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

今日のゴール（到達目標）

インタラクションを理解する

「やりとり」として具体的に説明できるようになります。人とシステムの対話を明確に捉えることが第一歩です。



失敗原因を診断できる

原則に基づいて問題点を見抜く力を身につけます。なぜ使いにくいのかを体系的に分析できるようになります。



改善案を提案できる

文言・導線・反応の3つの観点から具体的な改善策を考えられるようになります。実践的なスキルを習得します。



なぜ今"対話設計"が必要か

学校現場のリアル

使いにくいシステムは、生徒の学習意欲を低下させ、教員の業務負担を増大させます。日々の小さなストレスが積み重なることで、デジタルツールへの抵抗感が生まれてしまいます。

- 課題提出システムで「送れたか分からない」不安
- 欠席連絡フォームで何度も同じ情報を入力
- 予約システムで操作ミスによる二重予約
- アンケートで回答途中のデータが消失



これらの問題は、適切な対話設計によって解決できます。

用語整理：UI／UX／インタラクション

1	2	3
UI（ユーザーインターフェース）	UX（ユーザーエクスペリエンス）	インタラクション
見た目と操作面のデザインです。ボタンの配置、色、文字サイズなど、画面上で目に見える全ての要素を指します。	全体的な使用体験です。操作前の期待から操作後の満足感まで、ユーザーが感じる総合的な印象を含みます。	行動と反応の連鎖です。ユーザーの操作に対してシステムがどう応答し、次の行動を促すかという対話の流れを設計します。

インタラクションの基本モデル

人の行動

ユーザーがボタンをクリック、フォームに入力など、何らかの操作を行います。

システム反応

システムが応答を返し、結果を表示します。この反応が次の行動を導きます。

次の行動

反応を受けて、ユーザーは次に何をすべきかを理解し、行動を続けます。

重要な原則：ユーザーは反応から学びます

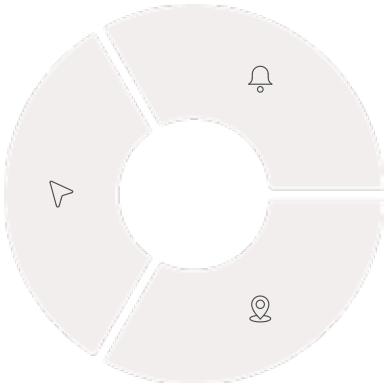
適切なフィードバックがあることで、ユーザーは
自然に正しい使い方を習得していきます。



対話設計で見るべき3点セット

①操作（Action）

ユーザーが何をするのか。クリック、入力、選択など、具体的な行動を明確にします。



②反応（Feedback）

システムがどう応答するか。成功・失敗・進行中など、状態を伝えるメッセージや視覚的変化を設計します。

③状態（State）

今どこにいるのか。何が起きたのか。現在地と変化を常に明示し、ユーザーの不安を解消します。

この3つが揃って初めて、スムーズな対話が成立します。

原則① 可視性

今どこにいて、次に何ができるかを明確に

ユーザーが迷わないよう、現在の状態と可能な操作を視覚的に示すことが重要です。情報を探す手間を最小限にすることで、スムーズな操作を実現します。

学校現場での具体例

- 課題の進捗状況（未着手／作業中／提出済み）
- 提出期限の残り時間表示
- 完了・未完了の視覚的な区別
- 未提出者リストの見やすい表示





原則② フィードバック

操作後の反応を明確に

クリック後に何が起きたのかを必ず伝えます。「送信しました」「保存しました」「処理中です」など、状態を明示することで、ユーザーの不安を解消します。

エラーは解決策とセットで

単に「エラーが発生しました」ではなく、「メールアドレスの形式が正しくありません。@マークを含めて入力してください」のように、原因と具体的な対処法を示します。

タイミングも重要：即座のフィードバックがユーザーの信頼を生みます。処理に時間がかかる場合は、進行状況を示すインジケーターを表示しましょう。

原則③ 一貫性

用語の統一

同じ機能には同じ言葉を使います。「送信」「提出」「登録」が混在すると混乱の原因になります。

配置の統一

重要なボタンは常に同じ位置に。ユーザーが迷わず操作できるよう、レイアウトのパターンを保ちます。

色の統一

赤は警告、緑は完了など、色の意味を一貫させます。場面によって意味が変わらないようにします。

操作の統一

似た機能は似た操作方法で。学習したパターンを他の場面でも活用できるようにします。

一貫性があることで、ユーザーは一度学んだことを繰り返し活用でき、使うほど楽になっていきます。

原則④ 誤操作耐性



事故を"起きない設計"にする

人は必ずミスをします。重要なのは、ミスが致命的にならないよう、システム側で備えることです。

具体的な対策

- **取り消し機能：**誤って削除してもすぐに復元できる
- **戻るボタン：**前の画面に簡単に戻れる
- **下書き保存：**途中の作業が自動で保存される
- **確認ダイアログ：**重要な操作前に再確認を促す

