

| | | | | | | | | | |
|-------|-------------------------|-----|---------|-----|----|---------|---|-----|------|
| 科目名 | システム設計 2 | | | | | | | 年度 | 2025 |
| 英語科目名 | System Design 2 | | | | | | | 学期 | 前期 |
| 学科・学年 | 情報処理科 システム開発コース 2 年次 | 必/選 | 必 | 時間数 | 30 | 単位数 | 2 | 種別※ | 講義 |
| 担当教員 | 鈴木 睦男 | | 教員の実務経験 | | | 実務経験の職種 | | | |

【科目の目的】

学生にソフトウェア開発における基本的な原則から応用、プロジェクトマネジメント、セキュリティ、プロフェッショナルスキルまでの広範なスキルと知識を提供し、実務でのソフトウェア開発において成功するための準備を行うこと。現代のソフトウェア開発環境で成功するための基盤を築くためのコアなスキルと知識を提供することを目指している。

【科目の概要】

ソフトウェアは現代社会のあらゆる領域に浸透しており、企業から個人まで多くの人々がソフトウェアを利用・提供している。そのため、ソフトウェア開発のスキルはますます重要であり、この科目によって学生は現代の技術社会での成功に向けて準備を整える。現代のテクノロジー環境での成功に向けて必要なスキルを習得するための基盤を提供し、学生がソフトウェア開発のプロフェッショナルとして成長するための重要な一歩となる。

【到達目標】

- ・ソフトウェア開発スキルの習得
プログラミング言語を使用して基本的なアプリケーションを開発できる能力を習得する。データベースを設計し、クエリを実行してデータを操作できるスキルを獲得する。ユーザーインターフェースとユーザーエクスペリエンスの設計と実装を行えるようになる。
- ・ソフトウェア開発プロセスの理解
ソフトウェア開発のライフサイクルを理解し、要件定義からテスト・運用までのプロセスを実践的に遂行できるようになる。プロトタイピングと反復開発を通じて、要件の変更への柔軟な対応能力を養う。

【授業の注意点】

この授業計画は、学生が実務でのソフトウェア開発において成功するためのスキルを網羅的に習得できるよう設計されている。ただし、具体的な授業内容やスケジュールは学生のニーズや進捗に合わせて調整が必要。また、実践的なプロジェクトや演習を通じて学生が実際の経験を積むことが重要。

評価基準＝ルーブリック

| ルーブリック 評価 | レベル5 | レベル4 | レベル3 | レベル2 | レベル1 |
|--------------|---------------------------------------------------------------------|-----------------------------------------------------------------|----------------------------------------------------|----------------------------------------------------------------|----------------------------------------------------------|
| | 優れている | よい | ふつう | あと少し | 要努力 |
| 到達目標 A | 学生は高度なプログラミング技術とデータベース設計の知識を持ち、創造的なアプリケーションの開発が可能です。 | 学生は複雑なアプリケーションを独自に開発できるスキルを持ち、プログラミングとデータベース操作に自信があります。 | 学生はプログラミング言語やデータベース設計の基本を理解し、単純なアプリケーションの開発が可能です。 | 学生は一部のプログラミング概念やデータベース設計について理解していますが、アプリケーションの独自開発にはまだ課題があります。 | 学生は基本的なプログラミングやデータベースの操作について理解が不足しており、アプリケーションの開発が難しい。 |
| 到達目標 B | 学生はアジャイル開発などの異なるプロセスやアプローチについて深い理解を持ち、適切なプロセスの選択と適応を行います。 | 学生はプロジェクトの計画、設計、実行、テスト、運用などのステージについて詳細な理解を持ち、プロジェクトを効果的に遂行できます。 | 学生はソフトウェア開発のライフサイクルと主要なプロセスを理解し、基本的なプロジェクト運営が可能です。 | 学生は一部のソフトウェア開発プロセスを理解していますが、プロジェクト全体の流れについて完全に把握できていません。 | 学生はソフトウェア開発の基本的なプロセスやライフサイクルについて理解しておらず、適切なプロジェクト管理が難しい。 |
| 到達目標 C | 学生はテストカバレッジの評価や負荷テスト、セキュリティテストなどの高度なテスト手法を理解し、品質保証の専門家としてのスキルを持ちます。 | 学生はテスト自動化やCI/CDの基本的な手法を理解し、効率的で継続的なテストプロセスを確立できます。 | 学生はテストの段階とその目的を理解し、ユニットテストや統合テストを実施するスキルを持っています。 | 学生は一部のテスト手法や品質保証の基本を理解していますが、網羅的なテスト戦略の構築が難しいです。 | 学生はテストの重要性やテスト手法について理解が浅く、品質保証の概念を適切に適用できていません。 |

