令和7年度

「地方やデジタル分野における専修学校理系転換等推進事業」

教員研修プログラム資料

工業専門課程 IT デジタルクリエイター科のカリキュラム開発

目 次

モジュール 1：カリキュラム開発と教材設計

1. オリエンテーションと目標設定 …… 1
2. カリキュラム設計基礎 …… 7
3. 教材開発の方法 …… 16
4. アクティブラーニング型授業の設計 …… 32
5. 教材評価と改善 …… 47

モジュール 2：インターフェースデザイン /UI デザイン

1.UI デザインの基礎理論 …… 55
2.UI 設計ツールの活用 …… 63
3. モバイルと Web の UI の違い …… 69
4. 使いやすさテスト …… 76
5.UI 演習 …… 81

モジュール 3：インタラクション & UX デザイン

1. インタラクションデザイン概論 …… 88
2.UX デザインのフレームワーク …… 96
3. ワイヤーフレーム演習 …… 102
4. ユーザーリサーチの方法 …… 108
5.UX 評価と改善 …… 113

モジュール 4：モーション & マルチメディアデザイン

1. モーションデザインの基礎 …… 120
2. アニメーションツール体験 …… 127
3.Web/ アプリにおける動きの活用 …… 133
4.SNS 動画・広告のモーション要素 …… 141
5. 実践演習 …… 149

モジュール 5：Web・SNS・紙からデジタルへの移行

1.Web デザイン基礎 …… 154
2. オンライン広告とバナー制作 …… 162
3.SNS ビジュアル制作 …… 168
4. 紙媒体からのデジタル移行 …… 176
5. 実践演習 …… 182