特に学校では、生徒も教員も時間に追われて操作することが多いため、誤操作への配慮が特に重要です。

原則⑤ 負荷の最小化

記憶より認知、入力より選択

人の記憶は不確実です。覚えることを要求するのではなく
見て選べるようにすることで、操作の負担を大幅に減らせます。

選択肢を提示

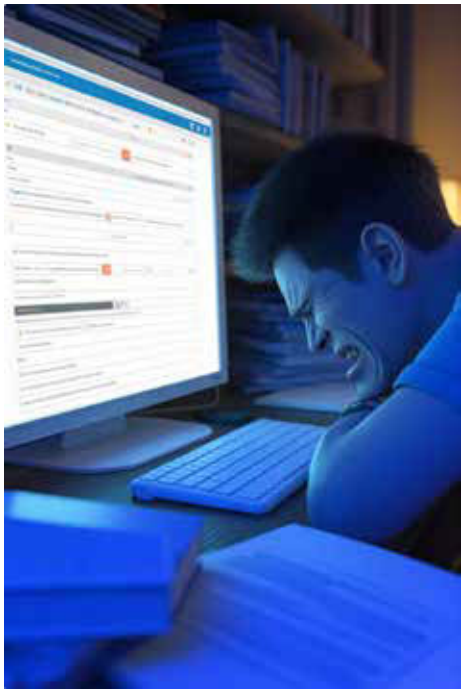
自由入力ではなく、プルダウンやチェックボックスで選べるようにします。

例文・テンプレート

文章を一から書かせるのではなく、ひな形を用意して編集してもらいます。

オート補完

入力途中で候補を表示し、すぐに選択できるようにします。



ケース①：提出フォームの事故例

よくある失敗パターン（Before）

「送信ボタンを押したけど、本当に送れたのか分からない…」

「どの項目が必須なのか分からなくて、エラーが出てから初めて気づいた」

「エラーメッセージが出たけど、何が悪いのか全然分からない」

これらは設計の問題であり、ユーザーの責任ではありません。原則を適用することで解決できます。

ケース①：原則で診断する

どの原則が欠けているか分析しましょう

1

可視性の問題

必須項目のマークが目立たない。進捗状況が見えない。完了状態が曖昧。どこまで入力すれば良いか分からない状態です。

2

フィードバックの問題

送信後の完了通知が弱い。処理中なのか完了したのか判断できない。ユーザーは不安なまま放置されます。

3

誤操作耐性の問題

下書き保存がない。再送信の導線がない。一度失敗すると最初からやり直しになり、ユーザーの負担が大きくなります。

複数の原則が同時に欠けていることが多いため、体系的に診断することが重要です。

改善の型：3観点で考える

01

文言を改善する

何をすれば良いか一文で明確に伝えます。「この情報を入力してください」ではなく、「氏名、学年、クラスを入力して送信ボタンを押してください」のように具体的に。

02

導線を改善する

次の行動を1つに絞ります。選択肢が多すぎると迷うので、「今すべきこと」を明確にし、余計なボタンは隠すか後回しにします。

03

反応を改善する

完了・保存・エラーの見える化を徹底します。色や位置、アニメーションを使って、状態変化を一目で分かるようにします。

この3つの観点で整理することで、具体的で実行可能な改善案が出せるようになります。



演習：1画面改善ワーク

個人ワーク → 共有

実際に手を動かして、改善案を考えてみましょう。以下のテンプレートを使って整理すると、考えやすくなります。

題材を選ぶ

- 課題提出フォーム
- 欠席連絡システム
- 施設予約画面
- アンケートフォーム

改善テンプレート

1. **つまずきポイント：**どこで困るか？
2. **欠けている原則：**5原則のどれか？
3. **改善案（3つ）：**
   - 文言の改善
   - 導線の改善
   - 反応の改善

完成したら、隣の方とアイデアを共有してみましょう。



まとめ＋次回予告

今日学んだこと

ミクロの視点：対話・反応・状態

今日は、1つ1つの操作とフィードバックに焦点を当て、インタラクションの基本を学びました。5つの原則を使えば、具体的な改善ポイントを見つけられます。

- 可視性
- フィードバック
- 一貫性
- 誤操作耐性
- 負荷の最小化

次回の内容

マクロの視点：体験全体の設計

次のコマでは、視野を広げて、ユーザーの体験全体を設計する方法を学びます。UXピラミッドやカスタマージャーニーマップを使い、始まりから終わりまでの流れを最適化していきます。

