現物	5h	—	人	D01
E01	E 当日 ◆花見当日	場所取り（朝 7 時・2 人）	シートを敷き終わる	6h	—	人	D02
E02		受付・会計・写真	参加者全員の受付	4h	—	人	E01
E03		片付け・ゴミの持ち帰り・原状回復	公園の人に確認してもらう	2h	—	人	E02
F01	F 後始末	会計報告と写真の共有、振り返り 1 枚	メンバーに送付	2h	—	協働	E03
Z01	Z 管理	雨天時の代替案（Plan B）を決めておく	代替日と室内案	1h	—	人	A02

ここが肝

① 許可（B02）が出る前に告知（C01）してはいけない——先行 ID がそれを止める。② 片付け（E03）を忘れるのが定番。③ 雨は「起こりうること」——次回のリスク管理で扱う。

空飛ぶ一畳敷 VR（10 日間・人 3 名+AI エージェント）

ID	マイルストーン	作業	完了条件（＝実物で確かめる）	日	担当	先行
A01	A 計画・仕様 D0	決めるべきことの質問リストを AI に作らせ、発注者に答えてもらう	12 問すべてに回答	0.5	AI → 人	—
A02		回答をもとに計画書（ゴール・決定事項・リスク）を書く	12 トラックに分かれている	0.5	協働	A01
A03		トラックごとの仕様書と Issue（1 作業＝1 Issue）	仕様 12 本・Issue 12 件	0.5	協働	A02
B01	B 基盤 D1～D2 ★最優先	Quest 3 で空のシーンが起動する（OpenXR・APK ビルド）	実機で起動を確認	1	人	A03
B02		2 名同期と位置合わせ（手動 2 点 → 自動）	2 台で同じ空間に立てる	2	人	B01
C01	C シーン制作 D1～D3	6 場面の骨組みを PC で再生できる状態にする	PC で通して再生	1	AI×6	A03
C02		各場面に実素材を入れる（3DGS・絵巻・木片）	Quest 実機で場面ごとに確認	2	AI + 人がレビュー	C01,B01
D01	D 語り・キャラ D1～D3	台本（story.json）を確定し、音声を一括生成	全台詞の音声ファイル	1	AI	A02
D02		語り部のモーション（座り・上半身）	不自然でないと人が判断	1	人	—
E01	E 統合・試験 D4～D6	10 分に連結（ホスト操作盤つき）	◆統合 v1 が通しで動く	1	協働	C02,D01,B02
E02		通し試験 1 名 → 2 名／性能調整（72fps）	◆2 名で 10 分完走	2	人	E01
F01	F 現地 D7～	◆正午フィーチャーフリーズ／最終 APK・機材梱包	以降はバグ修正のみ	1	人	E02
F02		現地設営・リハ → ◆デモ	会場の光量で位置合わせが通る	3	人	F01
Z01	Z 管理	毎朝 AI が前日の進捗を要約 → 人が当日の割当を決める	10 日分の記録	0.5/日	協働	—

ここが肝

① **B01（基盤）が最初**——これが無いとどの場面も実機で確認できない＝クリティカルパス。② 実装は AI に渡したが、**完了条件はすべて「実機で確かめる」**。③ **判断日**（技術が間に合うか）と**フリーズ日**を着手前に決めた。

NO！トレ（9 か月・企画会社+技術会社+県） — 長期・複数組織の例

ID	マイルストーン	作業	完了条件（合格の門）	期間	担当	先行
A01	A 立ち上げ 契約～6月中旬	キックオフ（役割・連絡経路・決め方を合意）	議事録に全員の合意	1週	人	—
A02		シナリオ 3 本の内容を委託者と詰める	県の監修 OK	4週	協働	A01
B01	B 教材開発 ～7月	AI が相手役になる会話の作り込み（3 シナリオ）	〔機械〕自動テスト 72 本が通る	6週	AI + 人	A02
B02		倫理のガードレール（言うてはいけないこと）	〔人〕県と読み合わせ	2週	人	A02
C01	C 端末・運営 6月中旬～7月	端末 3 台の調達・設定（初期化した状態から通しで確認）	〔人〕3 台×3 回＝9 回で確認	3週	人	—
C02		使用マニュアル作成 → 運営スタッフ研修	スタッフが 1 人で回せる	3週	協働	C01,B01
D01	D 会場交渉 7月～	県内 7 市の商業施設と交渉・日程確定	7 会場の日程が確定	8週	人	A01
D02		会場ごとの動線・電源・掲示の準備	チェックリスト全 OK	各1週	人	D01
E01	E 啓発活動 8月～12月	7 市を月 1～2 市のペースで巡回（体験 → 声かけ → アンケート）	◆12 月末に全 7 市完了	20週	人	C02,D02
F01	F 報告 1月～2月	アンケート集計・分析	集計表とグラフ	3週	AI 支援	E01
F02		業務報告書 → ◆2/26 納品／教材の運用引継ぎ	県の受領	4週	人	F01
Z01	Z 管理	版ごとに期日を引き直す（v1.1 → v1.2 → v1.3 → v2.0）	各版で〔機械〕〔人〕の門を通す	随時	人	—

ここが肝

- ① 完了条件を 〔機械〕＝毎回スクリプトで測る／〔人〕＝耳と実機でしか分からないに分けて書く。② 会場交渉（D01）は最初から動かす——相手のある作業は待ち時間が長い。③ 予備日には作業を入れない。

