

# **拠点大学院設立に向けた提案**

**2005年12月20日**

**(社)日本経済団体連合会**

**情報通信委員会**

**高度情報通信人材育成に関するWG**

# 目 次

目次

本提案の趣旨 .....	2
(1) 高度情報通信人材育成の必要性 .....	2
(2) 大学教育を巡るギャップ .....	2
(3) 先進的実践教育拠点の設立 .....	2
(4) 拠点の具体化に向けて .....	3
1. 拠点大学院での教育方針案 .....	4
(1) 産業界として求める高度 I T 人材像(拠点大学院修了段階) .....	4
(2) 高度 I T 人材に求める知識・スキル・コンピテンシー .....	5
(3) 拠点大学院に期待される教育体系 .....	7
(4) 拠点大学院におけるシラバス提案 .....	10
2. 拠点大学院の体制案 .....	12
(1) 拠点設置の前提となる事項 .....	12
(2) 拠点の運営において考慮すべき事項 .....	12
3. 拠点大学院に対する産業界としての支援案 .....	14

参考資料      拠点大学院カリキュラムモデル提案

## 本提案の趣旨

### (1) 高度情報通信人材育成の必要性

今や、ソフトウェアやシステム・インテグレーション技術は、パソコン、携帯電話、自動車、家電、産業機器等から、産業・行政・社会の基幹システムに至るまで活用され、わが国の中核技術として、産業全体、及び国家の競争力を支えている。しかし、現在、わが国では、プロジェクトマネージャー、アーキテクト、組込みソフトの専門家、CIO (Chief Information Officer)、情報セキュリティ人材、電子政府等を推進する上での政府のIT人材といった、これら技術を支える高度な情報通信人材の質・量の不足が危機的な状況にある。

今後のわが国の産業競争力、及び国家の発展、国家安全保障をも大きく左右する重要な国家的課題として、これら人材育成の抜本的強化が急務となっている。

### (2) 大学教育を巡るギャップ

しかし、現状、わが国では、高度IT人材の育成を担う中心的主体である、大学・大学院等の高等教育機関における取組みは、先進諸国と比較して十分なレベルに達していない。わが国では、高度IT人材の育成過程において、大学・大学院教育と就職後の企業の実践教育・実務の間には大きな溝があり、学生教育、企業内教育・研修、OJT、社会人再教育と続く、連続的な人材育成システムが構築されていない。企業においても、大学教育に関して、大学側と積極的に対話と連携・協力を図る姿勢に欠けていた面もあった。

### (3) 先進的実践教育拠点の設立

このような問題意識の下、(社)日本経済団体連合会では、2005年6月に、提言「産学官連携による高度な情報通信人材の育成強化に向けて」<sup>1</sup>をとりまとめた。本提言では、高度な情報通信人材の育成に向けて、産学官連携の下、ITの先進的実践教育拠点を設置し、実務に即した世界レベルのIT教育を実施することを提案している。

まずは、先進的実践教育拠点のモデルとなる拠点(大学院修士課程)を新たに設立し、産学官が各々、予算、インフラ、人材、ノウハウ等を集中的に投入し、海外の大学とも提携する形で、世界レベルの、わが国産業の競争力強化に寄与する人材を輩出したい。

産業界としては、関心を示していただいた大学と協力の下、このようなモデ

---

<sup>1</sup> <http://www.keidanren.or.jp/japanese/policy/2005/039/index.html> ご参照

ル拠点を、早ければ、2007年4月にプロジェクトマネージャー、及び組込みソフト等の専門家養成のための拠点を数拠点開設し、さらに早期に、CIO (Chief Information Officer)、ユーザー企業のITスペシャリスト、情報セキュリティのスペシャリスト等を養成する拠点を含め、10程度に拠点を拡大し、これらモデル拠点における成功モデルを確立し、広く各大学へ先進的実践教育の普及を図ることで、新卒段階で多数の高度IT人材を輩出できればと考えている。

#### (4) 拠点の具体化に向けて

6月の日本経団連提言では、このような拠点のフレームワークと大学、大学院教育に求めるIT知識・スキルの概要を提示した。このような拠点を大学側が設立する上で、産業界からより具体的に求める人材像とカリキュラムを提示してもらいたいとの声を多数いただいた。

そこで、日本経団連高度情報通信人材育成に関するWGでは、モデル拠点実現に向け、大学側の検討のベースとなる材料を提示するため、提言をさらに深掘りする形で、産業界として拠点到求める教育内容や教育体制など、より具体的な拠点のスペックについて検討を進めた。その結果として、ここでは、1. 拠点大学院での教育方針、2. 拠点大学院の体制、3. 拠点大学院に対する産業界としての支援 について案を提示する。

ここに提示した提案は、あくまで産業界として検討を行った結果である。当然、大学自体の教育・経営理念や置かれた状況により、多様な拠点の姿が想定される。本提案はあくまで大学側と産業界側が連携を進めていく上でのたたき台、あるいはスタート台としての位置付けであり、今後、本提案を元に対話を進め、拠点の具体化につなげてまいりたい。