| | | | | | |
|-----------|------------------------------------------------------------------------|-------------------------------------------------------|-----------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------|
| 到達目標 D | 学生は将来の技術トレンドや変化を予測し、それに適応するための戦略を立案できます。また、新技術のリーダーシップと導入を行えるスキルを持ちます。 | 学生は業界のトレンドや新技術の影響を正確に評価し、プロジェクトに導入する際の戦略を策定できます。 | 学生は新技術の調査と評価を行い、プロジェクトに導入するための基本的なスキルを持っています。 | 学生は一部の新技術に関する情報を追跡していますが、それらをプロジェクトに組み込む能力には課題があります。 | 学生は新しい技術やトレンドの調査や導入について理解が不足しており、変化する環境に適応できていません。 |
| 到達目標 E | 学生はアジャイル開発やスクラムなどの高度なプロジェクトマネジメント手法を理解し、複雑なプロジェクトの成功に貢献できます。 | 学生はプロジェクトの監視、進捗報告、リスク管理などを効果的に実行し、チームをリードする能力を発揮できます。 | 学生はプロジェクトの計画やスケジュール管理、チームコミュニケーションを理解し、プロジェクトチームで協力する能力を持っています。 | 学生はプロジェクトマネジメントの基本概念やチームワークの一部について理解していますが、それらを実践に活かすことが課題です。 | 学生はプロジェクトマネジメントの基本やチームワークの重要性について理解が不足しており、効果的な協力が難しいです。 |

【教科書】
指定書籍無し。必要に応じて資料等を配布する。

【参考資料】

【成績の評価方法・評価基準】
テストと課題の提出:学生は各テーマごとに実施されるテストや課題に取り組み、理解度や技術的なスキルを示します。テスト結果や提出物の品質が評価の対象です。
プロジェクトの実施と発表:学生はチームで実際のソフトウェアプロジェクトを計画し、開発し、最終成果物を発表します。プロジェクトの内容、進捗、品質、プレゼンテーションスキルなどが評価されます。