真葛焼 プロジェクションマッピング（展示 10/3・演出家+美術館+制作）

ID	マイルストーン	作業	完了条件	工数	担当	先行
A01	A 絵コンテ ～7月上旬	演出家の絵コンテを受け取り、 カット構成の一覧（manifest） にする	23 カットが 1 ファイルに	8h	人	—
A02		各カットの尺・動き・つなぎ方を決める	カットごとに 1 行の指示	12h	協働	A01
B01	B 3D 下地	Blender で奥行き（depth）とマスクを自動レンダリング	23 カット分が揃う	10h	自動	A02
C01	C 静止画 ～8月	色のパレットを解析し、画像生成 AI でキーフレームを作る	カットごとに 1 枚	40h	AI	B01
C02		花瓶を固定座標に合成（AI に描かせない）	全カットで花瓶の位置が同じ	12h	自動 + 人が確認	C01
C03		静止画の段階で演出家と合意する	「この絵で進めて」の返事	6h	人	C02
D01	D 動画化 ～9月中旬	image-to-video で各カットを動かす	23 本のクリップ	30h	AI	C03
D02		4K へ拡大 → つなぎ目を 数値で検証	境界の差が基準値以内	16h	自動 + 人	D01
E01	E 仕上げ ◆9月末 制作完了	組み立て・ループ化・音の同期（約 2 分・縦 4K・60fps）	通して再生できる	20h	協働	D02
E02		美術館に見てもらい、修正	◆先方 OK	8h	人	E01
F01	F 現場 ◆10/3 展示開始	実物の花瓶に合わせて投影を調整（マッピング）	形がぴったり合う	12h	人	E02
F02		常設運用の手順書（毎日の起動・停止・トラブル対応）	館の人が 1 人で回せる	6h	協働	F01
Z01	Z 管理	カット一覧（manifest）を更新し続ける	常に最新が 1 か所	随時	人	—

ここが肝

① 途中で **15 カット → 23 カットに増えた**。作業の一覧が 1 か所にあったので、組み直さずに済んだ。② **動画にする前に静止画で合意（C03）** —— 後戻りのコストが一桁違う。③ 実在する花瓶は **AI に描かせない**。人が決める線をどこに引くか。

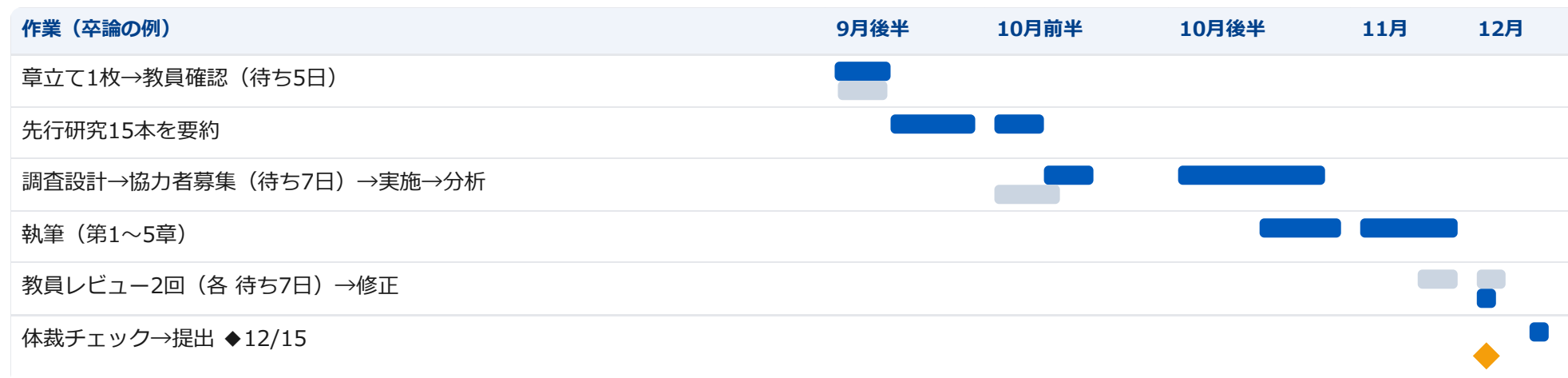
100% ルールで漏れやすい作業

漏れやすい作業	なぜ漏れるか
レビュー・確認・試作を見せる	「作る」だけを書き、「誰が見て OK と言うか」を書かない
待ち時間（返答待ち・納品待ち・審査）	作業 0 時間なので工数に出ない。でも日数はかかる
承認・許可・交渉	「始めてよい条件」を書かないと、許可前に告知するような事故が起きる
プロジェクト管理（打合せ・報告・予算）	「みんなでやる」と思って誰も書かない
片付け・引継ぎ・ドキュメント	終わったあとの作業は忘れる
AI 出力の検証	「AI が作ったから動くはず」と思い込む。今日の実習で必ず計上する

チェックの問い: ①成果物を全部作るための作業が入っているか ②同じ作業が 2 か所にないか ③「あとでやる」と先送りにした作業はないか

ガントチャート・依存関係・クリティカルパス

1910 年代、Henry Gantt が工場の作業を「タスク×時間軸」の横棒で描いたのが起源。フーバーダム、NASA、いまでも製品ロードマップに。WBS の「何をするか」に「いつ・誰が・何の後に」を加える。



- **依存関係**: 教員の承認が出るまで調査に入れない。灰色は**待ち時間**（作業 0h だが日数がかかる）
- **クリティカルパス**: 遅れると全体が遅れる連鎖 — 章立て → 承認 → 調査 → 分析 → 執筆 → レビュー → 提出。ここに余裕（バッファ）を置く
- AI で執筆が速くなっても、**教員の返答待ちは縮まない**。速くできるのは「自分の手番」だけ