## 1. 拠点大学院での教育方針案

### (1) 産業界として求める高度IT人材像(拠点大学院修了段階)

ITの市場や人材育成を巡る国際競争が激しさを増すなか、今後、新卒段階で国際標準レベルの質を備えた人材を多数輩出してもらわなければ、わが国が世界市場で生き残ることは難しい。

産業界としては、拠点大学院の修了生は、国際競争力と実務能力を備えた即戦力のIT技術者となり、企業におけるIT関連業務の中核的な担い手となることを求める。将来的には、IT分野で、わが国産業に留まらず、世界を牽引するリーダー的トップレベルの人材となることを期待する。

ソフトウェア・システムの開発・利用を巡る内外の市場動向、人材需給・育成の状況、技術革新の動向、企業の経営戦略等に照らし、産業界としては、拠点大学院修了レベルで、特に、以下のような人材を求める。

#### <産業界として求める高度IT人材像(拠点大学院修了段階)>

①要求分析、設計、プログラミング、テストなどのソフトウェアやシステムの設計開発に関する専門的なITスキルを備えるとともに、経営、工学、先端技術等幅広い専門性を備え、各種プロジェクトの全体像を理解し、マネジメントを行ったり、各分野においてITを高度に応用・利活用するための基礎を備えた人材

→将来的に、企業において、

(i) プロジェクトマネージャー(専門的なITスキルを有し、各種プロジェクトをマネジメントできる人材)、

(ii) CIO(Chief Information Officer: 高度なITスキルを有し、企業経営を実施できる人材)、

(iii) ユーザー企業におけるシステム開発要求やBPR(Business Process Reengineering)を担うスペシャリスト(専門的なITスキルを有し、ITを活用して自社の業務の効率化や事業革新を実現する人材)

を担う人材

②ソフトウェア・システム関連分野の高度な専門知識を備えたスペシャリストで、最先端技術の開発・実装、製品の品質確保・高度化等を担う人材

→将来的に、企業において、

(i) 組込みソフト、ソフトウェアエンジニア、特定技術等のスペシャリスト

(高度な専門のITスキルを有し、自社のソフトウェアの品質と生産性の向上に寄与し、国際競争力を持った商品開発ができる人材)、  
(ii)情報セキュリティのスペシャリスト(専門的な情報セキュリティに関する知識・スキルを有し、円滑な業務遂行を担保しつつ、自社の情報システムのセキュリティを確保する人材)  
を担う人材

## (2) 高度IT人材に求める知識・スキル・コンピテンシー

現状では、わが国の大学・大学院における情報工学教育と、企業が新卒者に求めるニーズとの間に大きなギャップがある。そのため、わが国では、高度IT人材の育成過程において、大学・大学院教育と就職後の企業の実践教育・実務の間には大きな溝があり、学生教育、企業内教育・研修、OJT、社会人再教育と続く、連続的な人材育成システムが構築されていない。大学・大学院では研究が重視されてきた傾向があり、大学の重要な使命である教育に対する取り組みは相対的に弱かった感がある。

産業界としては、大学・大学院と企業との間でギャップなく連続性ある人材育成システムを構築する必要があると考える。そのためには、少なくとも、新卒段階で、「①企業内の実践教育・業務に耐えうるIT、及び関連分野の高度な専門知識・スキルを身につけ、現在、企業が行っている新卒者向けのIT技術研修を受けずとも、即業務に対応できる人材」、「②企業内教育により、求める高度人材へと成長可能な基盤となる能力、素養を備えた人材」が輩出されることを期待する。企業においては、このような人材を社内教育・研修、OJT等を通じて、ITに関する技術的な専門知識、スキル、ノウハウ等を蓄積させ、世界トップレベルの高度IT人材へと育成していきたい。

そのためには、拠点大学院においては、現状でも大学・大学院で実施されている、ソフトウェア・システムに関する基礎の講義に加え、産業界からみて不足感がある、①ソフトウェア・システムの基礎の演習、応用の講義・演習、②工学、人文科学等の学際的領域をより充実させるべきではないかと考える。

具体的には、拠点大学院において、以下のような知識・スキルを学生に習得させ、修了段階において実行できるようになることを期待する。

## <拠点大学院の修了学生に期待する知識・スキルのレベル>

### ①ソフトウェア・システムの基礎、応用 の知識・スキル

- ・複数のプログラム言語を使いこなし、少なくとも1万行以上のプログラミングができる。
- ・UMLやDFD、ERDなどの各種ダイアグラミング技法を使って、簡単なシステムの要求分析や設計ができる。
- ・ハードウェア、ミドルウェア、ネットワーク、データベース、OS等のシステム開発にかかる要素技術の基本概念を理解し、利活用あるいは基本的な機能を設計開発できる。
- ・作業計画、各工程の作業内容、作業結果など、システム開発に係る知識を持ち、その手順、意義を理解し、利活用できる。
- ・品質、コスト、工程、リスクなどシステム開発の管理技術の基礎知識を持ち、その重要性、必要性を理解し、利活用できる。
- ・コンピュータ上のシミュレーションのみならず、実際にモノに触れ、実作業をした経験が少なからずあり、また、そうした経験の重要さを認識し、業務につなげることができる。
- ・要求分析、設計、プログラミング、テストを含むシステム開発プロセスを一通り経験し、各プロセスを実施可能なスキルを身につけている。
- ・実践的なプロジェクトマネジメントの知識・スキルを持ち、小規模のプロジェクトマネジメントができる。
- ・最先端の応用技術を知り、情報工学基礎との関連を理解することができる。