モジュール 6：先端デザイン領域（AI・AR/VR）

1.AI とデザイン自動化の現状 …… 188
2.AI 活用ワークショップ …… 197
4.AR/VR ツール紹介 …… 203

確認テスト　問題 …… 225
確認テスト　解答 …… 313

教員研修プログラム

モジュール2：インターフェースデザイン/UIデザイン

1.UIデザインの基礎理論



なぜUI理論が教員に必要なのか

学習と校務の効率を上げる

UIの良し悪しは、生徒の迷いや誤操作、教員の疲労に直結します。使いやすい設計は時間を節約し、本質的な教育活動に集中できる環境を作ります。

日常業務に直結する知識

- デジタル教材の設計
- LMSの画面構成
- 校務用フォームの改善
- 学校Webサイトの使いやすさ

本日の到達目標

UI原則の言語化

感覚的な「使いにくい」を具体的な原則に基づいて説明できるようになります。問題点を明確に指摘できる力を身につけます。

色の戦略的設計

色を装飾ではなく情報として扱い、意味と優先度を明確に伝える配色ができるようになります。

文字の階層設計

可読性を確保しながら、情報の重要度を視覚的に整理できる文字設計のスキルを習得します。

原則は"チェックリスト化”すると武器になる

感覚的な判断

「なんとなく使いにくい」では再現も改善もできません。

原則による評価

具体的な基準があれば、改善・評価・共有が可能になります。

実践的な武器

6つの原則を使いこなして、使いやすさを設計します。

本日扱う6原則：見える化・一貫性・フィードバック・認知負荷・エラー予防・既知の形



原則① 見える化（Visibility）

いま「どこ」「何ができる」「次」が分かる

利用者が現在地を把握し、次に何をすべきかを迷わず理解できる状態を作ります。情報を隠さず、必要なものを適切に表示することが重要です。



教育現場での具体例

- 課題の提出状態が一目で分かる
- 学習の進捗が視覚的に表示される
- 次のステップが明確に示される
- 現在のページ位置が分かる



原則② 一貫性（Consistency）

言葉の一貫性

同じ機能には同じ用語を使います。「提出」「送信」「登録」が混在すると、利用者は迷います。

見た目の一貫性

同じ役割のボタンは同じデザインに。色・形・サイズが揃っていると、学習コストが下がります。

位置の一貫性

保存ボタンはいつも右下、戻るボタンは左上など、位置を固定すると直感的に操作できます。

原則③ フィードバック（Feedback）

成功・失敗・処理中を即時に返す

成功のフィードバック

「保存しました」「提出完了」など、アクションが成功したことを明確に伝えます。

処理中の表示

読み込み中やアップロード中など、システムが動いていることを示します。

エラーの通知

何が問題で、どう修正すれば良いかを具体的に説明します。

☐ **フィードバックがないと：**保存できたか不明→二重クリック→重複送信→不安と混乱が発生



原則④ 認知負荷を下げる

読ませるより、見て判断させる

情報の段階化

全てを一度に見せず、必要な時に必要な情報を表示します。詳細は折りたたみ、概要だけを最初に見せる工夫が有効です。

選択肢の削減

選択肢が多すぎると決定疲れを起こします。本当に必要なものだけに絞り込みます。

01

複雑な操作を分割

一度に全部を求めず、ステップごとに進める設計

02

視覚的な手がかり

アイコン・色・配置で直感的に理解できる工夫

03

説明の最適化

必要最小限の文字で、明確に伝える

原則⑤ エラー予防と回復

起きない設計（予防）

- 必須項目を事前に明示する
- 入力例を常に表示する
- 不可能な操作を非表示にする
- 確認画面で最終チェックを促す



戻れる設計（回復）

- 「戻る」ボタンを必ず配置
- 「取り消し」機能を提供
- 削除前に確認ダイアログを表示
- 自動保存で作業を保護



原則⑥ 既知の形

押せる・入力・移動が見た目で分かる

ボタンらしいボタン	入力欄らしい入力欄	リンクらしいリンク
立体感や色で「押せる」と分かるデザインにします。平面的すぎると、ボタンと認識されません。	枠線やラベルで「ここに入力する」と明確に示します。背景と区別できることが重要です。	色と下線で「クリックできる」と伝えます。見た目が通常テキストと同じでは気づかれません。

利用者は過去の経験から、特定の見た目に特定の機能を期待します。その期待を裏切らないデザインが使いやすさの基本です。

色彩の基本

色は"装飾"ではなく"情報"

①状態を示す

成功は緑、注意は黄色、危険は赤。色で瞬時に状態を伝えます。

②階層を作る

重要な情報は強い色、補足情報は弱い色で優先度を表現します。

③視線を誘導

アクセントカラーで、見てほしい場所に注意を向けます。

基本の配色ルール

色数を絞ることが成功の鍵です。**ベースカラー（背景）＋文字色＋アクセントカラー＋状態色（成功・注意・危険）**の組み合わせで、ほとんどのUIを構成できます。





配色で事故りやすいポイント

学校現場でよくある失敗例

1

カラフルすぎて重要が不明

「楽しく」「華やか」を意識しすぎると、何が重要か分からなくなります。色の洪水状態では、本当に見てほしい情報が埋もれてしまいます。

2

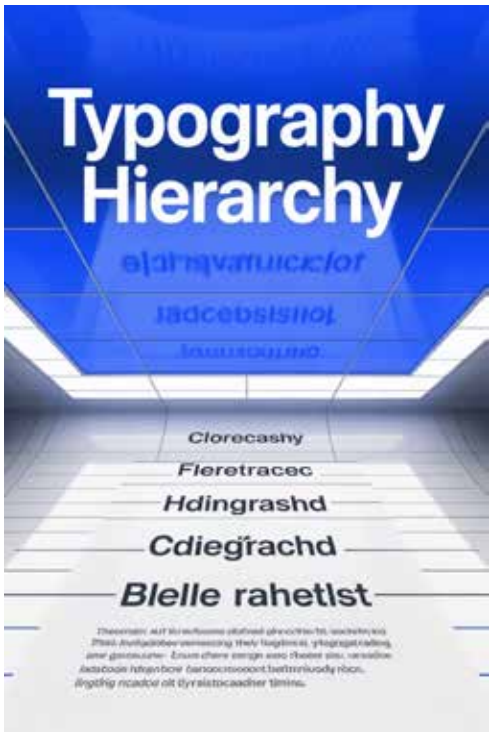
コントラスト不足で読めない

薄いグレーの文字、淡い背景色との組み合わせ。視力の弱い生徒や高齢の教員には特に厳しい設計です。

3

色だけで区別

「赤は未提出、緑は提出済み」と色だけで表現すると、色弱の方には区別できません。形・アイコン・テキストも併用します。



タイポグラフィの基本

読みやすさを支える3点セット

①階層の明確化

見出し・本文・注釈の差をサイズと太さで表現します。H1は大きく太く、注釈は小さく軽く。視覚的な段差が理解を助けます。

②可読性の確保

行間は文字サイズの1.5〜1.8倍、1行は30〜40文字程度が読みやすい目安です。詰め込みすぎると疲れます。

③一貫性の維持

同じ役割の文字には同じスタイルを適用します。見出しは常に同じフォント・サイズ・色で統一します。

文字の即改善ルール

教材・UI共通で今日から使えるテクニック

強調を乱発しない	長文は分解する	役割ごとに固定
太字・色・下線を多用すると、全てが目立たなくなります。本当に重要な1〜2箇所だけ強調します。	段落を短く、箇条書きを活用します。視覚的な余白が理解を深めます。	見出し・本文・注釈のスタイルを決めたら、全ページで統一します。
80%	65%	50%
可読性の向上	理解度の向上	作業時間の短縮
適切な行間と文字サイズで読みやすさが大幅に改善	階層化された文字情報で内容把握が容易に	一貫したスタイルで必要な情報を素早く発見