PRP（Project Re-Planning・計画の再設定） — 計画は直すもの

- 企画段階で作られる計画の多くは、実現方法を見失っていたり、現実離れしている
- PRP=計画を正しく修正し、ヒト・モノ・カネ・時間を必要な分だけ盛り込んで、**成功するべくして成功する計画に作り直すこと**

手順

1. 企画段階の計画（Ⅰ）を元に、必要な箇所を修正して実行計画（Ⅱ）を作る
2. 進行中に**実績データ**（Ⅲ）を取って蓄積する
3. 終了時に Ⅰ・Ⅱ・Ⅲ を比べ、差分を明確にする → どの計画が妥当だったかを判断

効果

- 大雑把な計画を立てる習慣が是正される
- 個人レベルでも、うまくいったこと・いかなかったことが客観視できる
- 見落としていた小さな成功と失敗が顕在化する

今日の実習で

最後に「協力者の返答が 3 日遅れる」と条件を変えて、計画を引き直してもらおう。

これが PRP の初体験。一畳敷では、これを**毎朝** AI と一緒にやっていた。

太田の事例「AI で WBS を作る」－ 昔と今

昔（～2023）

- ホワイトボードと Excel で自分で分解。Backlog に手で登録
- 抜け・漏れは、痛い目に遭ってから気づく
- 計画の更新は「時間があるとき」＝ほぼ更新されない

今（2026・一畳敷 VR の実物）

- **AI が初稿**を作り、**人が判断**して直す。分解の速さは 10 倍、でも確かめる仕事は増えた
- 切った作業がそのまま **AI への依頼文**になる（1 Issue = 1 エージェント）
- 計画の更新は**毎朝**。AI が前日の進捗を要約し、人が当日の割当を決める

6 段の流れ（次ページで実物）

1. **質問** — AI に「決めるべきこと」の質問リストを作らせ、発注者が答える
2. **計画** — 回答を渡して計画書（ゴール・決定事項・作業分解・担当・リスク）を書かせる
3. **仕様** — 作業ごとに「何ができたら完了か」を書かせ、人が確認
4. **Issue** — 1 作業 = 1 Issue。チェックリストで下位タスク
5. **実行** — AI エージェントが実装、人がレビューしてマージ
6. **毎朝の再計画** — AI が要約 → 人が割当（= PRP）

学生版：①②は今日 Claude/Gemini でそのままやる（Step 0～2）。④は「表の 1 行 = AI への依頼になる粒度」。⑥は「週 1 回、実績を入れて引き直す」。**スプレッドシートが正本、AI は初稿・質問・変換の相棒。**

空飛ぶ一畳敷 — 12 に分解した作業と、人／AI の担当

ID	作業（トラック）	担当
S01	MR 導入（天の声・表門）	実装 AI
S02	時の間（武四郎・地図・絵巻）＋アート生成	実装 AI ＋ 人（画像生成）
S03	山道ライド・高風居の外観	実装 AI
S04	室内・時の転換	実装 AI
S05	五つの木片（触れる→光る→語る）	実装 AI
S06	問いと答え・帰還	実装 AI
T01	Quest 3 基盤（XR・ビルド）最優先・初日	共同開発者（人）
T02	2 名同期・位置合わせ（手動 → 自動）	共同開発者（人）
T03	3DGS が Quest で動くかの検証（go/no-go）	人＋AI
T04	語り・音声パイプライン	実装 AI
T05	キャラクター（座り・鳥・同行者）	人
T06	演出統合（10 分のタイムライン・操作盤）	計画 AI が設計 → 実装 AI

この表から読み取ってほしいこと

- 依存の要点を 1 行で書く：「T01（XR 基盤）と T06（演出の契約）が最初。各シーンは PC 単体で動く状態から着手し、T01 が入り次第 Quest で確認」
- 人が握った作業：計画・仕様・レビュー・マージ・実機確認・go/no-go の判断。AI に渡したのは「完了条件が書けた実装」
- それでも事故は起きた：同じ Issue を 2 つのエージェントが同時に進めた → 「取ったら宣言する」規約を追加。規約は失敗から生まれる

対応する Issue の形

タイトル: [s05 五つの木片](#) / 本文: 仕様書へのリンク・完了条件・チェックリスト（☐ 触れると光る ☐ 語りが始まる ☐ PC で再生 ☐ Quest で確認） / ラベル: [ready](#)（AI が取ってよい）

10 日間の日程（D0～D7） — マイルストーンと「判断日」

日	夕方までの到達点 ◆	誰が何を
8/29 D0	◆ 計画・リポジトリ・仕様 12 本・Issues	人 + 計画 AI
8/30 D1	◆ 空のシーンが Quest で起動／全シーンの骨組みが PC で再生	人: XR 基盤／AI: 骨組み生成
8/31 D2	◆ 3DGS go/no-go（判断日） ／各シーンに実資産	実装 AI ×6（PC 2 台で並列）
9/1 D3	◆ 各シーンを Quest 実機で単体確認	人: PR レビュー → マージ → 実機
9/2 D4	◆ 10 分連結（統合 v1）／2 名同期／ 自動位置合わせ go/no-go	人 + AI
9/3-9/4	◆ 通し試験 v1（1 名）→ v2（2 名）／運用手順書	人
9/5 D7	◆ 正午フィーチャーフリーズ （以降はバグ修正のみ）／梱包	人
9/6-9/8	現地設営・リハ → ◆ デモ	人

