

科目名	システム開発グループ演習						年度	2026	
英語科目名	System development group seminar						学期	後期	
学科・学年	ITスペシャリスト科 AI・システム専攻 3年次	必/選	選	時間数	120	単位数	4	種別※	実習
担当教員	高橋	教員の実務経験		有	実務経験の職種		システムエンジニア		

**【科目の目的】**

システム開発プロジェクトの一環として、チームメンバーと協力して一つのシステムを企画、設計、開発する経験を通じて、技術的スキルだけでなく、チームワークや問題解決力、コミュニケーション能力を身につけることを目的とします。

**【科目の概要】**

この科目では、システム開発の一連のプロセスをグループで体験し、企画、要件定義、設計、実装、テスト、発表といった開発工程を通して実務に必要なスキルを身につけます。授業は全15回で、1回目から7回目までを「自由開発編」、8回目から15回目を「設計重視開発編」として構成しています。前半では自由な発想でシステム開発を進め、後半では設計書の作成を必須とすることで、設計の重要性とその実践方法を体験的に学びます。プロジェクトを通して、チームでの役割分担、計画的な進行、課題解決力、そしてシステムの品質向上を意識した開発スキルを養成し、最終発表で成果を発表することを目指します。

アクティブラーニングのアプローチにより、学生は知識を受け取るだけでなく、自らの手でシステムを創造し、実現する力を養うことができます。最終的には、プロジェクトの成果物を発表し、学びを共有することで、より深い理解と成長を図ります。

**【到達目標】**

- A. 企画からテストまでの一連のシステム開発プロセスを理解し、各工程の重要性を認識する。
- B. グループでの協力を通じて、役割分担や円滑なコミュニケーションの重要性を体験し、チームワークを向上させる。
- C. 要件定義や設計書作成を実践し、システム設計の基本を学ぶことで、設計の重要性を体感する。
- D. プログラミングやシステム実装を通じて、技術的スキルを高め、実際のシステム開発に必要な能力を習得する。
- E. プロジェクトの最終成果を発表し、他者からのフィードバックを受け入れることで、自らの成長を促進し、次回に活かす。

**【授業の注意点】**

授業はグループで協力しながら作業を進めるため、遅刻・欠席をする場合は他のメンバーに迷惑をかけてしまうことを念頭に入れること。やむを得ず遅刻・欠席する場合は、他のメンバーに連絡を取って全体の作業に遅延を生じないようにすること。ただし、授業時数の4分の3以上出席しない者は評価の対象とならない。

評価基準＝ルーブリック

ルーブリック評価	レベル5 優れている	レベル4 よい	レベル3 ふつう	レベル2 あと少し	レベル1 要努力
到達目標 A	システム開発の全工程を深く理解し、具体的な事例を挙げて説明できる。	システム開発の主要な工程を理解しており、基本的な内容を説明できる。	システム開発の流れを理解しているが、一部の工程について詳細が不足している。	システム開発プロセスの理解が不十分で、重要な工程を見落としている。	システム開発プロセスについてほとんど理解しておらず、説明が困難。
到達目標 B	グループ内でリーダーシップを発揮し、他者との協力を促進できる。	チーム内での協力が円滑で、コミュニケーションを適切に行うことができる。	チーム内の協力はできているが、コミュニケーションにおいて若干の改善が必要。	チーム内での協力が不十分で、コミュニケーションに課題が見られる。	チームワークが機能せず、コミュニケーションに大きな問題がある。
到達目標 C	要件定義や設計書の作成が非常に高い精度で行われ、明確で具体的な内容が記載されている。	要件定義と設計書の作成が適切に行われており、基本的な内容が網羅されている。	要件定義や設計書の作成が行われているが、具体性や明確さに欠ける部分がある。	要件定義や設計書の内容が不十分で、必要な情報が欠落している。	要件定義や設計書の作成がほとんどできておらず、内容が不明瞭。
到達目標 D	プログラミングスキルが非常に高く、独自の実装を行うことができる。	プログラミングが正確に行われ、基本的な機能を実装する能力がある。	プログラミングが行われているが、機能に不具合が見られることがある。	プログラミングにおいて基礎的な知識はあるが、実装が不十分である。	プログラミングスキルが低く、実装がほとんどできていない。
到達目標 E	発表が非常に分かりやすく、他者のフィードバックを積極的に受け入れて改善点を明確に示せる。	発表が明瞭で、フィードバックに対して前向きな姿勢で対応できる。	発表が行われるが、内容に対する理解が浅く、フィードバックを活用できていない。	発表が不十分で、フィードバックへの対応が受動的である。	発表が混乱しており、フィードバックを全く受け入れようとしていない。