まとめ

原則・色・文字＝"使いやすさの根拠"

今日学んだ6つのUI原則と色彩・文字の知識は、感覚ではなく理論に基づいた改善の武器です。
これらを使えば、教材やシステムの問題点を具体的に指摘し、解決策を提案できます。

退出チケット（理解確認）

以下の質問に短文で回答してください：

- 一貫性が崩れると何が起きる？
- フィードバックが弱い画面の問題点は？
- 色を状態に使う例を1つ挙げてください
- 強調乱発をどう整理しますか？
- 次回のテストで観察したい点は？

教員研修プログラム

モジュール2：インターフェースデザイン/UIデザイン

2.UI設計ツールの活用

到達目標

この1時間でできるようになること

Frame・Layerを理解

Figmaの基本構造でUIを組み立てる

Stylesで統一

色・文字スタイルでデザインを一貫させる

Componentsで再利用

部品化で効率的な制作を実現

Auto Layoutで構築

崩れにくいレイアウトを作成

Prototypeで共有

提出・レビューができる状態に

次の「使いやすさテスト」「UI演習」へつなぐ実践的なスキルを習得します

今日の提出物

成果物の定義

この3つを完成させることで
UI制作の基礎が身につきます



01

1画面UI（Frame）

完成した画面レイアウト

02

画面遷移1本（Prototype）

動作する画面遷移

03

共有リンク

閲覧またはコメント可能なリンク

提出＝「授業化」への第一歩評価・改善のサイクルが始まります



Figmaの基本概念

UI制作の骨格を理解する

Frame

画面の器（Mobile / Web）
すべてのUIはここから始まります

Layer

部品の基本単位
テキスト・ボタン・画像など

Group / Section

整理のための機能
複数のレイヤーをまとめる

命名ルールの例btn/primary、card/base など
共同編集しやすい名前をつけましょう

作業の流れ

UI制作の標準手順



1. Frame作成
   画面サイズを決定
2. グリッド設定
   余白の基準を決める
3. Styles作成
   文字・色を定義
4. 部品化
   Componentsを作る
5. Auto Layout
   崩れを防止
6. Prototype
   遷移を設定
7. 共有
   コメントで改善

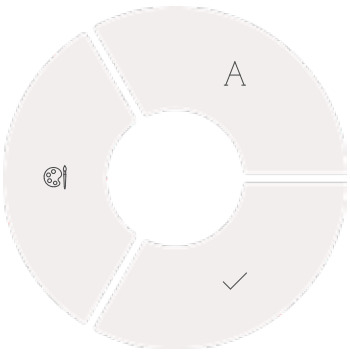
Stylesで統一を作る

色・タイポグラフィの管理

Color Styles

主色・アクセント・警告などを定義

- ブランドカラー
- 背景色
- 警告・エラー色



Text Styles

見出し・本文・注釈の階層

- H1〜H3（見出し）
- Body（本文）
- Caption（注釈）

統一の効果

学習者の迷いが減る

- 教材が一貫する
- 修正が楽になる
- プロの印象に

ルール例フォントサイズは3〜5段階に絞ると使いやすくなります

Componentsで教材テンプレになる

部品化のメリット

- 修正が一括反映
  1箇所直せば全体に適用
- 授業で配布しやすい
  学生に再利用してもらえる
- 制作スピード向上
  毎回作り直す必要なし

最小セット例

btn/primary 主要アクション
btn/secondary 副次的アクション
input/text テキスト入力欄
card/basic 基本カード





状態設計

UI基礎理論の実装ポイント

ボタンの状態	入力欄の状態
通常 デフォルトの見た目	未入力 プレースホルダー表示
押下 タップ時の反応	入力中 フォーカス状態
無効 使えない状態の表現	エラー 警告メッセージ付き