判断日（go/no-go）を先に決める

「3DGS が重ければ 360 実写に切替」「自動位置合わせが間に合わなければ手動 2 点」——**やめる条件と代替案を、着手前に書く**。これはリスク管理（9/23）。

毎日のリズム

朝: AI が前日の PR/Issue を要約 → 人が優先順位と割当を更新／日中: AI が自走／夕: 人がレビュー・マージ・実機確認。**計画の更新（PRP）が毎日回る。**

参考事例 — NO！トレ（9 か月の長期プロジェクト）：フェーズ表と「合格の門」

期間	主要作業	担当
契約～6月中旬	キックオフ／シナリオ詳細協議／大学との連携協議	企画会社＋技術会社
6月中旬～7月	端末 3 台の調達・設定／方言対応の学習開始／教材本番版	技術会社
7月	内部テスト／マニュアル／スタッフ研修／商業施設との交渉	企画＋技術
8月～12月	県内 7 市で順次啓発活動（月 1～2 市）	企画会社
1月	アンケート集計・分析／報告書	企画会社
2月	報告書提出（2/26）／教材の運用引継ぎ	全員

マイルストーン: 6月末 α版／7月末 β版＋研修／8月初旬 第1回啓発／12月末 全7市／2/26 納品。学生の案件とスケールは違うが、**型は同じ**。

改善計画（9/14 版）の「合格の門」

直し	合格の門＝何ができたら完了か
配信スクリプトは検査 NG なら配信を止める	〔機械〕 わざと壊して止まることを確認
初期化した端末で最初のひと言が出る	〔人〕 端末 3 台 × 3 回＝9 回で確認
方言対応を入れるかどうか	〔判断日〕 9/19 に go/no-go

版ごとの期日を引き直した（v1.1 9/17 → v1.2 9/22 → v1.3 9/25 → v2.0 9/28）。**9/29–30 は予備日で、作業を入れない**。「やらないと決めたもの 3 つ」も明記。

過年度の受講生が作った WBS（10 件・匿名化済み）

- 卒論作成／目に優しい速読アプリ／卒業研究／カジュアルゲームアプリ／就活（For 内定）／公務員試験合格／TOEIC 800 点／学内新聞づくり／進路決定
- それぞれに **WBS 表+ガント+リスク管理シート**。今日の実習の「ゴールの姿」です
- 各事例に太田のコメント（良い点／足りない点）を付けてあります

受講生ページ

➤ 過年度の受講生が作った WBS（参考事例）

➤ 元の Excel（匿名化済み）

<https://icu-course.pages.dev/SWD/wbs-examples/>（パスワードは授業で配布）

よくある「足りない点」（先に知っておく）

- 「指導教員の返答待ち」「協力者募集」のような**待ち時間**が工数に入っていない
- 「SPI 対策 90 人日」のように**大きすぎる作業**がそのまま（80 時間ルール）
- 担当・進捗が空欄のまま——ひとりでも「自分」と書き、進捗を更新して初めて PRP ができる
- テンプレの日付（2017 年）のまま——起点を自分の期日に合わせる

今日の WBS がこの 10 件より良くなる条件は 1 つ：「**誰が・何を見て・完了と言うか**」が書いてあること。

事例紹介 ① 配布している WBS フォーマットと、書き方のコツ

2022 年から配っている Google スプレッドシートのフォーマット（今日の提出テンプレの元）。**列の構成そのものが「何を決めるべきか」のチェックリスト**になっている。

列	書くこと	ねらい
WBS ナンバー	A01・A02…／B01…（マイルストーンごとに記号を変える）	依存関係と担当を、番号で指せるようにする
作業項目	2 階層くらい で記入（レベル2 = マイルストーン／レベル3 = ワークパッケージ）	深く掘りすぎない。7×7 ルール
工数（人日）	人日 =1 人が 1 日でできる作業量。 人時 でもよい	「誰かが何時間やるか」を数字にする
必要人数／担当	何人でやるか・誰がやるか	同じ時間に同じ人を重ねていないか分かる
進捗	% のほか「処理中／済」など状況でもよい	0/50/100% で十分（80 時間ルール）
開始予定・終了予定 ／開始実績・終了実績	上段が予定、 下段に実績 を入れる	予定と実績を比べる = PRP の材料になる
日付欄（右側）	1 日単位でも週単位でもよい。色を塗ってガントにする	依存と待ち時間が目に見える

作る順番（このシートの指示どおり）

やりたいことを考える → 達成したい目標 → **マイルストーンを 4〜7 個** → 各マイルストーンに必要な項目 → **ワークに落とす（全部で 30 個程度）** → スケジュールに落とす

太田のコツ（シートに書いてあるもの）

レイアウト: 列幅は小さく・セル結合／**Alt+Enter** で改行／色分けで進捗／日付は **10/10** 形式

記入: **1 つのセルに 2 つの作業を入れない**／具体的に（「調べる」→「競合 5 件の価格を調べて表にする」）

事例紹介 ② 昨年度（2025）の受講生 28 名は、何をプロジェクトにしたか

同じフォーマットで、28 名がそれぞれ自分のプロジェクトの WBS を作りました。テーマの一部（本人が名前を付けていたもの）：

テーマ	どんなプロジェクトか
Coin database	コインのデータベースを作る。収集・分類・検索の仕組み
Enhancing musical perception through color mapping	音楽の知覚を色で拡張する。研究とプロトタイプ
Multi-Modal BCI Game Interface with Adaptive Learning	脳波などを使うゲーム操作インターフェース。研究テーマをそのまま計画に
AR スタンプラリー作成による卒業論文執筆	制作と論文執筆を 1 つのプロジェクトとして並行管理
(ほか 24 名)	卒論・就活・試験・アプリ制作・イベント運営など。テーマ名を付けずに提出した人も多い