### ②学際的領域 の知識・スキル

社会経済全般のIT化や、技術の急速な進展を見据えて、単にITに留まらず、ITと経営、バイオ、ナノテク、機械、電気・電子等との融合領域にも適応可能な包括的な技術スキル、マネジメントスキル等の養成が必要である。

そのためには、

- ・経営、会計、法律、経済、公共政策、知的財産 等の社会科学、人文科学系の知識・スキル
- ・機械、電気・電子、通信、材料、構造、数学、物理、航空宇宙 等の工学、理学系の基礎的な知識・スキル
- ・バイオ、ナノテクノロジー 等の最先端技術の知識・スキル等の養成が必要である。

また、企業におけるIT実務の遂行において、最も求められる要素は、人間性、コミュニケーション能力、交渉能力、意欲・やる気、ストレス耐性、創意工夫、英語能力、論理的思考力、問題発見・解決能力、目標達成への推進能力、責任感、信頼感、プロジェクト遂行能力、チームワークとコラボレーション、リーダーシップ等といった基礎的な職業能力である。これら能力は、初等・中等・高等教育、企業、家庭、地域等各々が役割を果たし、養うべきものであるが、拠点大学院の教育においても、これらの能力の養成に十分考慮していただきたい。

### (3) 拠点大学院に期待される教育体系

このような卓越した知識・スキルを備えた人材を養成するため、拠点大学院では、学生の創造性と、実際のIT産業の現場で要求される実践的な技術の開発力と応用力を伸ばすための実践的な教育が望まれる。具体的には、学生に対して、講義・演習等を通じて、ソフトウェア・システムの設計開発に関わる、「技術要素」「開発技術」「管理技術」、関連分野等を一通り修得させた上で、実際のプロジェクト型システム開発の体験をさせ、企業に就職した段階で、直ちに実務に従事できる最低限の技術とスキルを修得させることを期待する。そのための教育内容、教育手法を下記の通り、提案したい。

#### ①教育内容

拠点大学院では、先に述べた「求める知識・スキル」を養成すべく、現在大学・大学院にて実施されている、基礎的なコンピュータ科学、ソフトウェア・システムの基礎理論の講義等に加え、その技術の変遷(基本原理等)についても触れ、学生が自ら技術の流れの方向を考えられるようにするとともに、下記のような教育内容を実践することが期待される。

#### <拠点大学院に期待される教育内容のモデル>

##### (i) ソフトウェア・システムの応用分野、実務領域

例えば、コンピュータハードウェア、オペレーティングシステム、プログラミング、オブジェクト指向、ソフトウェア設計、オープンシステム工学、オープンシステム工学、ソフトウェア開発方法論、ソフトウェア開発工学、ソフトウェア品質管理、要求分析、ソフトウェア構築要素技術、ITサービス管理、企業情報システム論、EA(Enterprise Architecture)、データベ

ース、分散データベース、ネットワーク(管理・運用)、組込みソフトウェア開発工学、情報セキュリティ、プロジェクトマネジメント、実プロジェクト演習、プロジェクトアーキテクチャー、プロジェクトリスクコントロール、プロセス改善 等

**(ii) 学際的領域・分野**

例えば、経営、会計、法律、経済、公共政策、知的財産、標準化戦略、金融工学、ロボット、材料、構造、数学、機械、電気・電子、化学、物理、バイオ、エレクトロニクス、電子商取引 等

さらには、ソフトウェアとこれらの専門分野を融合した分野(例えば、バイオテクノロジーとソフトウェアを組合せた「バイオテクノロジーアーキテクト」、医学とソフトウェアを組合せた「メディカルアーキテクト」、情報セキュリティと物理的セキュリティを組合せた「CSO」等) 等

**(iii) 基礎的職業能力**

例えば、英語、ビジネスリテラシー、リーダーシップ、ディベート 等

なお、上記のモデルカリキュラムを拠点大学院で実施する場合、学生が学部段階で、下記の一般教養科目、工学基礎科目、情報工学専門科目等を修得済であることを前提としている。

- ・離散数学 ・計算機概論 ・ソフトウェア概論 ・ネットワーク概論
- ・データベース概論 ・プロジェクト管理概論 ・プログラミング理論・言語・演習

組込み系の場合は、さらに下記科目を修得済であることを前提としている。

- ・物理学 ・電気回路 ・電子回路 ・論理回路 ・組込みシステム入門
- ・リアルタイムOS利活用入門 ・カーネル設計入門 ・組込みシステム開発工程入門 ・デジタル信号処理入門(DSPを含む) ・ハードウェア設計入門(SoC設計を含む)

