

## ソフトウェア工学(Software Engineering)

担当教員名	新井 イスマイル		
学科・専攻、科目詳細	電気情報工学科 情報工学コース 5年 後期 1単位 講義		
学科のカリキュラム表	専門科目 必修科目		
共生システム工学の科目構成表	専門工学科目 専門応用系		
学習・教育目標	共生システム工学	F-1(55%) G-2(25%) H-1(20%)	
	JABEE基準1(1)	(d)(e)(h)	
科目的概要	<p>ソフトウェアの需要が年々飛躍的に高まっているのに対して、ソフトウェアの生産性を急に改善するのは困難である。いわゆる「ソフトウェア危機」である。この「危機」を回避するためには、高品質のソフトウェアを効率的に開発する手法が求められている。この科目では、高品質のソフトウェアを効率的かつ組織的に開発するために発達してきたソフトウェア工学の、種々の技法・手法を講義する。</p>		
テキスト(参考文献)	(教科書) Mint (経営情報研究会) : 「図解でわかるソフトウェア開発のすべて」、日本実業出版社		
履修上の注意	受講に当たっては、プログラミングII・III、データ構造とアルゴリズム、またはそれらに相当する科目を修得しておくことが望ましい。		
科目的達成目標	<p>この科目では、一般的なウォーターフォールモデルによる段階的なソフトウェア開発のプロセスに沿って、各段階における目標と構造化技法について学習する。その上で、近年急激に発展している、オブジェクト指向パラダイムに基づくソフトウェア開発技法についても学習する。具体的な達成目標は以下の通りである。</p> <ul style="list-style-type: none"> <li>[1] 一般的なソフトウェア開発工程を理解すること。</li> <li>[2] 要求定義・プログラム設計・プログラミングの構造化技法を理解・習得すること。</li> <li>[3] オブジェクト指向のソフトウェア開発技法について理解すること。これらのソフトウェア開発技法の学習を通じて、実践的なソフトウェア開発の能力を身につけ、実践的な問題解決能力を醸成する。</li> </ul>		
自己学習	<p>目標を達成するためには、授業以外に次の自己学習が必要である。</p> <ol style="list-style-type: none"> <li>1) 適当な例を用いて、構造化分析を行ってみること。</li> <li>2) 適当な例を用いて、構造化設計を行ってみること。</li> <li>3) 適当な例を用いて、構造化チャートを記述してみること。</li> <li>4) 適当な例を用いて、UMLのクラス図等を記述してみること。</li> <li>5) ソフトウェア開発技法の最新の動向を調査し、概要を把握すること。</li> </ol>		
目標達成度(成績)の評価方法と基準	合格の対象としない欠席条件(割合)		1/3以上の欠課
	<p>評価方法：後期中間試験(50%)、後期期末試験(50%)      評価基準：達成目標の各々で習得すべき内容を以下に示す。</p> <ul style="list-style-type: none"> <li>[1]ソフトウェアの一般的な特性、ソフトウェア開発のプロセスモデル</li> <li>[2]構造化分析、構造化設計、構造化プログラミング</li> <li>[3]オブジェクト指向パラダイム、オブジェクト指向分析・設計、UML</li> </ul> <p>以上の内容を2回の定期試験(いずれも100点満点)で出題し、得点の平均が60点以上のものを合格とする。</p>		
連絡先	ismail@itc.naist.jp		

授業の計画・内容	
<b>第1週 ソフトウェア工学の概要</b>	ソフトウェアの一般定義、分類、歴史について学ぶ。
<b>第2週 ソフトウェアのプロセスモデル</b>	ウォーターフォール・モデルやプロトタイピング・モデル、スパイラルモデル等ソフトウェアのさまざまなプロセスモデルを紹介する。
<b>第3週 プロジェクト計画と管理</b>	プロジェクト計画の作成手法、ソフトウェア見積もり技法、プロジェクト管理手法について解説する。
<b>第4週 構造化手法による分析（1）</b>	ソフトウェアに求められるニーズを明確化して記述するための構造化技法の一つである、デマルコの構造化分析について解説する。主にDFDについて理解する。
<b>第5週 構造化手法による分析（2）、構造工程</b>	DD、ミニ仕様書、構造化言語について解説する。外部設計、内部設計等、設計工程の考え方について解説する。
<b>第6週 構造化設計</b>	プログラムのモジュール構造と、各モジュール間のインターフェースを設計するための構造化技法の一つである構造化設計について解説する。
<b>第7週 構造化プログラミング</b>	プログラムモジュール内の制御構造を詳細に設計する際の構造化技法として、構造化プログラミングについて解説する。
<b>第8週 中間試験</b>	
<b>第9週 オブジェクト指向</b>	オブジェクト指向パラダイムの背景と基礎的な考え方、モデリングについて解説する。
<b>第10週 オブジェクト指向開発</b>	オブジェクト指向での開発全体の流れ、目的と利点について解説する。
<b>第11週 UML</b>	オブジェクト指向のモデリングのための共有言語であるUMLについて解説する。
<b>第12週 ソフトウェアの品質</b>	ソフトウェアの品質について考察し、設計レビューについて解説する。
<b>第13週 テスト工程と技法</b>	テストの意義やテスト工程の種類、テスト技法について解説する。
<b>第14週 種々のトピック</b>	ソフトウェア開発に関連して、ソフトウェア開発環境・ソフトウェアの部品化/再利用・3階層アプリケーション等の概念について解説する。
<b>第15週 最先端事例</b>	ソフトウェア工学の学術研究事例について紹介する。
<b>期末試験</b>	