この 4 つから分かること

- ① 研究テーマそのものをプロジェクトにしてよい（卒論・ゼミの研究と兼ねられる）
- ② 制作と論文のように、2 つの締切があるものほど WBS が効く
- ③ 英語のテーマでも構わない（課題は日英どちらでも可）

提出のときに毎年ある失敗

共有設定を忘れる——URL を貼っても「閲覧可」にしていないと開けず、未提出扱いになる。昨年も全員分を個別に依頼し直しました。
提出したら、**シークレットウィンドウで自分の URL を開いて確かめる**のが確実です。

実習（17:30–19:20）

自分の計画を、AI と他者の力で実行可能にする

3 人組をつくって自己紹介 → 自分のプロジェクトを決める → **まず自分の頭で分解する** → AI に分解させる → 練習問題で「欠陥」を診断 → 自分の案を直す → 3 人で見直す → ガント → 提出（→ 授業後に再計画）

いきなり AI に聞かない。**最初の 7 分は、紙とペンだけ。**

自分が何を思いつき、何を思いつかなかったか——それが今日いちばんの収穫です。

3 人組をつくって、自己紹介（1 人 2 分・計 6 分）

近くの人と **3 人組**をつくってください（33 人 = 11 組）。この **3 人**で今日ずっと動きます——練習問題の診断（Step 3）も、計画の見直し（Step 5）も同じ組で。

話すこと（1 人 2 分）

1. 名前・メジャー・学年

2. 何か開発をしたことがありますか？

プログラミング・Web・ゲーム・動画・音楽・電子工作など。「ない」でも構いません

3. 何かプロジェクトをやったことがありますか？

学園祭・サークルの公演・イベント・部活の大会・研究・アルバイトの企画など。うまくいった／いかなかった話も

4. AI を使って、どんなことをしていますか？

レポート・翻訳・プログラミング・画像・相談相手。使っていない人は「使っていない」で OK

5. 今日、自分のプロジェクトにしたいこと（まだ決まっていなければ候補でよい）

なぜ最初にやるか

- ① このあと **自分の計画を人に説明して、質問してもらう**。相手のことを知らないと質問が出ない
- ② **経験も AI の使い方も人によって全然違う**。得意な人に聞けるようになる
- ③ 最終回のワークショップは **3 人チームで制作**。今日が最初の顔合わせ

聞く側のコツ

「へえ」で終わらず、**ひとつ質問する**。

「それは何人でやったの？」「いちばん大変だったところは？」「AI に任せてみて、うまくいかなかったことは？」

時間が余ったら、4 の「AI の使い方」を掘ってください。今日の実習で、その差がそのまま出ます。

実習の全体像（110 分・途中で 10 分休憩）

配分	Step	やること	残すもの
6 分	—	3 人組をつくって自己紹介 （開発経験・プロジェクト経験・AI の使い方・今日のテーマ候補）	今日ずっと動く 3 人組
8 分	0	目的・対象者・成果物・範囲外・期限・使える時間・ 最初の 1 週間の最小の一步（MVP） を 1 枚に書く	前提と完了条件
7 分	1	まず自分の頭で分解する （紙・AI なし）。思いつく作業を全部書き、個数を数える。書き切れなくてよい	自分の初稿
10 分	2	同じ前提を Claude（または Gemini）に渡し、WBS の初稿（プロンプト①②）。 手案はまだ渡さない	AI 初稿と入力文
10 分	3	配布の「欠陥入り練習 WBS」を 3 人で診断（5 つの問い）	不足・矛盾と影響
10 分	休憩		
15 分	4	手案と AI 案を比べて修正。「手案だけ／AI だけ／両方ない」を各 3 つ記録。100%・粒度・依存・担当の根拠をチェック	修正版
15 分	5	3 人組で相互レビュー（1 人 5 分）	指摘メモ
20 分	6	主要 8～10 作業に依存・担当・工数・待ち時間を入れ、プロンプト③で Mermaid gantt → mermaid.live で描画	ガント
10 分	7	Google スプレッドシート（テンプレ）に貼って提出	提出
課題	8	条件を変えて再計画する（PRP） ——授業後に自分でやる。次回 9/23 と課題1 で使う	延期・削減・並行化した内容

Claude が止まったら Gemini に同じ文を貼る。両方止まったら配布の「保存済み AI 初稿」で Step4 から続ける。AI を使うこと自体は必須ではない。

手順書：自分のプロジェクトの WBS を作る

はじめに（6 分）3 人組をつくって自己紹介

名前・メジャー・学年／開発をしたことがあるか／プロジェクトをやったことがあるか／AI をどう使っているか／今日のテーマ候補。この 3 人で今日ずっと動きます。

Step 0（8 分）前提を 1 枚に書く

目的／対象者（誰のため）／成果物／範囲外（やらないこと）／期限／使える時間（週 h）／最初の 1 週間の最小の一步。WBS が 30 個は出そうな題材を。

Step 1（7 分）まず自分の頭で分解する（AI なし・紙）

思いつく作業を全部書く。数を数える。書き切れなくてよい。ここを飛ばすと、今日の実習は意味がなくなります。

Step 2（10 分）AI に初稿を作らせる

claude.ai を開く → プロンプト①（質問に答える）→ プロンプト②（WBS の表）。手案はまだ渡さない。出力はコピーして保存。

Step 3（10 分）練習問題を 3 人で診断

配布「欠陥入り練習 WBS」に 5 つの問いを当てる。答えは後で開く。

Step 4（15 分）比べて直す

手案だけにあった／AI だけにあった／両方になかった作業を各 3 つメモ。チェック: 100%（管理・レビュー・待ち時間が入っているか）／粒度（0.5～8h）／先行 ID が循環しない／「AI 支援」の行にも人の責任者。