> **教材指導ポイント**状態を作る＝使いやすさを作る
> 文言は短く、行動が明確に（例：送信／確認／戻る）

Auto Layout
崩れないUIの最重要スキル
Auto Layoutの目的
文字量や要素追加でもレイアウトが
崩れない仕組みを作ります



教員向けアドバイス見た目より先に「構造」を
教えると生徒の理解が深まります

実演例：カードの構築

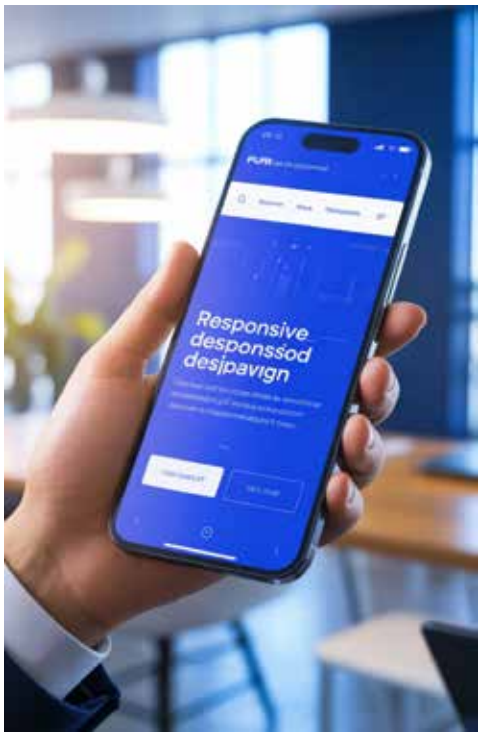
タイトル＋説明＋ボタンで構成



1
余白（Padding）
要素の内側の余白を設定

2
間隔（Spacing）
要素間の距離を管理

3
揃え（Alignment）
左右・上下の揃え方を指定



Constraints
画面サイズ変化に耐える設定

Mobile / Webの違い
各デバイスに最適化した表示を準備します

固定要素
ヘッダーなど位置を固定したい要素の設定

追従要素
ボタン幅など画面幅に合わせて変化

画像トリミング
アスペクト比を保ちながら調整





レスポンシブデザインの入口として理解し
次回以降のより深い学習に接続します



Prototype＆共有

テストとフィードバックの準備

遷移設定
タップで画面A→画面Bへ動作させる

プレビュー確認
実際の動作で体験をテスト

共有リンク作成
閲覧・コメント権限を設定

プレビューで発見

- ユーザーの迷いポイント
- 誤タップの可能性
- 遷移のスムーズさ
- 文言の分かりやすさ

共有でフィードバック

- 閲覧リンクで確認
- コメント権限で改善提案
- 次回の材料を収集
- チーム内での議論

> コメント＝次回の「使いやすさテスト」の貴重な材料になります

評価と次回予告
超ミニ・ルーブリック（5観点×各1点）



Frame整理
画面構成の明確さ



Style統一
色・文字の一貫性



部品化
Componentsの活用



Auto Layout
崩れない構造



遷移＋共有
Prototype完成度

次回の内容

使いやすさテスト
観察から改善点を抽出

UI演習
簡易アプリ画面の設計と共有



教員研修プログラム

モジュール2：インターフェースデザイン/UIデザイン

3.モバイルとWebのUIの違い

1時間で達成するゴール

前提条件の違いを説明できる

モバイルとWebの利用環境の違いを、具体的な根拠とともに説明できるようになります。

UI作り分け理由を言語化できる

「なぜこの配置にするのか」を、教員として学習者に説明できる状態を目指します。

レスポンシブの基本と実例を理解

実際のWebサイトや教材での適用方法を、実例を通じて学びます。

教材改善チェックができる

自校の教材ページや学校サイトを改善する具体的な観点を持ち帰れます。

UI設計の起点は「文脈（コンテキスト）」

UIとは何か

UIデザインは見た目だけではありません。**操作の設計＋情報の設計**が本質です。使う場所、時間、目的によって最適解は大きく変わります。

設計の出発点

授業設計と同じく、「誰が・いつ・どこで・何をするか」を明確にすることが最優先です。利用者の行動を起点に考える習慣が、良いUI設計につながります。



モバイル vs Web：利用環境の根本的な違い

モバイル環境

- 移動中や隙間時間での利用
- 短時間で完結させたい
- 通知や電話など割り込みが多い
- 片手操作や立ったままが多い

最短で目的達成できる導線設計が必須

Web環境（PC）

- 机に座って落ち着いて利用
- じっくり集中できる時間がある
- 情報の比較や参照がしやすい
- 複数のタブやウィンドウを開ける

俯瞰性と詳細性を両立した設計が可能

モバイルは「途中で止められる」前提で設計します。だから一画面で理解できる構造、戻りやすい導線
迷わせないシンプルな選択肢が重要になります。



操作の違い：タッチ vs マウスの特性

タッチ操作の課題

- 指で画面が隠れる
- 誤タップが起きやすい
- ホバー効果が使えない

マウス操作の利点

- 精密なクリックが可能
- ホバーで補助情報を表示
- 小さなリンクでも扱える

モバイル設計の鉄則

- タップ領域は44px以上
- 要素間に十分な余白
- 重要ボタンは固定配置

> Webの「ホバーで説明表示」や「小さなリンク」は、モバイルでは完全に破綻します。教材ページの提出ボタンやリンクが小さいと、学習者が誤操作してしまうリスクが高まります。

