

科目名	テスト技法							年度	2026
英語科目名	Software testing skills							学期	前期
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	必	時間数	30	単位数	2	種別※	講義
担当教員	山本	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】									
この授業は、ソフトウェアの品質向上を目的としたテスト技術の基礎から応用までを学ぶことを目指します。受講者は、同値分割法、境界値分析、デシジョンテーブル、状態遷移テスト、組合せテストといった代表的なテスト技法を体系的に理解し、実践的な演習を通して習得します。また、テスト計画の策定や報告書作成の技法も学び、実務に必要なテスト能力を向上させることを目標としています。									
【科目の概要】									
ソフトウェアテストの基礎から応用までを学び、品質の高いソフトウェアを作成するために必要なテスト技法を習得します。学生は、テストの種類や計画作成方法、テスト設計技法を理解し、実際の開発プロジェクトにおけるテストの実践に活かします。									
【到達目標】									
A. ソフトウェアテストの重要性と目的を理解する。 B. 各種テスト技法（単体テスト、結合テスト、システムテスト等）の違いと使用方法を習得する。 C. テストケースの設計、テスト計画の立案、バグ報告など、テスト業務を実務的にこなせるようになる。									
【授業の注意点】									
少人数による品質評価演習を中心に授業を展開する。演習を行う上で、これまで学習してきた知識の整理が必要となるため、しっかり復習しておくこと。また、品質評価演習は、少人数のグループ作業になるため、メンバーとしっかりコミュニケーションをとり、目的・目標をしっかりと理解し、能動的に授業に取り組むこと。理由のない遅刻や欠席は認めない。授業時数の4分の3以上出席しない者は定期試験を受験することができない。									
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	ソフトウェア品質におけるテストの役割を深く理解し、論理的に説明できる。	具体例を挙げながらテストの目的と重要性を説明できる。	テストの重要性を概念的に理解し、簡単な説明ができる。	テストの目的と重要性について部分的に説明できる。	テストの目的や重要性をほとんど理解していない。				
到達目標 B	プロジェクトに適したテスト技法を論理的に選択し、詳細な計画を立てて実行できる。	テスト技法の使用方法を理解し、シナリオに応じて適切に選択・実施できる。	各種テスト技法の目的と適用範囲を理解し、適切に選択できる。	いくつかのテスト技法について基本的な違いを説明できる。	テスト技法の違いや使用方法を理解していない。				
到達目標 C	高度なテスト計画立案、テストケース設計、適切なバグ報告ができ、他者のレビューを受けられる。	独自にテストケースや計画を作成し、バグ報告が正確かつ効果的にできる。	基本的なテストケースを設計し、簡単なテスト計画を立案・実行できる。	指導の下で一部のテスト業務を遂行できる。	テストケースの設計やバグ報告などのテスト業務を遂行できない。				
到達目標 D									
到達目標 E									
【教科書】									
ソフトウェアテスト技法練習帳 `知識を経験に変える40問`									
【参考資料】									
【成績の評価方法・評価基準】									
試験と課題、理解度確認の小テストを総合的に評価する。授業参加度、授業態度も評価に含まれる。									
※種別は講義、実習、演習のいずれかを記入。									