**②教育手法**

拠点大学院では、学生がこのような実務的知識・スキルを習得できるよう、下記のような、従来にはない教育手法の採用を提案したい。

## ＜拠点大学院に期待される教育手法のモデル＞

### (総論)

- ・ 講義は座学だけでなく、討論、実習を組み合わせる
- ・ 常に、最新の技術や社会・企業ニーズを加味したテーマ選択を行う
- ・ 修士論文に開発事例を認める、もしくは卒業設計に代えることも認める
- ・ 国際化に対応するため、最大限、英語による講義、演習を取り入れる
- ・ 入学段階でアドバイザー(教員)が相談に乗り、学生が自分の関心、能力、今後のキャリアに最も適した教育を受けられるようにする。また、十分な知識、スキルの習得に至らない学生に対して、アドバイザー(教員)が改善提案をし、実践するようなサポート体制を準備する

### (演習)

- ・ 学生が知識を習得しやすくするため、演習は短期集中で実施する(同じ科目を週数回、講義演習を行う)
- ・ 演習では、役割、責任、権限を分担して行うプロジェクトベースドラッシングを取り入れる。例えば、企業システム、受託システムをグループで構築する。
- ・ 中長期にわたる国内外の企業、研究機関へのインターンシップを行う

### (外部連携)

- ・ 学際的領域・分野の教育を行うため、学部、研究科、大学間をまたがった単位取得、教員確保を行う
- ・ 企業の最先端の技術者との交流の機会(テクニカルトリップ)を設ける
- ・ 産学関係者参加の下での学生による成果発表会の開催
- ・ 教育カリキュラム検討に外部関係者の意見を取り入れる

等

勿論、ITを取り巻く環境は急激に変化を続けており、社会や企業が求める知識・スキルも多様化かつ深化を遂げている。技術革新の動向に対応して、より社会や企業のニーズに即した教育内容・教育手法について、継続的な改善が必要である。

#### (4) 拠点大学院におけるシラバス提案

拠点大学院では、学生は、自らの関心、能力、キャリアデザインに従って、以上のような教育内容に基づき、科目選択を行うことになる。拠点大学院に期待される具体的なカリキュラムの提案を別添参考資料として掲載する。

拠点大学院においては、高度IT人材の育成に向け、より効果的なカリキュラム編成を行っていただくため、下記の通り、修了に必要な単位取得の要件を提案したい。

##### ＜拠点大学院に期待される単位要件のモデル＞

###### ①ソフトウェア・システム関連の専門科目

例えば、ソフトウェア開発と実習、オープンシステム開発と演習、プロジェクトマネジメント、……を必須科目とする。

###### ②学際的領域・分野

ITサービス、企業情報、法規・会計、ビジネスリテラシー、……を必須科目とする。

###### ③インターンシップ

国内外の企業や研究機関に対する、最低1ヵ月以上のインターンシップを必須科目とする。

一例として、ビジネス系アプリケーションのプロジェクトマネジメントを専攻する学生の、後述する拠点大学院カリキュラム提案における履修モデルを下記に示す。

##### ＜プロジェクトマネジメントの専攻(大学院修士2年コース)の履修例＞

プログラミング理論

オープンシステム工学

オープンシステム工学演習

ソフトウェア開発方法論の理論と実践

プログラミング言語と演習

ソフトウェア開発工学

ソフトウェア開発工学演習  
ITサービス管理  
企業情報システム論  
法規・会計入門  
ビジネスリテラシー  
リーダーシップ  
セキュリティ基礎と情報資産保護  
プロジェクトマネジメントの理論と実践  
プロジェクトリスクマネジメントとプロセス改善  
インターンシップ

＜組み込み技術の専攻(大学院修士2年コース)の履修例＞

組み込みシステム構造  
組み込みシステム技術要素  
ユーザビリティ  
組み込みネットワーク  
組み込みシステム開発技術  
高品質ソフトウェア開発  
技術戦略  
デジタル信号処理  
ソフトウェア実装技術  
製品仕様作成技術  
組み込みOS  
プロセッサ設計開発  
製品戦略  
プロジェクト定型  
実践型プロジェクト実習  
インターンシップ

## 2. 拠点大学院の体制案

産業界としては、このような拠点の設立に向け、産学官の連携・協力の下、大学・大学院側等において、下記の事項が整備されることを期待する。

### (1) 拠点設置の前提となる事項

拠点において、産業界の協力が持続的かつ有効に機能するためには、下記の点が最も重要であると考ええる。

- 産業界のニーズも踏まえ、世界レベルの高度なIT実践教育を行うこと。
- 学長、研究科長、専攻長、教授会、大学事務局をはじめ、大学全体として、新設拠点設立の趣旨を十分に理解し、内部的なコンセンサスを形成すること。また、継続的に大学全体として拠点の運営に最大限の支援を行うこと。
- 新設拠点の設立に必要な教員及び学生の定員を必要数確保すること。また、そのために必要な経費等は大学側で確保すること。
- 学生に対する成績評価を厳格にし、出口管理を徹底するとともに、産業界側からのフィードバックを踏まえ、学生の質の向上に継続的に取り組むこと。