今日学んだミクロの原則と、次回のマクロの視点を組み合わせることで、本当に使いやすいシステムを設計できるようになります。

教員研修プログラム

モジュール3：インタラクション & UXデザイン

2.UXデザインのフレームワーク

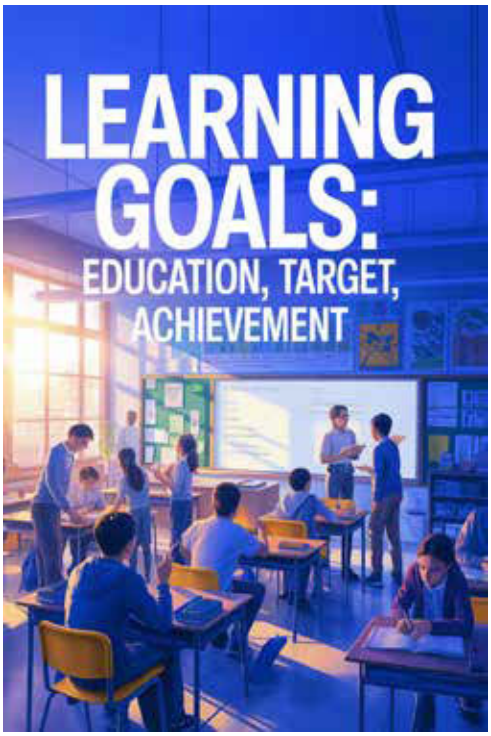
本日のゴール

UIとUXの違いを説明できる
見た目と体験の違いを理解し、教育現場で使い分けられるようになります

UXピラミッドで課題の階層を整理できる
問題の原因が「価値」「使いやすさ」「見た目」のどこにあるか診断します

カスタマージャーニーで体験の時系列を可視化できる
利用前から利用後まで、体験の流れと痛点を把握します

改善を仮説として立て、次回の評価へつなげる
検証可能な改善案を作り、実践的なUXサイクルを回します



UIとUXの違い

UI（ユーザーインターフェース）

画面上の**見た目**や**操作部品**を指します。ボタンの配置、色の選択、文言の表現など、目に見える要素すべてがUIです。

- ボタンの大きさや位置
- 色使いやフォント
- アイコンやラベル

UX（ユーザーエクスペリエンス）

利用前から利用後までの**体験全体**を指します。不安を感じる、迷う、納得する、達成感を得るといった感情の流れがUXです。

- 使う前の期待と不安
- 操作中の迷いや気づき
- 完了後の満足感や学び

教育現場での具体例

LMSでの課題提出、学校ホームページでの情報検索、イベント申込システム、校務支援ツールなど
日常的に触れるすべてのシステムにUIとUXが存在します。



なぜ教育でUXが効くのか

学習の摩擦を減らす

「使える教材」や「わかりやすい仕組み」は、学習者が本来の学びに集中できる環境を作ります。操作に迷う時間を減らせば、その分だけ学習時間が増えます。

学習意欲への影響

迷いや不安は学習意欲を大きく削ります。提出方法がわからない、再提出の仕方が不明確、といった小さなストレスが積み重なると、離脱や提出ミスにつながります。

運用コストの削減

UX改善は「学習成果」だけでなく「運用コスト」にも効果があります。問い合わせの減少、教員の説明時間の短縮、トラブル対応の削減など、運営側の負担も軽減されます。

2つのフレームワークで体験を可視化する



UXピラミッド

体験を**階層**で診断し、優先順位をつけます

問題の根本原因が「価値」にあるのか「見た目」にあるのかを見極め、効果的な改善策を立てられます。

カスタマージャーニー

体験を**時系列**で把握し、痛点を特定します

利用前の不安から利用後の振り返りまで、どこで躓くのかを明らかにし、的確なタイミングで改善を打てます。



2つを組み合わせることで、「どこで」（ジャーニー）「何が」（ピラミッド）問題なのかが明確になり、検証可能な改善仮説を立てられます。

UXピラミッドとは



01

上ほど本質、下ほど表層

ピラミッドの上層は「なぜ使うのか」という本質的な価値を、下層は「見た目の印象」といった表層的な要素を表します。

02

見た目から直す前に本質を確認

デザインを美しくする前に、「そもそも価値があるか」「使いやすいか」を確認することが重要です。表層だけ直しても根本的な問題は解決しません。

03

課題の原因を階層で説明

「この問題は価値の階層の課題です」「使いやすさに問題があります」といった形で、課題の原因をどの階層の問題かで説明できるようになります。

UXピラミッドの5階層



1	価値（Value） 使う理由があるか。目的達成や安心感を得られるか。そもそもこのシステムやツールは必要なのか。
2	有用性（Usefulness） 必要な機能があるか。やりたいことを実現できる機能が揃っているか。
3	使いやすさ（Usability） 迷わず操作できるか。直感的に理解でき、スムーズに目的を達成できるか。
4	信頼性・アクセシビリティ 壊れない、見える、誰でも使える。端末による差、読みやすさ、障壁のなさ。
5	見た目（Aesthetics） 印象、統一感、読みやすさ。視覚的に魅力的で、情報が整理されているか。

教育現場に当てはめる視点

価値の視点

「このシステムで何が**できるようになる**のか」が明確に伝わっているか。学習者や教員にとっての目的達成が見えているか。

使いやすさの視点

課題の提出、確認、再提出のプロセスで迷いが生じないか。操作手順が直感的で、エラー時のフィードバックが適切か。

信頼性の視点

端末による表示の差、通信環境への配慮、文字サイズの調整可能性、色覚への配慮など、誰でも安心して使えるか。

見た目の視点

重要な情報に強弱がついているか。余白が適切で読みやすいか。情報の優先順位が視覚的に表現されているか。





カスタマージャーニーとは

画面単体ではなく「前後を含む体験」を見る

1つの画面だけを改善しても、その前後でユーザーが困っていれば意味がありません。利用前の準備から利用後の振り返りまで、一連の流れで体験を捉えます。

行動＋接点＋感情＋痛点で「体験の詰まり」が見える

何をしているか（行動）、どこで触れているか（接点）、どう感じているか（感情）、何に困っているか（痛点）を整理すると、改善すべきポイントが浮かび上がります。

改善は時系列のどこに打つかが決まる

ジャーニーマップを使えば、「準備段階で不安を解消する」「実行中の迷いを減らす」など、改善のタイミングと内容が明確になります。

カスタマージャーニーの基本型

認知
存在を知る

準備
使う準備をする

実行
実際に使う

完了
目的を達成する

振り返り
結果を確認する

各フェーズで確認すべき観点

定性的な視点

- **行動**：ユーザーは何をしているか
- **接点**：どのツールや場所で触れているか
- **気持ち**：どんな感情を抱いているか
- **痛点**：何に困っているか、どこで躓くか