Step 5（15 分）3 人で相互レビュー

1 人 5 分。自分の計画を説明 → 他の 2 人が質問「無理はないか」「足りないものは」「AI に任せすぎでは」。

Step 6（20 分）ガント

主要 8～10 作業に依存・担当・工数・待ち時間 → プロンプト③ → 出た Mermaid を mermaid.live（登録不要）に貼る → PNG 保存（任意）。

Step 7（10 分）提出

WBS テンプレ（Google スプレッドシート）をコピーし、1 枚目に表、2 枚目に Mermaid テキスト、3 枚目に手分解メモ + 差分。「リンクを知っている全員が閲覧可」にして、共有シートの自分の行に URL を貼る。

[提出先スプレッドシート](#)

Step 8（課題・授業後）条件を変えて再計画する

「協力者の返答が 3 日遅れる」「使える時間が週 2 時間減る」。何を延期・削減・並行化したか、重要な変更を 3 つ書く。最初の一步（MVP）は守る。次回 9/23 と課題1 で使います。

困ったとき

Claude 上限 → Gemini に同じ文を貼る／両方ダメ → 配布「保存済み AI 初稿」で Step 4 へ／ネットが不安定 → 紙で Step 4～7 をやり、後日入力。

配布物

[欠陥入り練習 WBS](#)

[保存済み AI 初稿（卒論の例）](#)

[WBS テンプレ（xlsx）](#)

[mermaid.live](#)

プロンプト① — 前提を対話で詰める（Step 2 の前半）

あなたは経験豊富なプロジェクトマネージャーです。私のプロジェクトを、WBS（作業分解）にできるまで具体化してください。

プロジェクト：「（1行で）」

期限：「」 成果物：「」 対象者（誰のため）：「」

使える時間：「週 ○ 時間」 範囲外（やらないこと）：「」

最初の1週間の最小の一步：「」

足りない情報があれば、5 問以内で一つずつ質問してください。

私が全部答えたら、「目的／対象者／成果物／期限／制約（時間・お金・スキル）／範囲外／最初の一步」を 10 行以内でまとめてください。

分からないことは推測せず、「未確認」と書いてください。

ポイント: **AI に質問させる**のが今風。自分でも気づいていない前提（誰が承認するのか、いつまでに何を見せるのか）が出てくる。一置敷の「キックオフ質問 12 問」と同じ発想。

プロンプト② — WBS の初稿（Step 2 の後半）

上のまとめをもとに、WBS を作ってください。条件：

- レベル1=フェーズ（「プロジェクト管理」を必ず含める）、レベル2=成果物、レベル3=作業。合計 30~50 件
- 各行に：ID、親ID、成果物／作業、完了条件（何ができたら終わりか）、担当する人、AI が支援できる部分、確認者、人の工数（時間）、待ち時間（日）、先行ID
- 最下位の作業は 8 時間以内。AI が支援する行は「生成 oh + 人の検証 oh」に分ける
- 返答待ち・納品待ち・審査などの「待ち時間」は、作業とは別の行にする
- 出力は Markdown の表。最後に「見落としやすい作業」を 5 つと、「仮定したこと」「未確認のこと」を箇条書きで

AI の WBS にありがちな穴（Step 4 で探す）

レビュー・利用者確認がない／承認や交渉がない／待ち時間が 0／粒度が不均一／依存を無視して同じ人に同時刻の作業／「AI がやる」行に人の責任者がいない

禁止事項

個人情報・他人の名前・機密をプロンプトに入れない（「指導教員」「協力者 A」でよい）。

Mermaid とは — 文字で書くと図になる記法（無料・登録不要）

- **Mermaid** = テキストで図を書くためのオープンソースの記法。フローチャート・ガントチャート・シーケンス図などが書ける
- AI はこの記法を知っているので、「Mermaid の gantt で」と頼めば**図をテキストで出してくれる**。画像を作らせるより速くて、無料枠でも止まらない
- 描画は **mermaid.live**（ブラウザ・登録不要）に貼るだけ。PNG/SVG で保存できる
- 同じテキストを GitHub の Markdown や Notion に貼っても図になる。**テキストなので直しやすく、提出・共有しやすい**

行	意味
gantt	ガントチャートを描く宣言
dateFormat YYYY-MM-DD	日付の書き方
section 研究計画	見出し（WBS のレベル 1）
章立て1枚 :a1, 2026-09-17, 3d	作業名：ID, 開始日, 日数
教員確認（待ち） :a2, after a1, 6d	after = 先行タスクの後に始める（依存関係）
提出 :milestone, m1, 2026-12-15, 0d	マイルストーン（◆）

```
gantt
  title 卒業論文（例）
  dateFormat YYYY-MM-DD
  section 研究計画
  章立て1枚           :a1, 2026-09-17, 3d
  教員確認（待ち）   :a2, after a1, 6d
  先行研究15本       :a3, after a1, 14d
  section 執筆
  第1～2章           :b1, after a3, 18d
  section 提出
  提出               :milestone, m1, 2026-12-15, 0d
```