情報密度の違い：同時に見せられる量

モバイルの特性

縦スクロール前提で、一度に表示できる情報が限られます。そのため「優先順位の削ぎ落とし」が設計の核心になります。

- 重要情報を最上部に配置
- 補足情報は下層に展開
- 一画面一メッセージが理想



Webの特性

俯瞰性が高く、複数の情報を並べて比較することができます。サイドバーやタブで情報を構造化できます。

- 複数カラムで並列表示
- 関連情報を同時に参照可能
- 詳細な表やグラフも扱える



モバイルでは「全部見せる」ことができません。だから重要情報を先に、補足情報を後に置く設計が必要です。
この考え方は、授業資料の見せ方にも直結します。



導線設計の違い：ナビゲーション

モバイルのナビゲーション

ボトムナビゲーション：親指で届く下部に主要機能を配置

ハンバーガーメニュー：省スペースで多くの項目を格納

固定CTA：重要なアクションボタンを常に表示

Webのナビゲーション

グローバルナビゲーション：上部に主要カテゴリを常時表示

パンくずリスト：階層構造を可視化して現在地を明示

サイドバー：補助的な導線や関連情報を配置

学習者が「今どこにいるのか？次にどこへ行けばいいのか？」を常に把握できることが最優先です。
教材サイトでは特に、戻り方と次の行動が明確でないと離脱率が上がります。

入力の違い：フォーム設計（教育現場で特に重要）

01

モバイルでの入力は苦痛

小さな画面でのキーボード入力は時間がかかりミスも起きやすい。
選択式や自動補完を活用することが重要です。

02

Webでは比較的入力可能

物理キーボードがあるため、長文入力や複雑なフォームにも対応できます。
ただし、それでも簡潔さは重要です。

03

小テスト・アンケートの設計

モバイル前提なら「**入力最小化**」が基本原則。
自由記述を多用すると回答率が大幅に落ちます。

推奨される入力形式

- 選択式（ラジオボタン・チェックボックス）
- スライダーやステッパー（数値入力）
- ドロップダウンメニュー
- 段階評価（星評価など）



まとめ：違いの本質は「前提条件」にある

画面サイズ
表示領域の物理的な制約

利用文脈
どこで、いつ、どのように使うか

入力環境
フォームや回答のしやすさ

操作方法
タッチとマウスの特性の違い

導線設計
ナビゲーションの構造と配置

情報密度
同時に見せられる情報量

これらの違いを踏まえた上で、同一コンテンツを破綻させずに提供する設計手法が**レスポンシブデザイン**です。
次のセクションで詳しく学びます。

レスポンシブデザインとは何か

よくある誤解

レスポンシブは単なる「縮小技術」ではありません。画面を小さくすればいいという発想では、使いづらいUIができてしまいます。

正しい理解

レスポンシブは「優先順位に合わせてレイアウトを組み替える設計」です。読みやすさ、操作しやすさ、目的達成を守ることが本質です。

教材・学校サイトにも必須

学習者の多くがスマートフォンでアクセスします。モバイル対応が不十分だと、学習体験そのものが損なわれます。

情報の再構成が鍵

デバイスごとに「何を先に見せるか」「どう操作させるか」を再設計することで、すべての環境で快適な体験を提供できます。





基本①：モバイルファーストの考え方

小さい画面から設計を始める

制約が厳しいモバイル画面で設計すると、本当に必要な情報だけが残ります。装飾や余計な要素は自然に削ぎ落とされます。

重要情報→補足情報の順に整理

限られたスペースの中で優先順位を明確にする訓練になります。「何を最初に見せるべきか」が自然と明確になります。

教材も「結論→詳細」の構造に

授業資料や教材ページも、この考え方が有効です。学習者はまず結論を知りたがります。詳細は後から追加すればいいのです。

モバイルファーストは、情報設計そのものの質を高める方法論です。
教材制作でも「本当に伝えたいこと」を明確にする訓練として活用できます。

基本②：ブレークポイントとレイアウトの切替

画面サイズに応じて、**列数や配置を動的に変える**ことで、すべてのデバイスで読みやすさと操作性を守ります。単に幅が広がったら余白が増える、というだけではありません。