定量的な視点

- **改善機会**：どこに介入すれば効果的か
- **指標**：完了率、所要時間、問い合わせ件数、再提出回数など

授業で実際に活用する際は、この基本型をベースに、対象とするシステムや教材に合わせてカスタマイズしましょう。

2つのフレームワークで改善仮説を立てる

ジャーニーで痛点の場所を特定

カスタマージャーニーを使って、体験のどのタイミングで問題が起きているかを明確にします。

ピラミッドで原因階層を特定

UXピラミッドを使って、その問題が「価値」「使いやすさ」「見た目」のどの階層に起因するかを診断します。

改善を検証可能な仮説にする

「こう変えたら、こうなるはず」という形で、測定可能な仮説を立てます。

仮説テンプレート

> もし**（変更内容）**をしたら、**（測定指標）**が**（期待する変化）**する。なぜなら**（痛点・理由）**だから。

例：もし提出ボタンの上に「提出前の確認リスト」を表示したら、再提出率が20%減少する。
なぜなら、学生は提出前に必須項目を見落としていることが多いから。

まとめ＆次回への接続

今日の要点

- UIとUXの違いを理解する
  見た目と体験は異なる概念です
- ピラミッドで優先順位を
  問題の原因がどの階層にあるかを診断します
- ジャーニーで全体像を
  時系列で体験の流れと痛点を把握します
- 2つを組み合わせて仮説を
  検証可能な改善案を立てます