↑ これをそのまま mermaid.live に貼ると、下のような横棒の図になる。



➤ [Mermaid gantt の公式リファレンス](#)

プロンプト③ — Mermaid gantt に変換 (Step 6)

上の WBS のうち主要な作業 8~10 件を、Mermaid の gantt に変換してください。

- dateFormat YYYY-MM-DD、開始日 2026-09-17、期限 「」
- section=レベル1、タスク=作業。先行タスクは after を使う
- 待ち時間は別タスクにし、名前の末尾に（待ち）と書く
- マイルストーンを 3 つ (milestone) : 最初の一步、中間、提出
- コードブロックだけを出力してください

```
gantt
  title 卒業論文（例）
  dateFormat YYYY-MM-DD
  section 研究計画
  章立て1枚          :a1, 2026-09-17, 3d
  教員確認（待ち）   :a2, after a1, 6d
  先行研究15本       :a3, after a1, 14d
  section 執筆
  第1〜2章           :b1, after a3, 18d
  提出               :milestone, m1, 2026-12-15, 0d
```

出てきたコードを **mermaid.live** に貼ると図になる（登録不要）。PNG/SVG で保存できる。同じテキストは GitHub や Notion にも貼れる。

描画が成功しても計画が正しいとは限らない。**WBS=何を揃えるか／依存=どの順か／ガント=いつ**、の 3 つを区別する。ガントは今日の提出物では任意。

Step 4 — 手案と AI 案を比べて、自分の計画に直す（15 分）

記録する（3 枚目のシートへ）

- 手案だけにあった作業（AI が見落とした。たいてい、あなたしか知らない事情）× 3
- AI 案だけにあった作業（自分が見落とした。たいてい、管理・確認・待ち）× 3
- 両方になくて足した作業 × 3

この「差分」が今日の評価対象。AI の表をそのまま出しても点にならない。

チェックリスト

観点	問い
100%	プロジェクト管理・レビュー・待ち時間・片付けが入っているか。範囲外が混ざっていないか
粒度	1 行が 0.5～8h か。「開発 40h」は分割。「メール 5 分」は親に吸収
依存	先行 ID が循環していないか。「すべて」は依存を書いていないのと同じ
担当	「AI 支援」の行にも人の責任者。AI に任せる根拠は「完了条件が明確・定型・低リスク」か
時間	「依頼文を書く 30 分」と「返答を待つ 3 日」を分けたか
一歩	最初の 1 週間の最小の一歩が、表の最初の方にあるか

Step 7 — 提出（10 分）と、今日の評価の見方

提出物（Google スプレッドシート 1 本）

- 1 枚目「WBS」：テンプレの列（ID・親ID・成果物/作業・完了条件・担当する人・AI 支援内容・確認者・人の工数・待ち時間・先行ID・開始・終了）
- 2 枚目「Mermaid」：プロンプト③の出力テキスト（図の PNG は任意）
- 3 枚目「手分解メモ」：Step 1 の写真かメモ、Step 4 の差分 3+3+3、Step 7 の変更 3 件

提出先（共有スプレッドシート）

docs.google.com/spreadsheets/d/1kbV29tDhY94VxteGnGc6owFN-YeZ0enoVvnZjyb4pJE

手順: ① 上のシートを開く（ICU の Google アカウントで）② **テンプレのシートを自分のドライブにコピー**（ファイル＞コピーを作成）
③ 自分の WBS を書く ④ **「リンクを知っている全員が閲覧可」** に設定 ⑤ 共有シートの自分の行に、氏名・学籍番号・URL を貼る

学籍番号の行が無い人は、いちばん下に追加してください。URL が開けない設定のままだと「未提出」になります。

今日の提出で見るところ（課題1 の土台）

観点	見る
100% ルール	管理・レビュー・検証・待ち時間が入っているか
粒度・依存	0.5～8h、循環なし、クリティカルパスが言えるか
人/AI の分担	AI に任せる根拠。人の責任者がいるか
検証の記録	差分 3+3+3 と、再計画の変更 3 件が書いてあるか

行数の多さ、AI の使用量、図の美しさは**評価しない**。「自分で判断した跡」を評価する。

Step 8【課題】条件を変えて再計画する（PRP の初体験）

この Step は**授業では行いません**。提出したあと、次回 9/23 までに自分でやってください（課題1 の一部）。

条件変更（どちらか、または両方）

- 協力者（教員・依頼先・仲間）の返答が **3 日遅れる**
- 使える時間が **週 2 時間減る**

やること

1. クリティカルパス上の作業に印をつける
2. 何を**延期・削減・並行化**するか決める。**最初の一步（MVP）は守る**
3. 重要な変更 3 件と理由を 3 枚目のシートに書く（提出済みのシートに追記してよい）