モバイル：1列レイアウト

縦スクロールで一つずつ情報を提示。誤タップを防ぐために余白を十分に確保します。

タブレット：2列レイアウト

画面が広がったら、関連する情報を横に並べて比較しやすくします。

デスクトップ：3列レイアウト

大画面では、より多くの情報を同時に見せて俯瞰性を高めます。サイドバーも活用できます。

> 特に表、一覧、メニューなどの構造的なコンテンツは、ブレークポイントによる切り替えの影響を大きく受けます。目的に応じた見せ方を選択することが重要です。

実例A：学校案内（紙→Web）のレスポンシブ化

PC版の設計方針

俯瞰性を活かす：学科の比較や、複数の導線を並列表示することで、訪問者が自分で情報を探しやすくします。

- 学科一覧を横並びで比較
- トップページに主要情報を集約
- サイドバーで補助的なリンクを配置



モバイル版の設計方針

行動を先に：資料請求ボタンを最上部に固定し学科一覧は縦スクロールで展開。背景情報は後から見られるようにします。

- 固定CTAで資料請求を促進
- 重要情報を上から順に配置
- 詳細ページへの導線を明確に



実例B：時間割・カリキュラム表（表が崩れる問題）

表はモバイルで最も崩れやすい形式です。目的が「比較」なのか「確認」なのかによって、最適な解決策は変わります。

1

PC：表で俯瞰が最適

デスクトップでは、時間割全体を一覧で表示し、曜日と時限を交差させた表形式が最も分かりやすい形式です。

2

カード化（科目ごとに分割）

各科目を独立したカードとして縦に並べます。検索や絞り込みと相性が良く、詳細情報も追加しやすい形式です。

3

絞り込み機能（学年・曜日で表示）

ドロップダウンやタブで条件を選択し、該当する時間割だけを表示します。目的の情報に素早くアクセスできます。

4

横スクロール（最終手段）

表の構造をそのまま残し、横にスクロールできるようにします。直感的ですが、操作性は劣るため最終手段として検討します。

教材設計でも頻出の論点です。表を使う前に「本当に表が必要か？」を問い直すことでよりシンプルで効果的な見せ方が見つかることがあります。

教員研修プログラム

モジュール2：インターフェースデザイン/UIデザイン

4.使いやすさテスト



今日できるようになること

01

実施

最小の手順でユーザビリティテストを実施できるようになります

02

記録

観察した事実を体系的に記録できるようになります

03

優先順位

改善の優先順位を決められるようになります

受講後は、観察を記録し、改善の優先順位を決められるようになります。最後に「改善TOP3」を作って提出できれば、このレッスンは合格です。学校現場やプロダクトチームで、すぐに実践できる型を習得しましょう。

成果物：記録シート＋改善TOP3を作成します

ユーザビリティテストとは



感想会ではなく行動観察

ユーザビリティテストは、感想を集める時間ではありません。ユーザーの行動を観察し、つまずきの原因を特定して改善するための手法です。

「使いにくい」ではなく、**「どこで・何が起きたか」を事実で記録します**。たとえば、10秒止まった、戻るを2回押した、欠席の言葉を探して連絡を開いた、などです。

止まる 操作が中断し、ユーザーが次の行動を迷っている状態	往復 前の画面に戻る、同じ箇所を繰り返し見る動作
誤操作 意図しない場所をタップする、間違ったボタンを押す	発話 「あれ？」「どこだろう」などの迷いを示す言葉

事実が残ると、次に直すべき箇所が明確になり、授業でのフィードバックも"批評"ではなく"改善の材料"になります。これがユーザビリティテストの本質です。



観察の共通言語　4つの観点で整理する

観察を授業にするには、共通言語が必要です。ここでは4つの観点で整理します。つまずきを見つけたら、この4観点のどれに当てはまるか分類してください。分類できれば、改善の方向性が定まります。

見つけられる
情報やボタンに視線が届くか。配置や視認性の問題を観察します。

理解できる
言葉やアイコンの意味が伝わるか。認知負荷の問題を観察します。

迷わず進める
導線や順序が自然か。タスクフローの問題を観察します。

戻れる
間違っても回復できるか。エラー回復の問題を観察します。



レッスン全体の流れ

短く回す型で成果を出す

このレッスンでは、現場で回る"短い型"を扱います。ユーザビリティテストは長いほど良いわけではありません。
授業や研修では、**短いサイクルで回すほど効果が出ます**。

準備
役割分担と最小セットを整える

実施
ユーザーの行動を観察する

記録
事実を体系的に記録する

改善
優先順位を決めて改善する

今日は、準備→実施→記録→改善の順に進め、最後に改善TOP3を決めます。
まずはこの型をコピーできることを目標にしてください。1タスクでも十分に成立します。

準備：役割分担と最小セット



必要な要素は最小に

学校現場で再現できるよう、必要な要素は最小にします。授業に導入する場合は
役割カードを用意すると運用が安定します。ここまで整えば、テストは始められます。

進行役
タスクを提示し、質問をする役割です。
誘導しないことが重要です。

ユーザー役
実際に操作し、思考を声に出す役割です。
自然な振る舞いを心がけます。

観察者
行動を記録し、つまずきを記録する役割です。
事実に集中します。

準備する3点

- **テスト対象**：Figmaプロトタイプや画面の画像
- **タスク**：目的型で書かれた具体的なシナリオ
- **記録シート**：観察事実を記録するテンプレート