次回やること



テーマ：UX評価と改善サイクル

- 仮説をユーザーテストで検証する方法
- 改善サイクルを回す実践プロセス
- ルーブリック的に評価指標を整える（提出ミス率、完了率、所要時間など）

今日学んだフレームワークを実際の評価と改善に活かしていきます。

教員研修プログラム

モジュール3：インタラクション & UXデザイン

3.ワイヤーフレーム演習

本時のゴール

ユーザーとタスクの明確化

ユーザーと主要タスクを1つに絞り込みます

画面遷移への分解

タスクを論理的な画面遷移に分解します

ワイヤーフレーム作成

3〜6画面の低忠実度ワイヤーを作成できるようになります

UX評価への接続

次のユーザーテストに繋がる成果物を作ります

なぜワイヤーフレームが必要か



構造と流れを優先

見た目のデザインより先に、情報構造とユーザーの流れを確定させます

コスト削減

実装・制作コストをかける前に、設計のズレや問題点を早期発見できます

合意形成の加速

教員間や学内調整での説明がスムーズになり、意思決定が速くなります



ワイヤーフレームの定義

低忠実度のワイヤーフレームは、四角・線・文字だけで十分です。デザインの詳細は後回しにし、構造に集中しましょう。

01

ユーザーと主要タスク

誰が何を達成したいのかを明確に定義します

02

画面遷移フロー

タスクを完了するまでの画面の流れを設計します

03

各画面の要素配置

画面ごとに必要な要素とその配置を決定します

モデル化の題材

学校現場で使える3つのパターン

A. 欠席・遅刻連絡

保護者から担任への連絡をスムーズにするアプリ機能

- 朝の忙しい時間でも素早く入力
- 電話不要で確実に伝達
- 履歴の確認が可能

B. 課題提出とフィードバック

生徒と教員の双方向コミュニケーション機能

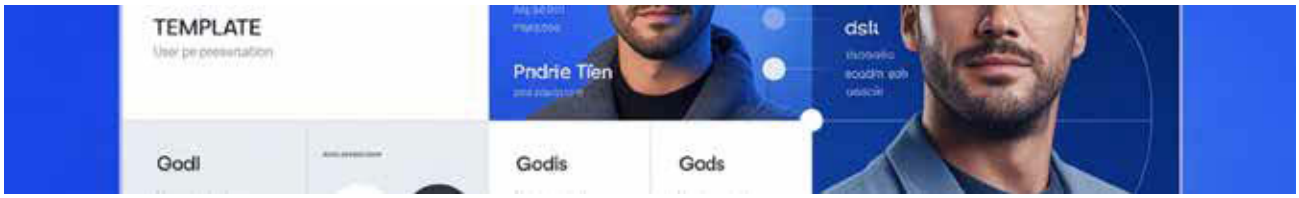
- デジタルでの課題提出
- 教員からのコメント返信
- 提出状況の可視化

C. 学校案内・申込

受験生・保護者向けのオープンスクール予約システム

- 学校情報の閲覧
- イベント日程の確認
- オンライン申込完結

重要なルール：1機能につき1タスクに絞って設計します。複数の機能を詰め込むと設計が複雑になり評価が困難になります。



Step 1：ユーザーと主要タスクの固定

想定ユーザー
保護者、生徒、教員など、具体的なユーザー像を1つ選択します
例：「朝忙しい保護者」「通学時間が長い生徒」

目的（タスク）
ユーザーが達成したい具体的な行動を定義します
例：「欠席連絡を1分以内で完了させる」

成功条件
タスクが成功したと判断できる基準を設定します
例：送信完了画面が表示される、履歴が残る、担任に通知が届く

制約条件
設計上の前提や制限事項を明記します
例：入力項目を最小限にする、個人情報に配慮、スマホ操作前提



Step 2：タスク分解と画面遷移

最小の画面数（3〜6画面）で、ユーザーがタスクを完了できるフローを設計します。画面が多すぎると離脱の原因になります。

ホーム画面
機能へのエントリーポイント

入力画面
必要情報を収集

確認画面
入力内容を確認

完了画面
成功のフィードバック

重要ポイント：「戻る」「修正」ができる設計を必ず入れましょう。ユーザーがミスを修正できることで、安心感と信頼感が生まれます。

Step 3：画面の基本骨格



上部エリア

画面タイトルと戻るボタンを配置します。ユーザーが今どこにいるか常に分かるようにします。

中央エリア

主要な情報表示と入力フォームを配置します。最も重要なコンテンツエリアです。

下部エリア

主ボタン（Primary）と補助ボタン（Secondary）を配置します。アクションの優先順位を明確にします。

原則：主ボタンは1つに絞る

複数の同じレベルのボタンがあると、ユーザーは迷います。次に進むべき行動を1つに絞ることで、迷いを消し、タスク完了率が向上します。

UI要素の選び方

授業で教えやすい基本原則

選択優先の原則

入力は「選択＞記述」を基本にします。選択式は速く、誤りが減り、ユーザーの負担を最小化できます。

例：欠席理由はプルダウンメニュー、日付はカレンダー選択

情報の優先順位

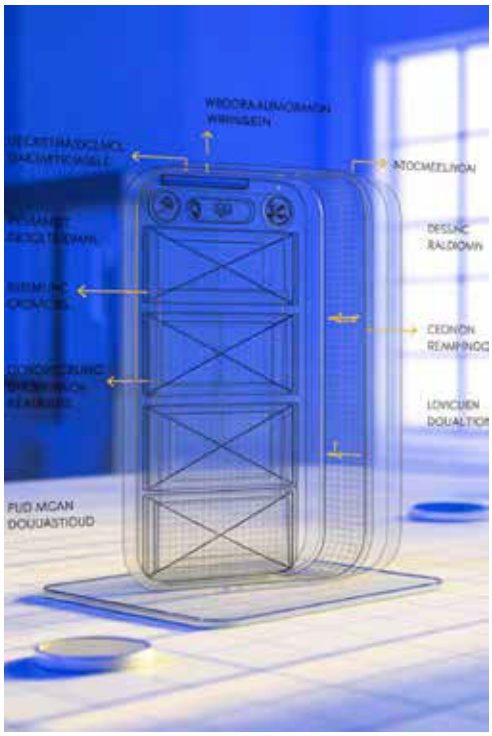
情報は重要順に上から並べます。ユーザーが最初に見るべき情報を画面の上部に配置しましょう。

例：タスク名→期限→詳細の順

即時フィードバック

エラーはその場で表示します。後出しのエラーは離脱の原因になります。リアルタイムで検証しましょう。

例：未入力、形式エラーを入力中に表示



ワイヤーに入れる注釈

各画面に最低1つ、できれば3つの注釈を入れましょう。
注釈は設計意図を言語化し、レビューや改善を容易にします。

1

意図の明示

「この要素は何のためにあるのか？」

例：「欠席理由の選択肢を5つに絞ることで、入力時間を30秒短縮する」

2

挙動の説明

「ユーザーが操作すると何が起こるのか？」

例：「送信ボタンを押すと確認画面に遷移し、内容を再確認できる」

3

例外処理の記述

「失敗したらどうなるのか？」

例：「未入力の場合は赤字でエラーメッセージを表示」「通信不良時は再送信ボタンを表示」

品質チェック　UX観点の3つの質問

次のユーザーテストと改善に直結する重要なチェックポイントです。

Q1
最初の1タップで迷わない？

ユーザーが画面を開いた瞬間に、次に何をすべきか明確ですか？

タイトル、ボタン配置、視覚的階層を確認しましょう。

Q2
主ボタンは1つに見える？

複数のボタンがある場合、どれが主要なアクションか一目で分かりますか？

色、サイズ、配置で優先順位を表現しましょう。

Q3
入力が面倒すぎない？

テキスト入力を最小限にし、選択式に変更できませんか？

ユーザーの負担を減らすことが完了率向上の鍵です。

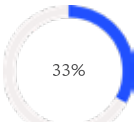


成果物と評価

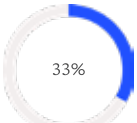
提出物（ユーザーテストに回せる形）

- ユーザーと主要タスク
  1〜2行で明確に記述
- 画面遷移フロー
  矢印で繋いだシンプルな図でOK
- ワイヤーフレーム
  3〜6画面分を作成
- 注釈
  各画面に最低1つ以上の説明を追加