これが PRP

計画は「立てて終わり」ではなく、条件が変わるたびに引き直すもの。

一畳敷では毎朝、NO！トレでは版ごとに引き直した。みなさんは**週 1 回**、実績を入れて引き直す—次回 9/23 からその習慣を始める。

次回 9/23 につながる

次回はリスク管理。「協力者の返答が遅れる」のような**起こりうることを**洗い出して、対策を計画に組み込みます。今日の再計画はその予行演習です。

次回まで（授業外学習 225 分の目安）と課題1

次回 9/23（水・祝日ですが授業あり）：リスク管理

- **WBS の見直し（60 分）**：今日の WBS を 1 人で読み直し、「想定していなかった作業」を 3 つ足す。実績（今週やったこと）を入れる
- **環境（未了の人・15 分）**：claude.ai の無料登録と gemini.google.com を使える状態に。Claude は Settings > Capabilities で「Code execution and file creation」を ON
- **予習（30 分）**：自分の WBS を見ながら「うまくいかないとしたら何が原因か」を 5 つ書き出す → 次回のリスク登録票の材料
- 次回は、今日の WBS に**リスク登録票**を足し、計画を引き直します。今日のスプレッドシートをそのまま使うので消さないこと

課題1「プロジェクト計画書」－ 出題: 今日 / 締切: 10/7（水）24:00（9/23 の 2 週間後）・Moodle

1. 前提（Step 0 の 1 枚・最初の一步を含む）
2. WBS（担当＝人／AI 支援／確認者、完了条件、待ち時間つき）
3. ガント（Mermaid テキストか、スプレッドシートの日付列）
4. **リスク登録票**（9/23 の授業で作る）と、それを反映した計画の修正
5. **「最初の一步」を実際にやった記録**（写真・メール・1 枚など、やった証拠）
6. AI 利用ログ: どこで AI を使い、どこを自分で直したか（差分 3+3+3）

形式: 今日と同じスプレッドシート 1 本にまとめる（新しく作らず、今日提出したものを育てる）。シート名は「WBS／Mermaid／手分解メモ／リスク登録票／最初の一步」。別途 PDF などは不要。日英どちらでも。

今日の提出（Step 8）→ 9/23 でリスクを足す → 10/7 までに提出、の 3 段階。今日の分だけで完成させなくてよい。

コメントシート（19:20-19:30・授業時間内に Moodle へ）

1. **自己紹介:** メジャー・学年／プログラミング経験（なし・少し・ある）／AI をどう使っているか
2. **つくりたいもの・プロジェクトにしたいもの**（1 つ以上。なぜそれか。今日の実習で選んだものと違っててもよい）
3. 今日、**AI が見落としたこと／自分が直したことを 1 つずつ**
4. 質問・要望（次回答えます）

共有したくない部分は //共有なし 文章 // で囲んでください。コメントは次回シェアし、質問には次回答えます。

欠席した人は「欠席レポート」（A4 2 枚・9/23 24:00 まで・Moodle の別提出先）。今日の実習の WBS を添えてください。

出席は Moodle の出席機能でも登録してください（コメントシートとは別）。

参考リンク

ツール（無料）

[↗ claude.ai](#)[↗ gemini.google.com](#)[↗ mermaid.live（Mermaid を描画）](#)[↗ Mermaid gantt 記法](#)[↗ Claude Artifacts の使い方（公式）](#)[↗ Gemini 無料枠の上限（公式）](#)

データ（2025-2026）

[↗ DORA 2025 Report](#)[↗ Stack Overflow: AI trust gap](#)[↗ Veracode 2026 GenAI Code Security](#)[↗ DORA 2026 ROI（InfoQ）](#)

AI への作業分解・委譲

[↗ Anthropic: human-agent teams](#)[↗ GitHub Spec Kit（specify→plan→tasks）](#)[↗ PMI: AI in project work 標準（2026-06）](#)[↗ arXiv: Manager Agent](#)[↗ arXiv: Delegation cues](#)

PM の古典・MVP

[↗ WBS（Wikipedia）](#)[↗ Gantt chart](#)[↗ Brooks's law](#)[↗ PMBOK 第8版の要点](#)[↗ Wrike: PM の基本（日本語）](#)[↗ MVP（Wikipedia）](#)

名言の出典

[↗ Brooks 1975](#)[↗ Hamilton \(WIRED 2015\)](#)[↗ Hopper 1986](#)[↗ Dijkstra EWD249](#)[↗ Thompson 1984](#)[↗ Karpathy 2023](#)

まとめ — 今日の 7 つ

	今日やったこと	持ち帰ってほしいこと
1	この授業の位置づけ	題材はソフトウェアだが、学ぶのは「 つくりたいものを、他の人と協力して実現する 」方法論。卒論・就活・学園祭・事業でも手順は同じ
2	「プログラミングは2割」	残り 8 割は、何を・誰のために・どう作り、どう確かめ、どう届けるか。AI で実装が速くなった分、 確かめる仕事と決める仕事が増えた （DORA・Stack Overflow・Veracode）
3	まず自分の頭で分解した	AI に聞く前に自分で考える。 自分にしか書けない作業 （自分の事情・関係者・締切）と、 自分が忘れがちな作業 （管理・レビュー・待ち時間）の両方が見えた
4	WBS の 3 つのルール	100% （漏れなく・重複なく。管理とレビューも入れる）／ 粒度 （次の確認までに完了を判定できる単位）／ 7×7 （人が扱える広さ・深さ）
5	順序と時間を入れた	依存関係・クリティカルパス・ 待ち時間 （作業 0 時間でも日数はかかる）。AI で速くできるのは 自分の手番だけ
6	AI に分解させて、自分で直した	AI は 初稿・質問・変換 の相棒。 判断と責任は自分 。「AI 支援」の行にも人の確認者を置く。見積りは「生成 + 検証」
7	条件を変えて引き直した（PRP）	計画は「立てて終わり」ではない。 条件が変わるたびに引き直す 。削るときも 最初の一步（MVP）は守る

今日の問い（持ち帰ってください）

自分の計画のうち、自分にしか決められないことは何か。