### (2) 拠点の運営において考慮すべき事項

産業界の支援対象決定にあたっては、以上の事項にあわせて、下記の事項についても参考材料としたい。

#### (体制)

- ・拠点は大学院修士課程を基本とする。
- ・学際教育を進めるため、関連する他学部・研究科(機械、電気、経営等)との連携がとれているか。
- ・教員人事、カリキュラム・教材の決定、カリキュラム編成、講座運営、評価など拠点の運営に関して、産業界の意向を積極的に汲みとるとともに(例えば、産業界関係者を含む評価委員会の設置等)、人材、予算、ノウハウ等内外の必要な資源を積極的に活用しているか。
- ・運営状況、内外の情勢変化等に対応して、組織、体制、カリキュラム等の改編など機動的な改善を行うか。産学官の検討の場の設置など、それを担

保する体制・機能はあるか。

- ・定期的に教育の第三者評価を実施するなど、カリキュラム・組織・運営等のチェック、改善を行う仕組みはあるか。

#### (物理的事項)

- ・新設拠点稼動に必要なインフラ(土地、施設等)、備品を整備・用意できるか。
- ・教員室、院生室、共同研究室、計算機室など、実験研究室と同等のスペースを用意できるか。

#### (教員)

- ・IT関連分野の実務経験者であるか。非常勤講師を含め、企業等外部機関からも積極的に教員を受け入れるか。
- ・身分については、専任教員と同等の権利を保障しているか(法人特任教授(仮称)と称することも可能)。非常勤講師についても、大学としてその身分を正式に認めるとともに、旅費、講義準備のための諸経費を含む必要な経費と謝金についてもできる限り負担できるか。
- ・教育、人材育成を第一の職責とし、研究は本人の希望する範囲で行えるか。
- ・専任教員が絶えず企業の最新のニーズ、ノウハウをキャッチアップする仕組みはあるか。

#### (学生)

- ・新設拠点について内外に積極的にPRし、優秀な学生の確保に努めているか。
- ・単位認定など学生のインセンティブを付与する形で、数ヶ月にわたる、より長期のインターンシップを積極的に実施するか。
- ・修士論文提出を修了の必須要件とはしない(大学評価・学位授与機構等の枠組みを活用して、修士論文なくとも学位付与を認める)。

### 3. 拠点大学院に対する産業界としての支援案

産業界としては、以上のような教育の仕組み、体制を整備し、高度IT人材育成のモデル拠点として、相応しいと判断した拠点に対して、拠点の設置、運営、評価など、下記のような支援を行う。

#### (全般)

拠点に対して、必要な政府予算等が重点的に投入されるよう、政府への政策提案、働き掛けなど、産業界として支援を行う。

#### (運営支援)

- ・企業のノウハウを提供する形で、ビジネスの実環境に基づく実践的なカリキュラム、プログラム策定支援、ソフトウェアや開発ツールを含めた教材の提供を行う。また、必要に応じて、教育関連機材・機器の提供を行う。
- ・第一線で活躍する企業人を教員として派遣する(専任教員(1拠点数名)+非常勤講師の派遣)。
- ・実績に応じて、社員の教育研修、社会人再教育の場として活用するなど多面的な連携を行う。
- ・その他、拠点の運営、評価、体制・カリキュラム等の改編など、継続的な支援を行う。

#### (教育支援)

- ・プロジェクト経験、ものづくり体験等の実践教育の場の提供、特に、学生の長期インターンシップ(海外を含む)の受け入れを行う。あわせて、教員向けのインターンシップの受け入れも行う。
- ・教員、学生に企業の実務を経験してもらうため、企業の実プロジェクトを積極的に発注する。
- ・産業界の了解の上で、学生・教員が必要とする技術調査、最先端技術応用習得等を支援する。

#### (採用)

- ・拠点の修了生について、能力に応じて、企業として採用面での優遇策を用意する。
- ・拠点に対して、採用後の人材評価のフィードバックを行う。等

本提案に対するご質問、ご意見、さらには、産学連携による拠点大学院の設置にご関心を持たれ、その具体化を希望される方は、下記まで連絡をいただければ幸いです。

**【本件に関するご連絡先】**

(社)日本経済団体連合会 産業本部 情報通信担当

電 話 : 03(5204)1500 (代表)

ファクシミリ : 03(5255)6257

E-mail : [joho@keidanren.or.jp](mailto:joho@keidanren.or.jp)

以 上

# 拠点大学院カリキュラムモデル提案

■ 一般	
コンピュータハードウェア.....	1
オペレーティングシステム.....	2
プログラミング理論.....	4
オブジェクト指向概説.....	5
ソフトウェア設計論.....	7
オープンシステム工学.....	9
オープンシステム工学演習.....	10
ソフトウェア開発方法論の理論と実践.....	11
プログラミング演習.....	12
ソフトウェア開発工学.....	13
ソフトウェア開発工学演習.....	14
IT サービス管理.....	15
企業情報システム論.....	16
法規・会計入門.....	18
ビジネスリテラシ.....	19
リーダーシップ.....	20
セキュリティ基礎と情報資産保護概要.....	22
■ 要素技術	
データベース.....	23
ネットワーク.....	24
■ セキュリティ	
情報セキュリティ.....	26
■ 組み込みソフトウェア	
オペレーティングシステム設計と実装.....	28
組み込みソフトウェア開発工学演習.....	29
■ プロジェクトマネジメント	
プロジェクトマネジメントの理論と実践.....	30
プロジェクトリスクコントロールとプロセス改善.....	32