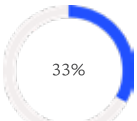
評価基準（ミニルーブリック）



タスクの明確性
ユーザーと目的が具体的に定義されている



フローの完全性
画面遷移が論理的で破綻していない



設計の根拠
各要素に明確な理由と注釈がある

教員研修プログラム

モジュール3：インタラクション & UXデザイン

4.ユーザーリサーチの方法

今日のゴールと学習の流れ

01

3つのリサーチ手法を理解

インタビュー、観察、アンケートの特徴と使い分けを学びます

02

実践的な質問設計を習得

そのまま使える質問例とNG例を通じて、効果的な質問の作り方を身につけます

03

リサーチ計画書を完成

問い・対象・手法・質問・記録方法・倫理配慮を含む1枚の計画書を作成します

この研修で作成した計画書は、次回の「UX評価と改善」研修へつながります。リサーチ結果からワイヤーフレーム作成、そしてUX評価へと段階的に進めていきます。

UXリサーチで「思いつき改善」を防ぐ



個人の好みや経験則だけでUI改善を進めると、実際のユーザーニーズとズレが生じ、かえって使いにくくなることがあります。

リサーチで明らかになる3つの軸

- **どこで詰まるか**：具体的な行動の障壁
- **なぜ詰まるか**：背景にある理由や文脈
- **どれくらい起きるか**：問題の頻度と重要度

3つの手法を組み合わせることで、深さと広さの両面から根拠のある改善策を導き出せます。

リサーチクエスチョンの作り方

明確な問いを立てることが、効果的なリサーチの第一歩です。広すぎる問いは焦点がぼやけ、具体的な改善につながりません。

✕ 悪い例

「このシステムが使いにくい理由は何ですか?」

→ *広すぎて何を聞けばよいか分からない*

✓ 良い例

「誰が／どの場面で／何に迷い／どう対処しているか」

→ *具体的で行動が見える*

問いを作るフォーマット

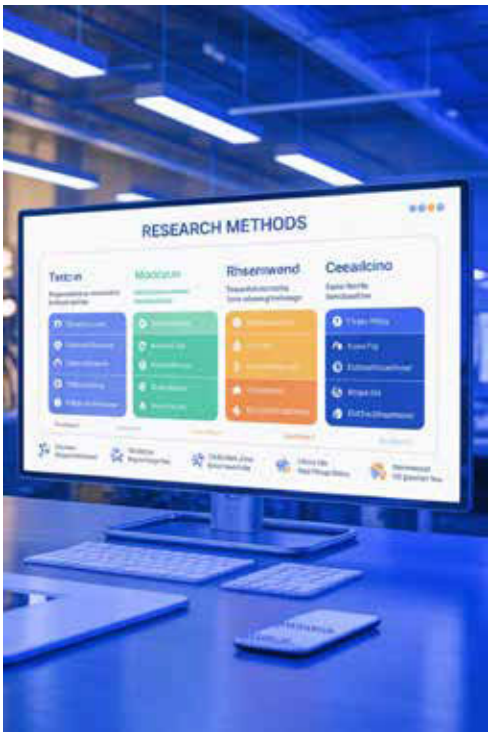
Who × **When/Where** × **What** × **Why/How**

- 例：新任教員が／朝の連絡業務で／連絡アプリのどの機能に迷い／どのように解決しているか
- 例：保護者が／提出期限直前に／学校フォームのどこで詰まり／なぜ途中離脱するのか

3つの手法の使い分け

1	インタビュー **目的**：理由・背景を深掘りする **人数**：3〜5名程度の少人数でOK **強み**：「なぜ」を引き出せる、文脈が分かる
2	観察 **目的**：実際の行動を捉える **対象**：自然な操作場面を記録 **強み**：言行不一致を発見できる、無意識の行動が見える
3	アンケート **目的**：傾向・頻度を把握する **規模**：母数を増やして全体像を掴む **強み**：定量的な根拠が得られる、優先順位がつけやすい

推奨の組み合わせ：インタビューで仮説を立て → 観察で行動を確認 → アンケートで傾向を検証する流れが効果的です



インタビュー設計の基本

人数の目安

3〜5名のインタビューで、主要な「詰まりどころ」はほぼ見えてきます。
完璧を目指すより、まず少人数で始めることが重要です。

質問の基本的な流れ

1. 事実
   いつ使った?
2. 行動
   何をした?
3. 感情
   どう感じた?
4. 理由
   なぜそうした?
5. 理想
   どうあるべき?

避けるべきNG質問

- 「便利だと思いませんか?」→ *誘導質問*
- 「なぜ使わないんですか?」→ *詰問調*
- 「たぶん○○ですよね?」→ *推測の押し付け*

記録方法

メモは必須。可能であれば録音も併用しますが、必ず事前に同意を取得してください。

インタビュー質問例（そのまま使える）

以下の質問は実際のインタビューでそのまま使えます。対象システムに合わせて単語を置き換えてください。

1. 直近で使った状況を教えてください
   いつ、どんな目的で使いましたか?
2. その時の目的は何でしたか?
   何を達成しようとしていましたか?
3. 迷った瞬間や困った場面はありましたか?
   具体的にどこで、どう感じましたか?
4. その時、何を見て判断しましたか?
   どんな情報やヒントを探しましたか?
5. 代わりの方法を試しましたか?
   別の手段や回避方法はありましたか?
6. 理想の流れはどんなものですか?
   どうなっていたらスムーズでしたか?
7. 1つだけ直すなら、どこですか? それはなぜですか?
   最も改善してほしい点とその理由を教えてください



観察調査で行動の真実を捉える

観察調査の目的は、「言っていること」ではなく「実際にやっていること」を記録することです。
ユーザーは自分の行動を正確に説明できないことが多いため、観察が重要になります。

観察すべき行動の種類

- 操作の流れと順序
- 迷いや戸惑いの瞬間
- 画面を行ったり来たりする動き
- 何かを探している時間
- 一時停止や考え込む様子
- 誤タップや操作ミス
- 他の人に助けを求める行動
- 諦めて別の方法を探す瞬間

記録シートの基本構成

時刻	行動	発言	詰まり/困った様子	仮説
9:05	提出ボタンを探す	「あれ、どこだっけ」	15秒間画面を上下	ボタン位置が分かりにくい?