**【教科書】**

プロジェクトに応じて学生個々に指定

**【参考資料】**

**【成績の評価方法・評価基準】**

作品の内容や完成度、中間発表や最終発表の内容、グループ作業の状況などを総合的に評価する。

※種別は講義、実習、演習のいずれかを記入。

科目名		システム開発グループ演習			年度	2026
英語表記		System development group seminar			学期	後期
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	オリエンテーションとプロジェクト概要説明	プロジェクトの全体像、目的、チーム編成を理解し、開発の方向性を明確にする。	1 プロジェクトの全体像理解	プロジェクトの目的や進行スケジュール、評価方法を理解する。	2	
			2 チーム編成と役割分担	チームメンバーとの協体制を整え、役割分担を決定する。		
			3 開発の方向性を明確化	チームとしての目標を共有し、方向性を定める。		
2	要件定義とアイデアブレスト	自由にアイデアを出し、開発するシステムの要件をチームで具体化する。	1 自由なアイデア出し	チームで積極的にアイデアを出し合い、可能性を広げる。	2	
			2 要件の具体化	開発するシステムの要件を具体的に定義する。		
			3 コンセプトの決定	開発システムのコンセプトを共有し、チームで方向性を一致させる。		
3	システムの概要設計	システムの主要機能を設定し、チームで役割分担を決める。	1 主要機能の決定	システムの主要機能を定め、開発の枠組みを設計する。	2	
			2 役割分担の確立	各メンバーの役割を確認し、担当範囲を具体的に決める。		
			3 全体の設計方針を確認	チーム全員で設計方針を確認し、理解を深める。		
4	プロトタイプ作成と初期開発	システムの基本的な形を構築し、開発の方向性を固める。	1 プロトタイプ作成	システムの基本形を構築し、開発の土台を作る。	2	
			2 開発の方向性を固める	プロトタイプに基づき、今後の開発方向性を確認する。		
			3 初期機能のテスト	開発したプロトタイプが期待通りに動作するかテストする。		
5	各機能の開発と検証	各メンバーが担当機能を開発し、チームで共有しながら進捗を確認する。	1 担当機能の開発	各メンバーが担当する機能を個別に開発する。	2	
			2 進捗状況の共有	チーム内で進捗を共有し、課題点を早期発見する。		
			3 初期機能の検証と改善	開発した機能を検証し、改善点を洗い出す。		
6	システム統合と動作確認	各機能を統合し、システム全体として動作するか確認する。	1 各機能の統合	各メンバーが開発した機能を統合し、全体として動作するか確認する。	2	
			2 システム全体の動作確認	システム全体の動作を確認し、問題点を特定する。		
			3 バグ修正と品質向上	バグや不具合を修正し、システムの品質を向上させる。		
7	プレゼンテーションとフィードバック	開発したシステムをチームで発表し、他チームや指導教員からフィードバックを得る。	1 システム発表	開発したシステムをプレゼンテーション形式で発表する。	2	
			2 他チームからのフィードバック受領	他チームや指導教員からのフィードバックを受け入れる。		
			3 改良点の整理	フィードバックを基に、次の開発に向けた改良点を明確にする。		
8	プロジェクト再設定と設計書の重要性理解	設計書作成の重要性を学び、システム設計の基本を理解する。	1 設計書作成の重要性学習	設計書作成がプロジェクトにおいて重要である理由を理解する。	2	