本シラバスは、IT スペシャリストを中心に、産業界から見て現象に不足感がある科目について作成したものであり、必要性を網羅したものではない。また、全ての学生に提示された科目を全て取ることをもとめるものでもない。

本シラバスは、

- 北海道大学大学院情報科学研究科 複合情報学専攻 寄附講座
- 日本経済団体連合会 高度情報通信人材育成に関するサブWG 参加企業の新人研修
- 情報処理学会 大学の理工系学部情報系学科のためコンピュータサイエンス教育カリキュラム J97

を基にして策定した。

策定においては、別紙1に示す内容を拠点大学院入学者の知識・能力と想定した。また、次の大学および大学院のシラバスや調査資料を参考にした。

- Information and Communications University (@韓国)
- Stanford University, Computer Science Department (@米国)
- その他国内の大学・大学院約10校
- 大学などにおけるIT教育実態調査報告書(情報系学科卒業生の視点) [経済産業省委託調査]
- 教育訓練システム開発・実証成果報告書(産学協同実践的IT教育訓練支援事業)  
[経済産業省委託事業]
- 産学連携による創発型IT教育に関する調査報告書

## コンピュータハードウェア

コース概要	<ul style="list-style-type: none"> <li>論理関数の基礎および論理回路の設計と解析について学ぶ。具体的には、論理関数の単純化、論理関数の性質、基本的な組み合わせ回路、組み合わせ回路の設計、順序回路の設計について講義と演習を行う。</li> <li>コンピュータのハードウェア構成と各機能部品(制御、演算、メモリ、入出力、通信)の仕組みを講義で理解し、簡易型 RISC コンピュータの開発演習を通してシステムの設計方法を理解する。</li> </ul>
受講前提	なし
研修方法	講義および演習
期間	13 回 (週 1 講時)
到達目標	<ul style="list-style-type: none"> <li>加算などの情報処理の仕様から論理式を構成できる。</li> <li>論理関数の諸性質を理解している。</li> <li>基本的な組合せ回路の動作を理解している。</li> <li>コンピュータ・アーキテクチャ(制御、演算、メモリ、入出力、通信)について理解している。</li> <li>簡易型 RISC コンピュータの設計・構築方法を知っている。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>論理回路 (講義と演習) <ul style="list-style-type: none"> <li>論理回路とは</li> <li>論理代数と論理式</li> <li>論理関数 (論理関数の単純化、クワイン・マクラスキ法、論理関数の諸性質、種々の論理関数など)</li> <li>基本的な組合せ回路</li> <li>組合せ回路の設計</li> <li>順序回路と順序機械 (順序回路の設計手順、順序機械の最小化など)</li> </ul> </li> <li>コンピュータ・アーキテクチャ <ul style="list-style-type: none"> <li>制御アーキテクチャ</li> <li>演算アーキテクチャ</li> <li>メモリアーキテクチャ</li> <li>入出力アーキテクチャ</li> <li>通信アーキテクチャ</li> </ul> </li> <li>RISC 型コンピュータ開発演習 <ul style="list-style-type: none"> <li>汎用デジタル IC を用いた簡易型 RISC コンピュータを設計し、マシン語プログラムによる動作確認を行う。</li> </ul> </li> </ol>
関連コース	
備考	

## オペレーティングシステム

コース概要	● オペレーティング・システムの基本概念を、講義・演習・事例を通して学ぶ。
受講前提	なし
研修方法	講義中心（ページングといったシミュレータ開発などの演習ができる単元は、演習を組み込むのが望ましい）
期間	13回（週1講時）
到達目標	● オペレーティング・システムの基本概念を理解している。
コース内容	<ol style="list-style-type: none"> <li>1. オペレーティング・システムの概要 <ul style="list-style-type: none"> <li>・ オペレーティング・システムとは何か</li> <li>・ オペレーティング・システムの歴史</li> <li>・ オペレーティング・システムの概念</li> <li>・ オペレーティング・システムの構造</li> </ul> </li> <li>2. プロセス <ul style="list-style-type: none"> <li>・ プロセスの概要</li> <li>・ プロセス間通信</li> <li>・ プロセス・スケジューリング</li> </ul> </li> <li>3. メモリ管理 <ul style="list-style-type: none"> <li>・ スワッピングやページングを使用しないメモリ管理</li> <li>・ スワッピング</li> <li>・ 仮想記憶</li> <li>・ ページ置換アルゴリズム</li> <li>・ ページング・アルゴリズムのモデル化</li> <li>・ セグメンテーション</li> </ul> </li> <li>4. ファイル・システム <ul style="list-style-type: none"> <li>・ ファイル</li> <li>・ ディレクトリ</li> <li>・ ファイル・システムの実装</li> <li>・ セキュリティ</li> <li>・ 保護機構</li> </ul> </li> <li>5. 入出力 <ul style="list-style-type: none"> <li>・ 入出力ハードウェアの基本</li> <li>・ 入出力ソフトウェアの基本</li> <li>・ ディスク</li> <li>・ クロック</li> <li>・ 端末</li> </ul> </li> <li>6. デッドロック <ul style="list-style-type: none"> <li>・ 資源</li> <li>・ デッドロック</li> <li>・ デッドロックの探知と復旧</li> <li>・ デッドロックの回避</li> <li>・ デッドロックの防止</li> </ul> </li> <li>7. 事例－UNIX/Linux <ul style="list-style-type: none"> <li>・ UNIX/Linux の歴史</li> <li>・ UNIX/Linux の概要</li> <li>・ UNIX/Linux の基礎概念</li> <li>・ UNIX/Linux のシステム・コール</li> <li>・ UNIX/Linux の実装</li> </ul> </li> </ol>
関連コース	