アンケート設計のポイント

回収率を上げる設計

設問数は**5〜10問+自由記述1つ**が理想です。
長すぎると回答率が下がります。

取得したい主な軸

- 利用頻度（毎日/週1/月1等）
- 詰まる場面（選択肢で提示）
- 迷いの程度（5段階評価）
- 使用端末・環境
- 総合満足度

選択式は**傾向**を、自由記述は**理由**を拾う役割です

よくある失敗例

1問に2つ聞く

✕「使いやすくて便利ですか?」
→ 使いやすさと便利さは別の質問に

専門用語が多い

✕「UIのユーザビリティは?」
→「画面の使いやすさは?」と平易に

選択肢が偏っている

「とても良い/良い/普通」だけでは、否定的な意見を拾えません

リサーチ計画書のアウトプット

本日の研修の最後に、以下の項目を含む1枚の計画書を完成させます。この計画書が次回の「UX評価と改善」研修の出発点になります。

リサーチクエスチョン

Who × When/Where × What × Why/Howの形式で1つ明確に記述

対象・方法・規模

誰に/どの手法で/何名・何時間かけて実施するか

質問内容

インタビュー5問、またはアンケート5問を具体的に記載

記録方法

メモ/録音/画面録画など、データをどう残すか

倫理配慮

同意取得・個人情報保護・校内ルール遵守の方法

次回研修では、この計画に基づいて実施したリサーチ結果を分析し、Fact（事実）→ Insight（解釈）→ HMW（改善仮説）の流れでUX改善案を作成していきます。

教員研修プログラム

モジュール3：インタラクション & UXデザイン

5.UX評価と改善



今日のゴール

ユーザーテスト実施

目的・仮説・タスクを明確に定義し、実際のテストを短時間で実行できるようになります

課題の言語化

観察した行動からIssueを具体的に抽出し、改善につながる形で記述できるようになります

改善サイクル完了

優先順位を付けて改善案を変更点に落とし込み、次の検証計画まで作成できるようになります

UX評価の3つの原則

評価を始める前に、チーム全体で共有すべき考え方を確認しましょう。

タスク達成で評価する

個人の「好き嫌い」ではなく、**ユーザーがタスクを完了できたか**という客観的な基準で判断します。デザインの美しさより機能性を重視します。

行動観察を優先する

ユーザーが「言ったこと」より「**実際にやったこと**」を観察します。発言と行動が異なる場合、行動データの方が信頼性が高いです。

失敗は設計のヒント

ユーザーの失敗や迷いは、**設計を改善するための貴重な情報**です。人を責めるのではなく、システムの問題として捉えます。

評価の4つの観点

ユーザーテスト中に注目すべき具体的なポイントを整理します。

成功・失敗

タスクの達成度を記録します。完全成功、部分成功、失敗の3段階で評価し、どの段階でつまずいたかを明確にします。

所要時間

タスク完了までの時間を測定します。想定時間との差異がUIの効率性を示す重要な指標になります。

迷い・停滞

ユーザーが止まる、戻る、探すといった行動を観察します。これらは情報設計やナビゲーションの問題を示すサインです。

誤操作

意図しないボタンのクリックや入力ミスを記録します。UIの明確性や操作性の課題が見えてきます。



60分で回す改善サイクル

本日の実践的なワークフローを6つのステップで進めていきます。

テスト設計
目的・仮説・タスクを明確に定義します（10分）

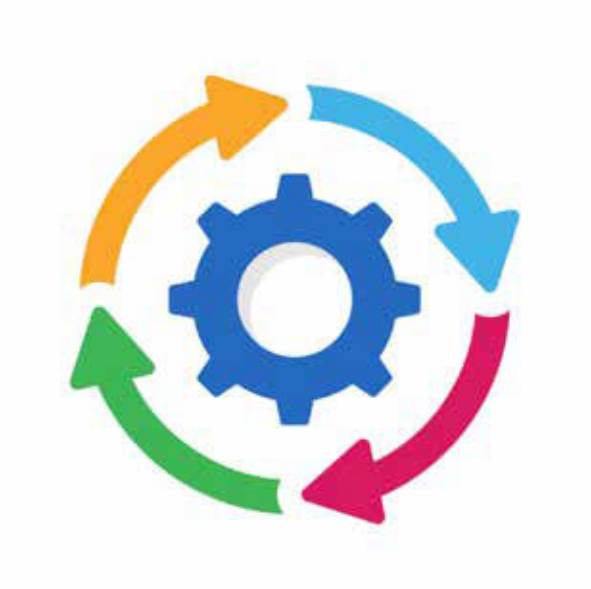
ミニユーザーテスト
短時間で複数人を対象に実施します（15分）

Issue抽出
観察から課題と原因仮説を導きます（10分）

優先順位付け
Impact×Effortで判断します（10分）

改善案作成
具体的な変更点に落とし込みます（10分）

再テスト計画
次の検証サイクルを設計します（5分）



テスト対象の例

欠席連絡ミニアプリ

今日のワークショップでは、学校の欠席連絡を送るシンプルなアプリケーションを題材として使用します。

- **前提条件**：初見ユーザーがスマートフォンで操作
- **画面形式**：ワイヤーフレームまたは既存の画面案
- **想定シーン**：朝の忙しい時間帯に素早く連絡したい保護者