			2 プロジェクト再設定	フィードバックをもとにプロジェクトの方向性を再設定する。		
			3 設計書作成の基本理解	設計書に含まれるべき基本要素を理解する。		
9	要件定義	要件を再確認し、システムの詳細要件定義と設計書作成を進める。	1 ユーザー要件とシステム要件の違い理解	ユーザーが求めるニーズと、システムが実際に提供すべき機能を明確に区別できる。	2	
			2 要件定義書の作成	要件定義書のフォーマットに従い、ニーズを漏れなく文書化できる。		
			3 要求事項の優先順位設定	ユーザーにとっての重要度と開発リソースを考慮し、実現可能な優先順位を設定できる。		
10	設計	要件定義をもとに設計書を作成し、開発の効率化と品質向上を図るための詳細なシステム設計を行う。	1 システム設計とプログラム設計の違い理解	システム全体の構造設計と、プログラム単位の詳細設計を区別し、それぞれの意義を説明	2	
			2 データフロー図・ER図作成	システム内でのデータの流れや関連性を図式化し、関係性を理解できる。		
			3 ユーザーインターフェース設計	ユーザーの操作性を考慮したインターフェースをデザインし、画面遷移の構成ができる。		
11	設計	要件定義をもとに設計書を作成し、開発の効率化と品質向上を図るための詳細なシステム設計を行う。	4 設計書の記述方法理解	設計書のフォーマットに従い、機能仕様やデータ仕様を正確に記述できる。	2	
			5 システム構成図やクラス図作成	システム全体の構造やクラスの関係を図として表し、構成を説明できる。		
			6 設計のレビューと改善	設計書をチームでレビューし、設計の漏れや不備を発見し、改善点を提案できる。		
12	実装	設計書に基づき、実際のシステム開発を行い、実装スキルと進捗管理能力を養う	1 開発環境の整備と共有	開発環境を適切にセットアップし、チームで共有できる。	2	
			2 基本機能の実装	設計書の仕様に沿って、基本的な機能を正しく実装できる。		
			3 バージョン管理ツールの使用	Git等のバージョン管理ツールを使い、実装の変更点を管理できる。		
13	実装	設計書に基づき、実際のシステム開発を行い、実装スキルと進捗管理能力を養う	4 TDDの基本理解	TDDの基本的な手順に沿って、テストケースを作成しながら開発できる。	2	
			5 機能統合と結合テスト	複数の機能を統合し、システム全体での動作を確認するテストを実施できる。		
			6 バグ修正とコードレビュー	発見したバグの修正と、チーム内でのコードレビューを通じて品質向上を図れる。		
14	テスト	開発したシステムが要件に適合しているかを確認し、不具合の洗い出しや修正を行うテスト技法を習得する	1 単体テストと総合テスト理解	各テストの目的と適用範囲を理解し、適切なテスト項目を設定できる。	2	
			2 テストケースの作成と実行	要件に基づいたテストケースを作成し、テストの実施と記録を行える。		
			3 テスト結果の分析と改善	テストの結果から不具合を分析し、改善すべき点を抽出・文書化できる。		
15	最終発表と振り返り	最終成果物を発表し、プロジェクト全体の振り返りと学びを共有する。	1 最終成果物の発表	チームで作成した成果物をプレゼンテーション形式で発表できる。	2	
			2 プロジェクトの総括と学びの共有	開発を通じて得た知見やスキルを振り返り、次に活かせるよう整理できる。		
			3 自己評価とフィードバック受け取り	自分やチームの達成度を評価し、今後の成長点を見出せる。		

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