ダウンロード教材：観察記録シート（PDF/Word）をご用意しています

成否を分ける3つのルール

ここが最重要です

誘導しない	思考発話
教えた瞬間に観察が壊れます。沈黙は迷いの証拠です。助けないでください。	ユーザー役は考えを声に出します。「今、何を探しているか」が重要です。

観察優先

感想や意見は最後にまとめて聞きます。テスト中は行動観察に集中します。



NGフレーズ

- 「そこじゃなくて」
- 「普通こっち」
- 「正解は…」

これらはすべて観察を壊します。テスト中に沈黙があっても、助けないでください。沈黙は迷いの証拠であり、改善のヒントです。数秒待ってから「今、何を探していますか？」と質問することで、ユーザーの思考を引き出すことができます。

タスク設計：目的型で書く

テストの質はタスクで決まる

タスクは"目的"で書いてください。ボタンを指定すると答えを渡してしまいます。



❌ ボタン指定

「欠席連絡ボタンを押してください」

これでは観察になりません

⭕ 目的達成

「今日の欠席連絡を先生に送ってください」

迷いが自然に現れます

欠席連絡	締切確認	連絡事項
今日の欠席連絡を先生に送る	提出物の締切を確認する	明日の連絡事項を探す

生徒作品でも同様に、"この機能で何を達成するか"をタスクにしてください。実際の生活シーンで書くと、迷いが自然に現れます。



実演：進行台本

そのまま読んで使える

進行は台本化すると安定します。開始前に必ず言うのは2つ。実施中は誘導しない質問に限定します。
止まっても助けず、数秒待ってから質問してください。

1 開始前
「これはあなたの能力テストではなく画面のテストです」「思ったことは声に出してください」

2 実施中の質問
「今、何を探していますか」「次に何が起こると思いますか」「その操作を選んだ理由は」

3 終了時
「一番迷ったところ」「言葉で引っかかった箇所」を短く確認します

推奨：3分のデモ動画（良い進行例）と1分のNG例動画（誘導してしまう例）を参考にしてください

記録と改善　事実から優先順位へ

記録の3ステップ

事実を書く
秒数・往復・誤操作・発話を記録

4観点で分類
見つけられる/理解できる/迷わず進める/戻れる

重大度を付ける
0=好み、1=迷うが達成、2=多くが迷う、3=達成不能

改善の優先順位

改善は全部やりません。**TOP3で十分です**。優先順位は「影響×頻度×修正容易性」で決めます。

1 **影響が大きい**

2 **頻度が高い**

3 **直しやすい**

改善あテンプレート

問題（事実）	原因仮説
改善案	成功条件

成功条件は次のテストで確認します。ここまでやって初めて学習サイクルになります。

授業では、ミニテスト→記録→改善TOP3→次回再テスト。この循環を回すことで、生徒は"作って終わり"から"検証して改善する"へ成長します。

最終提出：改善TOP3（観点／重大度／改善／成功条件）を入力して修了証を取得しましょう

教員研修プログラム

モジュール2：インターフェースデザイン/UIデザイン

5.UI演習



本日のUI演習：ゴール

この演習では、1時間で「簡易アプリ画面」を設計し、共有していただきます。実践を通じて、UI設計の基礎を体験的に学びます。

成果物

6画面＋2遷移プロトタイプ＋共有リンク

制作時間

60分で完成を目指します

次のステップ

使いやすさテストにそのまま使用します

演習の位置づけ

モジュール2で学んだ理論を実践に接続

01

UI基礎理論

原則・色・タイポグラフィを実際の画面に反映させます

02

ツール操作

Figma / Adobe XDの操作を制作を通じて定着させます

03

デバイス特性

モバイルとWebの違いをモバイル前提設計で体験します

04

テスト準備

使いやすさテストに使用可能な画面一式を完成させます





演習課題

学校向け「欠席連絡」ミニアプリのUIを設計せよ

対象ユーザー

保護者の方（スマートフォン使用）

目的

欠席連絡を**1分以内**に完了できるようにする

条件

初めて使う人でも迷わずに操作できること

要件（絶対条件）

以下の5つの条件をすべて満たす必要があります。これらは評価の基準となります。

1 画面数