備考	
----	--

## プログラミング理論

コース概要	<ul style="list-style-type: none"> <li>プログラミングにおける重量な概念のエッセンスを、講義と演習で習得する。</li> </ul>
受講前提	なし
研修方法	講義および演習（※演習問題の例については、参考文献を参照下さい）
期間	26回（週1講時）
到達目標	<ul style="list-style-type: none"> <li>プログラミングの基本的な理論を理解している。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>プログラムの基礎理論 <ul style="list-style-type: none"> <li>チューリング機械</li> <li>オートマトン</li> <li>計算の理論（プログラム実行時間）</li> <li>計算可能、計算困難、計算不能性</li> </ul> </li> <li>プログラミング入門 <ul style="list-style-type: none"> <li>プログラミングを始めるには <ul style="list-style-type: none"> <li>プログラムの書式、文、プログラムの実行、文の意味と構造</li> </ul> </li> <li>データの読み取りと記憶</li> <li>構造化のプログラミング</li> </ul> </li> <li>数値計算と関数 <ul style="list-style-type: none"> <li>関数、ループの作成と関数、最適値の探索、情報を配列に格納する、総和・極小・極大を求める、プログラミングのパターン、ものの並べ方と関数の特殊な性質など（詳細は参考文献を参照下さい）</li> </ul> </li> <li>プログラム言語の翻訳 <ul style="list-style-type: none"> <li>構文生成規則</li> <li>意味論</li> <li>ループプログラムの翻訳</li> <li>プログラミング言語</li> </ul> </li> <li>並列計算 <ul style="list-style-type: none"> <li>多数のプロセッサを同時に使う</li> <li>並列計算</li> <li>通信プロセス 他</li> </ul> </li> <li>機械語プログラム <ul style="list-style-type: none"> <li>プログラムのアセンブル</li> <li>プログラムの実行</li> </ul> </li> </ol>
関連コース	
備考	【参考文献】アスキー(株)『やさしいコンピュータ科学』参照

## オブジェクト指向概説

コース概要	<ul style="list-style-type: none"> <li>• Java 言語を用いて、オブジェクト指向の概念を講義と演習で学ぶ。また、UML によるモデリング表記についても学ぶ。そして、フレームワークの活用を通してオブジェクト指向設計に触れる。</li> </ul>
受講前提	<ul style="list-style-type: none"> <li>• Java 言語の文法を知っていて、簡単なプログラムが書けること。</li> </ul> <p>※参考文献①の上巻相当の知識があること。</p>
研修方法	講義および演習（※演習問題の例については、参考文献を参照下さい）
期間	13 回（週 1 講時）
到達目標	<ul style="list-style-type: none"> <li>• オブジェクト指向の概念を理解している。</li> <li>• オブジェクト指向の特徴を活かした Java プログラムを理解している。</li> <li>• オブジェクト指向の典型的なデザイン・パターンの幾つかを理解している。</li> <li>• UML の表記を理解し、小規模なシステムの設計資料(ダイアグラム)を読むことが出来る。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>1. Java 言語を通したオブジェクト指向の理解（講義） <ul style="list-style-type: none"> <li>・ クラスとインスタンス</li> <li>・ スーパークラスとサブクラス</li> <li>・ 例外</li> <li>・ インタフェース</li> <li>・ ガーベッジコレクション</li> <li>・ スレッド</li> <li>・ パッケージ</li> </ul> </li> <li>2. Java 言語を通したオブジェクト指向の理解（プログラミング演習） <ul style="list-style-type: none"> <li>・ リスト、2 分木などの簡単なデータ構造の実装などを題材に、2～3 クラス程度の小規模プログラムの作成を行い、オブジェクト指向の概念についての理解を深めると共に、Java によるオブジェクト指向プログラミングを体験する。 <ul style="list-style-type: none"> <li>- 適切なアクセス制御の設定（public, protected, private, package）</li> <li>- 等値と複製の実現方法の理解（clone, equals や関連メソッドの実装、強い一致と弱い一致の理解）</li> <li>- モニターの理解（synchronized の設定）</li> <li>- シリアライゼーションの実装</li> <li>- コレクション・インタフェースによる抽象化 など</li> </ul> </li> </ul> </li> <li>3. UML によるオブジェクト指向モデリング <ul style="list-style-type: none"> <li>・ UML とは</li> <li>・ UML が提供するビューと記法</li> </ul> </li> <li>4. デザイン・パターン <ul style="list-style-type: none"> <li>・ 参考文献③よりいくつかのパターンを取り上げて解説する。</li> <li>※クラス図や相互作用図といった UML 表記を出来る限り多用していく。</li> </ul> </li> <li>5. フレームワークを通したオブジェクト指向の理解（プログラミング演習） <ul style="list-style-type: none"> <li>・ Struts といった Java で構築されたフレームワークを活用して簡単なソフトウェアを構築し、オブジェクト指向設計についての理解を深める。 <ul style="list-style-type: none"> <li>- フレームワークの理解（ソースコード読破）</li> <li>- ※ソースコードから UML クラス図などを作成し対象フレームワークを理解する。学生の理解度によっては、クラス図などの設計資料も事前に与えてもよい。</li> </ul> </li> </ul> </li> <li>6. フレームワークの活用(システム構築)</li> </ol>
関連コース	
備考	【参考文献①】 ソフトバンク（株）Java 言語プログラミングレッスン（下巻）