※種別は講義、実習、演習のいずれかを記入。

| 科目名 | | システム設計 2 | | | 年度 | 2025 | |
|------|-----------------------|------------------------------------------------------------|------|---------------------------|-----------------------------------------------|------|--|
| 英語表記 | | System Design 2 | | | 学期 | 前期 | |
| 回数 | 授業テーマ | 各授業の目的 | 授業内容 | 到達目標=修得するスキル | 評価方法 | 自己評価 | |
| 1 | システム開発の基本概念とプロセス | ソフトウェア開発の一般的なフローと重要な用語を理解し、ウォーターフォールモデルとアジャイル開発の違いを認識する。 | 1 | ソフトウェア開発の基本概念の理解 | ソフトウェア開発の意義とプロセスについて説明できる。 | 1 | |
| | | | 2 | ウォーターフォールモデルとアジャイル開発の比較 | ウォーターフォールモデルとアジャイル開発の特徴と適切なシナリオを説明できる。 | | |
| | | | 3 | ソフトウェア開発プロセスの段階の把握 | ソフトウェア開発プロセスの異なる段階を列挙し、各段階の目的を理解できる。 | | |
| 2 | プロジェクト計画と要件定義の重要性 | プロジェクト計画とスコープの定義がなぜ重要かを理解し、要件収集の基本的な手法に慣れる。 | 1 | プロジェクト計画の要素と作成手法の理解 | プロジェクト計画の要素と作成手法を説明できる。 | 1 | |
| | | | 2 | スコープの定義と範囲の管理 | プロジェクトスコープの範囲を適切に定義し、変更管理の重要性を理解できる。 | | |
| | | | 3 | 要件収集の技法とツールの活用 | 要件収集の主な技法やツールを説明し、適切な方法を選択できる。 | | |
| 3 | ユーザーストーリーの作成と要件収集 | ユーザーストーリーを作成し、適切な要件を収集するスキルを習得する。 | 1 | ユーザーストーリーの作成と優先順位付けの方法理解 | ユーザーストーリーを正しく作成し、重要度に応じて優先順位をつける方法を理解する。 | 1 | |
| | | | 2 | 要件収集のためのワークショップの実施 | 要件収集ワークショップを計画し、効果的に実施する手法を習得する。 | | |
| | | | 3 | ユーザーとのコミュニケーションの重要性理解 | ユーザーとのコミュニケーションの重要性を理解し、要件の適切な理解と調整の方法を学ぶ。 | | |
| 4 | ソフトウェア設計の基本原則とアーキテクチャ | ソフトウェアの設計原則やレイヤードアーキテクチャについて理解し、優れた設計を行うスキルを身につける。 | 1 | ソフトウェア設計の原則とパターンの理解 | ソフトウェア設計の基本原則 (DRY, SOLIDなど) と設計パターンについて理解する。 | 1 | |
| | | | 2 | レイヤードアーキテクチャとコンポーネント設計 | レイヤードアーキテクチャの概念を理解し、コンポーネントベースの設計方法を学ぶ。 | | |
| | | | 3 | データベース設計と正規化の基本 | データベースの基本設計原則と正規化の概念を理解する。 | | |
| 5 | プログラミングと実装の基礎 | 選択したプログラミング言語を使用して基本的なコーディングを行い、コーディングスタンダードとベストプラクティスを学ぶ。 | 1 | プログラミング言語の基本構文とデータ型の理解 | 選択したプログラミング言語の基本構文とデータ型を理解する。 | 1 | |
| | | | 2 | コーディングスタンダードとベストプラクティスの適用 | クリーンコードの原則とコーディングスタンダードを適用する。 | | |
| | | | 3 | ユニットテストとデバッグの基本 | ユニットテストの価値を理解し、デバッグの基本的な手法を習得する。 | | |
| 6 | ソフトウェアテストの基本とテスト計画 | ソフトウェアテストの重要性を理解し、テスト計画とテストケースの作成方法を学ぶ。 | 1 | ソフトウェアテストのタイプと目的の理解 | ソフトウェアテストの異なるタイプとそれらの目的を理解する。 | 1 | |
| | | | 2 | テスト計画とテストケースの作成方法 | テスト計画の作成手法とテストケースの設計方法を学ぶ。 | | |
| | | | 3 | バグトラッキングと品質保証の基本 | バグトラッキングの重要性と品質保証のプロセスを理解する。 | | |
| 7 | 自動化テストとCI/CDの導入 | 自動化テストの手法やCI/CDの導入方法を学び、効率的なテストとデプロイプロセスを理解する。 | 1 | 自動化テストの概念と利点の理解 | 自動化テストの意義と利点を理解し、手動テストとの比較ができる。 | 1 | |
| | | | 2 | CI/CDの基本とツールの導入 | CI/CDの流れとツール (Jenkins, Travis CIなど) の導入方法を学ぶ。 | | |
| | | | 3 | テストカバレッジと自動化テストの適用 | テストカバレッジの重要性を理解し、自動化テストのカバレッジを向上させる。 | | |