最低6画面を設計してください

2 ナビゲーション
下部タブ・戻るボタン・メニューなどのナビゲーション要素を配置します

3 入力フォーム
欠席理由・日付などの入力フォームは必須です

4 送信完了画面
送信完了を示す画面を作成してください

5 プロトタイプ
最低2つの画面遷移を接続します

設計ブリーフ

最初の5分で埋めましょう

想定ユーザー
保護者（片手操作が多い、朝の時間帯に使用）

ユーザーの困りごと
朝の忙しい時間帯に、入力が面倒で時間がかかる

成功条件
迷いゼロ・入力最小・確認が簡単

主要タスク
①欠席連絡の入力 ②送信完了の確認

トーン＆マナー
安心感・学校らしさ・読みやすさを重視

画面構成　6画面テンプレート

以下の構成を基本として、ユーザーの行動フローを意識した画面設計を行います。

1. ログイン/本人確認
   または初回案内画面
2. ホーム
   今日の情報＋「欠席連絡」CTA
3. 欠席連絡フォーム
   入力画面
4. 入力エラー状態
   未入力など1つ以上
5. 確認画面
   送信前チェック
6. 送信完了
   次にやること導線



UI設計の必須ルール

学んだ原則を画面に反映させる

1画面1目的

各画面には明確な単一の目的を持たせます

主要CTAは1つ強く

他のボタンは控えめにして、階層を作ります

視覚階層

見出し→本文→ボタンの明確な階層を構築します

一貫性

ボタン形状・余白・ラベル名を統一します

色の役割

主要・注意・背景など、役割で色を決めます

モバイル前提の注意点

スマートフォンでの使用を前提とした設計で、ユーザビリティを最大化します。



- 片手操作

  重要なボタンは画面下部に配置し、親指で届きやすくします

- タップ領域

  小さすぎないサイズで、押し間違いを防止します

- 文字の読みやすさ

  小さすぎない文字サイズ、十分な行間を確保します

- 入力負担の軽減

  プルダウン・ラジオボタンなど選択式を優先します

- 導線の明確さ

  「戻る」ボタンで迷子にならない設計を心がけます

制作手順　Figma / Adobe XD 共通プロセス

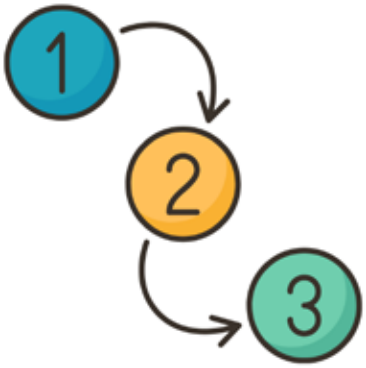
フレーム作成
スマートフォンサイズのフレームを設定します（例：375×812px）

コンポーネント作成
ボタン・入力欄・カードなどの再利用可能な部品を先に作ります

画面配置
6画面を複製して並べます

テキスト階層統一
見出し・本文・注釈のスタイルを全画面で統一します

プロトタイプ接続
2つ以上の画面遷移を接続します



セルフチェック

提出前に5分間チェックしましょう

1

発見しやすさ

3秒で「欠席連絡」が見つかりますか？

2

入力の最小化

入力項目が最小限になっていますか？

3

エラー案内

エラー時の案内が分かりやすく表示されていますか？

4

送信前確認

送信前に入力内容を確認できますか？

5

一貫性

全画面でUIが統一されていますか？

評価基準　ルーブリック（簡易版）

以下の5つの観点から評価を行います。各項目を意識して制作を進めてください。

目的の明確さ

ユーザーが迷わず操作できる設計になっているか

導線

開始→入力→確認→完了の流れが自然か

視覚階層

見出し・重要・補助の区別が明確か

一貫性

部品・余白・文言が統一されているか

読みやすさ

文字・コントラスト・情報量が適切か



提出物・共有方法

提出物

- ①共有リンク
- ②ホーム画面のスクリーンショット
- ③設計ブリーフ（1枚）

共有設定

リンクを知っている人が閲覧可能に設定してください

ファイル名

例：UI演習_欠席連絡_氏名

提出後は、次回の使いやすさテストで実際にユーザーに触ってもらい
フィードバックを得ます。完成度を高めて提出しましょう！

令和７年度「地方やデジタル分野における専修学校理系転換等推進事業」
工業専門課程 IT デジタルクリエイター科のカリキュラム開発

教員研修プログラム資料

令和８年２月

学校法人第一平田学園 中国デザイン専門学校
〒700-0842 岡山県岡山市北区船頭町 12
電話:086-225-0791

●本書の内容を無断で転記、掲載することは禁じます。