実際のプロダクトでは、自社のサービスやプロトタイプに置き換えて実践してください。





テスト設計テンプレート

明確な目的と仮説を立てることで、観察すべきポイントが絞られます。

01

目的を定義する

何を確認したいのか具体的に記述します。
例：「初見ユーザーが1分以内に欠席連絡を完了できるか検証する」

02

仮説を立てる

うまくいく条件や失敗する要因を予測します。
例：「トップ画面に入口があれば迷わずに進める」

03

タスクを設計する

2つまでの具体的なタスクを用意します。
例：「欠席連絡を送る」「理由を入力して送信する」

04

成功基準を決める

時間、成功率、迷い回数など、定量的な基準を設定します。
例：「60秒以内、成功率80%以上」



3人1組の役割分担

効果的なユーザーテストには、明確な役割分担が不可欠です。

テスター

実際に操作する人

- 思考を声に出しながら操作する
- 自然な行動を心がける
- わからない時も素直に反応する

モデレーター

進行役・質問者

- テスターを誘導しない
- 適切なタイミングで質問する
- 中立的な態度を保つ

観察者

記録係

- 迷いや誤操作を記録する
- 所要時間を測定する
- 印象的な発言をメモする

モデレーターの台本

言っていい3つの質問

1 「今、何をしようとしていますか？」

ユーザーの意図を確認します

2 「そう思った理由は何ですか？」

判断の根拠を深掘りします

3 「そのまま進めてください」

自然な行動を促します

NG行動

- 誘導する質問「ここを押してみては？」
- 操作を指示する「次はこうしてください」
- 解説する「これはこういう意味です」

テスターの自然な行動を妨げないことが
最も重要です。

観察ログの記録フォーマット

一貫した形式で記録することで、後の分析が容易になります。

1	タスク達成度 ○（完全成功）、△（部分成功）、×（失敗）で記録します
2	所要時間 タスク開始から完了までの秒数を測定します
3	迷いの記録 「止まる」「戻る」「探す」の行動がどこで発生したか記録します
4	誤操作の詳細 何を押した？何が起きた？を具体的に記述します
5	印象的な発言 ユーザーの率直な感想や困惑を短くメモします

観察からIssue化へ

観察した事実を、改善可能な課題として構造化します。

1 観察事実

起きたことをそのまま記述

例：ユーザーが入口を探して
画面を3回往復した

2 原因仮説

なぜ起きたのか推測

例：ボタンの文言が統一されておらず
どれが入口か判断できなかった

3 影響の明確化

何が困るのか定義

例：時間が増える、途中で諦める
送信ミスが起きる

4 改善案

解決策を1行で提案

例：入口を固定し、文言を
「欠席連絡」に統一する

重み付け：Severity 0〜3

すべての課題を同時に直すことは不可能です。影響度で優先順位を付けましょう。

⚠ Severity 3：致命的

タスクが完全に失敗する、システムが動かないなど、**即座の対応が必要**な問題です。すべてのユーザーに影響します。

❗ Severity 2：重大

大きな遅延やストレスを引き起こす問題。タスクは完了できるが、**ユーザー体験が著しく損なわれます**。

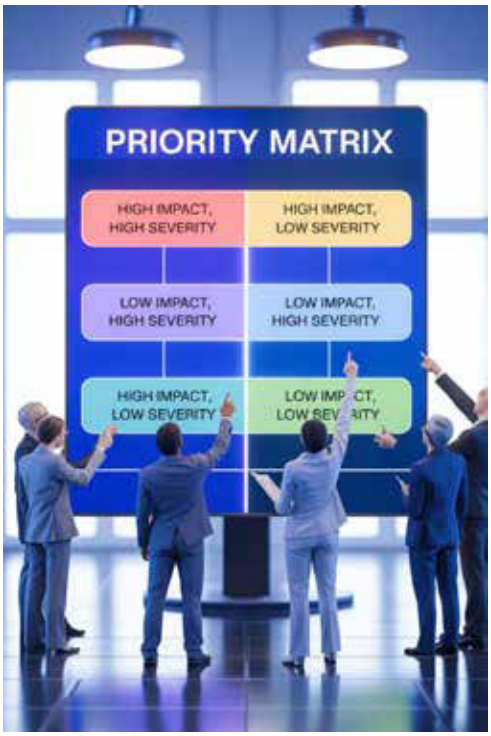
ⓘ Severity 1：軽微

一時的な迷いや小さな違和感。影響は限定的ですが、**改善すればより良い体験**になります。

☆ Severity 0：好みレベル

個人の好みや意見の範囲。機能には影響しないため**優先度は最も低く**設定します。

ポイント：Severity 3と2から優先的に着手し、限られた時間で最大の改善効果を狙います。



Impact × Effort で優先順位を決める

2×2マトリクスで改善項目を分類し、実装の優先順位を明確にします。



まず直すべき項目

Impact高 × Effort低の領域が最優先です。大きな効果が期待でき、実装コストも低いため、すぐに取り組むべき改善項目です。

改善を「変更点」で書く

抽象的な改善案ではなく、**具体的な変更内容**に落とし込みます。

- 文言を「連絡」→「欠席連絡」に変更
- ボタンをトップ画面の1画面目に配置
- 理由入力を自由記述から選択式に変更
- 送信確認ダイアログを追加

改善サイクルを閉じる

最後のステップは次の検証計画です。改善が効果を発揮したか確認しましょう。

次の目的	次のタスク	成功基準
今回の改善点が本当に効果があったかを検証します	検証したい改善点に絞って1つのタスクを設計します	例：「30秒以内に完了」「戻る操作0回」など定量的に設定

テスト → Issue → 優先度 → 改善 → 再テスト

この**5ステップのサイクル**が、UX改善の基本フレームワークです。繰り返すことで継続的にプロダクトの品質を向上させることができます。

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。