| | | | | | | |
|----|-----------------------|---------------------------------------------------------------------------------------|---|---------------------------|----------------------------------------------------|---|
| 8 | デプロイと運用 | デプロイメント戦略の選択とクラウドコンピューティングの基本、モニタリングとログ管理のスキルを習得し、安定した運用とメンテナンスが可能なシステムを構築する能力を向上させる。 | 1 | ソフトウェアデプロイメント戦略と選択肢の理解 | ソフトウェアデプロイメント戦略の選択肢とその適切な使用法を理解する。 | 1 |
| | | | 2 | クラウドコンピューティングとサーバーインフラの基本 | クラウドコンピューティングの基本概念とクラウドプロバイダの選定基準を理解する。 | |
| | | | 3 | モニタリングとログ管理の重要性と手法 | ソフトウェアシステムのモニタリングとログ管理の重要性を認識し、ツールの利用法を理解する。 | |
| 9 | ユーザー受け入れテストとフィードバック処理 | ユーザー受け入れテストの実施とフィードバック処理のスキルを習得し、ユーザーエクスペリエンスの向上とアプリケーションの品質向上に貢献する能力を向上させる。 | 1 | ユーザー受け入れテストの目的とプロセスの理解 | ユーザー受け入れテストの重要性とプロセスを理解し、適切なテストケースを設計できる。 | 1 |
| | | | 2 | ユーザーフィードバックの収集と処理方法 | ユーザーからのフィードバックを収集し、適切に分析して改善する方法を学ぶ。 | |
| | | | 3 | ユーザーエクスペリエンスとアクセシビリティの考慮 | ユーザーエクスペリエンスとアクセシビリティの重要性を理解し、設計とテストに適用する。 | |
| 10 | プロジェクト成果物と評価 | プロジェクト成果物の管理と評価、品質保証のスキルを習得し、プロジェクトの成功と持続可能な価値の提供を実現する能力を向上させる。 | 1 | プロジェクト成果物の整理とドキュメンテーション | プロジェクト成果物の整理と適切なドキュメンテーションの作成方法を学ぶ。 | 1 |
| | | | 2 | プロジェクトの評価基準と品質管理 | プロジェクトの評価基準と品質管理の方法を理解し、成果物の品質を確保する。 | |
| | | | 3 | アーカイブと知識蓄積の重要性 | プロジェクトの終了後にも継続的な価値を提供するためのアーカイブと知識蓄積の重要性を理解する。 | |
| 11 | 新技術と将来のソフトウェア開発トレンド | 新技術の継続的な学習とソフトウェア開発の将来に向けた洞察を養い、変化する環境での成功に向けたスキルを向上させる。 | 1 | 新技術の調査と探求の方法 | 新しい技術とツールを探し、評価する方法を学ぶ。 | 1 |
| | | | 2 | 将来のソフトウェア開発トレンドの予測と分析 | ソフトウェア開発の将来的なトレンドと変化について予測し、分析する能力を養う。 | |
| | | | 3 | 技術の導入とチームへの展開 | 新しい技術の導入とチームへの展開方法を理解し、成功させるスキルを学ぶ。 | |
| 12 | プロジェクトマネジメントとチーム協力 | プロジェクトの効率的なマネジメントとチームワークの強化を通じて成功に向けたスキルを習得する。 | 1 | プロジェクトマネジメントの基本原則と方法 | プロジェクトマネジメントの基本原則と手法を理解し、プロジェクトの進捗とリソースの管理を学ぶ。 | 1 |
| | | | 2 | チームコラボレーションとコミュニケーションの強化 | チームコラボレーションとコミュニケーションの重要性を認識し、効果的なコラボレーション方法を学ぶ。 | |
| | | | 3 | スクラムとアジャイル開発の導入 | スクラムとアジャイルの基本概念を理解し、プロジェクトに導入する方法を学ぶ。 | |
| 13 | インダストリアルスタンダードとセキュリティ | 業界標準とセキュリティの基本、データプライバシーと法的規制への適合を理解し、安全なソフトウェアの開発と運用のスキルを習得する。 | 1 | 業界標準とベストプラクティスの理解 | ソフトウェア開発の業界標準とベストプラクティスを理解し、その重要性を認識する。 | 1 |
| | | | 2 | セキュリティの基本原則と脆弱性の管理 | ソフトウェアセキュリティの基本原則と脆弱性の管理方法を学ぶ。 | |
| | | | 3 | データプライバシーと法的規制の理解 | データプライバシーと関連する法的規制（GDPR、HIPAAなど）を理解し、遵守するスキルを習得する。 | |
| 14 | プロフェッショナル開発者とキャリア準備 | プロフェッショナルなソフトウェア開発者としてのスキルセットやキャリア準備、職業倫理を向上させる能力を習得する。 | 1 | プロフェッショナル開発者としてのスキルセット | プロフェッショナルな開発者として必要なソフトスキルとコンピュータサイエンスの理解を強化する。 | 1 |
| | | | 2 | キャリア準備と面接対策の指導 | ソフトウェア開発者としてのキャリア準備と面接対策をサポートする。 | |
| | | | 3 | 職業倫理とプロフェッショナリズムの強化 | 職業倫理とプロフェッショナリズムの重要性を認識し、倫理的な開発と行動を強化する。 | |
| 15 | プロジェクト発表とフィードバック | プロジェクトの成果を効果的に発表し、フィードバックと自己評価を通じて継続的な成長意識を持ち、将来への展望を考える能力を向上させる。 | 1 | プロジェクト発表の準備とプレゼンテーションスキル | プロジェクトの成果とプロセスを効果的に発表し、プレゼンテーションスキルを向上させる。 | 2 |
| | | | 2 | 同僚からのフィードバックと自己評価の重要性 | プロジェクト仲間や同僚からのフィードバックを受け入れ、自己評価と向上への意識を高める。 | |
| | | | 3 | 学習の振り返りと未来への展望 | 科目全体の学習を振り返り、今後の学習やキャリアへの展望を考える。 | |

評価方法：1. 小テスト、2. パフォーマンス評価、3. その他

自己評価：S：とてもよくできた、A：よくできた、B：できた、C：少しできなかった、D：まったくできなかった

備考 等