科目名		テスト技法			年度	2026
英語表記		Software testing skills			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	イントロダクションとテスト技法の基礎	ソフトウェアテストの重要性を理解し、テスト技法の基本概念を学ぶ	1 テストの目的と重要性	ソフトウェアテストの意義や品質保証の役割を説明できる。	1, 2	
			2 テスト技法の種類	同値分割法や境界値分析などの基本的なテスト技法の概要を理解する。		
			3 テストプロセス	テスト計画から結果報告までの流れを理解し、説明できる。		
2	同値分割法と境界値分析(1)	同値分割法と境界値分析の基本理論を理解し、簡単な問題に適用する	1 同値分割法の理解	同値分割法の概念を理解し、単純な条件に適用できる。	1, 2	
			2 境界値分析の理解	境界値分析の必要性を理解し、境界値を特定してテストケースを作成できる。		
			3 シナリオへの応用力	簡単なシステムに対して同値分割法と境界値分析を適用し、テストケースを作成できる。		
3	同値分割法と境界値分析(2)	実際のシステムに同値分割法と境界値分析を適用し、テストケース設計の技術を向上させる	1 実践的な問題への応用	複数の条件がある問題に対して同値分割法と境界値分析を適用できる。	1, 2	
			2 テストケースの網羅性の向上	テストケースが網羅的であることを判断し、不足を補う力を身につける。		
			3 結果の分析と改善点の抽出	実施したテストケースの結果を分析し、改善すべき点を発見できる。		
4	デシジョンテーブル(1)	デシジョンテーブルの理論を学び、条件分岐を視覚的に整理する方法を習得する	1 デシジョンテーブルの基本	デシジョンテーブルを用いて条件分岐を整理し、視覚化できる。	1, 2	
			2 単純な分岐条件の設計	単純な条件分岐に対して、デシジョンテーブルを適用し、テストケースを作成できる。		
			3 デシジョンテーブルの効果的な使用方法	デシジョンテーブルの利点を説明し、テストケースの効率化を考えられる。		
5	デシジョンテーブル(2)	複雑な条件分岐問題をデシジョンテーブルを用いて解決する技術を実践する	1 複雑な条件分岐への適用	複雑な条件分岐に対してデシジョンテーブルを用いてテストケースを設計できる。	1, 2	
			2 テストケースの最適化	冗長なテストケースを削減し、効率的なテスト設計を行える。		
			3 チーム内での共有と議論	作成したデシジョンテーブルを使って他者と議論し、フィードバックを反映できる。		
6	状態遷移テスト(1)	状態遷移テストの理論を学び、システムの状態遷移に基づいたテスト設計を理解する	1 状態遷移テストの基本理解	状態遷移テストの理論を理解し、状態と遷移を表現できる。	1, 2	
			2 状態図の作成	単純なシステムの状態図を作成し、システムの挙動を可視化できる。		
			3 テストケースの作成	状態図に基づいて、状態遷移に関するテストケースを設計できる。		
7	状態遷移テスト(2)	状態遷移テストを用いて、システムの複雑な動作を検証する方法を学ぶ	1 複雑なシステムの状態遷移	複雑なシステムの状態遷移に対するテストケースを設計できる。	1, 2	
			2 エッジケースの発見	予想外のエッジケースを発見し、テストに反映できる。		
			3 テスト結果の分析	状態遷移テストの結果を分析し、システムの安定性を評価できる。		
8	組合せテスト(1)	組合せテストの基本概念を理解し、複数の要素の組み合わせによるテストを設計する	1 組合せテストの基礎理解	組合せテストの目的を理解し、複数の要素の組み合わせを検討できる。	1, 2	
			2 ペアワイズテストの実践	ペアワイズテストを使用して効率的なテストケースを作成できる。		
			3 ラルフチャートの利用	ラルフチャートを活用し、組合せテストの効率を向上できる。		
9	組合せテスト(2)	実際のシナリオで組合せテストを適用し、ラルフチャートを活用したテスト設計を習得する	1 高度な組合せテスト	多様な条件の組み合わせに対応できるテストケースを作成できる。	1, 2	
			2 リスクベースの組合せテスト	リスクを考慮した組合せテストのアプローチを適用できる。		
			3 複雑なシナリオへの対応力	複雑な組み合わせ問題に対して効果的なテストケースを設計できる。		
10	総合演習	各テスト技法を統合し、実際の問題に対してテストケースを作成する能力を高める	1 各テスト技法の統合	学んだテスト技法を複合的に用いて総合的なテストケースを設計できる。	1, 2	
			2 実際のケーススタディの適用	ケーススタディを通じて実践的なテストスキルを向上させる。		
			3 テスト設計の最適化	最適なテスト設計を追求し、効率的なテストケースを作成できる。		
11	試験計画と結果報告	テスト計画の立案とテスト結果の効果的な報告方法を学ぶ	1 試験計画の策定	効果的な試験計画を立案し、スケジュールや範囲を定められる。	1, 2	
			2 テスト結果の報告方法	テスト結果を分かりやすく報告し、関係者に情報を共有できる。		
			3 課題の発見と対策	テスト結果から課題を抽出し、改善策を提案できる。		
12	演習(1)	学んだテスト技法を使って個別の問題を解決し、フィードバックを通じて理解を深め	1 基本テスト技法の実践	学んだ基本テスト技法を実際の問題に適用し、問題解決できる。	1, 2	
			2 個別フィードバックの受け入れ	フィードバックを活用し、自身のテスト設計力を向上できる。		
			3 テストケースの再評価	テストケースの適切さを評価し、必要に応じて修正できる。		
13	演習(2)	さらなる演習を通じて、異なるテスト技法の適用方法を実践的に学ぶ	1 応用テスト技法の適用	応用的なテスト技法を使って、より複雑な問題に対応できる。	1, 2	
			2 グループディスカッション	チームメンバーと意見交換し、協力してテスト設計を改善できる。		
			3 プレゼンテーション力の強化	テスト結果を整理し、他者に説明できるプレゼンテーション力を高める。		
14	演習(3)	より高度な問題に対して、複数のテスト技法を組み合わせることで解決する	1 総合的なテスト設計の実践	複数のテスト技法を統合し、包括的なテストケースを作成できる。	1, 2	
			2 高度なフィードバック対応	高度なフィードバックを理解し、テスト設計のさらなる改善を図れる。		
			3 テスト結果の分析力	テストの結果を分析し、潜在的な問題点を特定できる。		
15	演習結果のフィードバック	演習結果をフィードバックし、学びを振り返り、今後のテスト設計に活かす方法を考察する	1 成果発表と総括	演習で足りなかった部分を知り、改善できる。	1, 2	

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