【参考文献②】 ソフトバンク（株）Java プログラムデザイン

【参考文献③】 ソフトバンク（株）デザインパターン改定版

## ソフトウェア設計論

コース概要	<ul style="list-style-type: none"> <li>ソフトウェア開発での設計の目的と代表的な手法、および関連する要素知識について解説する。</li> </ul>
受講前提	<ul style="list-style-type: none"> <li>いずれかの言語で簡単なプログラミングができること。</li> <li>システム開発工程についての基礎的な知識を有すること。</li> <li>ソフトウェア開発における要求定義について理解していることが望ましい。</li> </ul>
研修方法	講義
期間	13回（週1講時）
到達目標	<ul style="list-style-type: none"> <li>ソフトウェア設計の目的を理解している。</li> <li>初歩的なソフトウェア設計が行える。</li> <li>新しい設計技術について知っている。</li> <li>より深い設計技術を習得するための基本的なスキルが身につく。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>ソフトウェア設計の目的 <ul style="list-style-type: none"> <li>ソフトウェアアーキテクチャ</li> <li>機能要求と非機能要求の実現</li> </ul> </li> <li>ソフトウェア開発における設計の位置付けと種類 <ul style="list-style-type: none"> <li>外部設計とシステム機能設計（詳細設計）</li> <li>概念設計と技術設計</li> </ul> </li> <li>設計手法 <ul style="list-style-type: none"> <li>機能中心設計</li> <li>データ中心設計</li> <li>イベント中心設計</li> <li>オブジェクト指向設計</li> <li>ユーザインターフェース設計</li> <li>プロトタイピング手法 など</li> </ul> </li> <li>設計情報の表現手法 <ul style="list-style-type: none"> <li>UML 記法、各種構造化ダイアグラミング技法 など</li> </ul> </li> <li>モジュール <ul style="list-style-type: none"> <li>モジュール分割</li> <li>結合度</li> <li>凝集度</li> <li>抽象化</li> <li>契約による設計 など</li> </ul> </li> <li>デザインパターンとフレームワーク <ul style="list-style-type: none"> <li>MVC モデル</li> <li>レイヤ構造</li> </ul> </li> <li>エラー処理（例外処理）と障害予防</li> <li>設計の評価検証</li> <li>アプリケーション種別ごとの設計 <ul style="list-style-type: none"> <li>バッチシステム</li> <li>対話型システム（TSS、2-tier C/S、3-tier C/S、Web）</li> <li>制御系システム</li> </ul> </li> <li>アプリケーション間の設計と分散アプリケーション <ul style="list-style-type: none"> <li>ミドルウェア</li> <li>オブジェクトバス、サービスバス</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>・ アプリケーション連携</li> </ul> <p>11. ソフトウェア設計に関するトピックス</p> <ul style="list-style-type: none"> <li>・ MDA、MDD</li> <li>・ アスペクト指向</li> <li>・ 依存性の注入 (Dependency Injection)</li> <li>・ アジャイル開発/リファクタリング</li> </ul>
関連コース	<ul style="list-style-type: none"> <li>● データベース設計</li> <li>● オープンシステム工学</li> <li>● ネットワーク設計</li> </ul>
備考	

## オープンシステム工学

コース概要	<ul style="list-style-type: none"> <li>● ヘテロジニアスな複合体である情報システムは、プロトコル、データ、ソフトウェア、更には人間に至るまでの相互運用性実現が要求される。この基盤を支えるオープンシステムについて、理論的側面から体系的に講義するとともに、実例として最新の情報システムの構成要素となっている主要な基盤ソフトウェア技術を取り上げ解説・議論する。</li> </ul>
受講前提	<ul style="list-style-type: none"> <li>● ソフトウェア技術(Java, HTML, XML, OS 等)に関して基礎的な知識を有することが望ましい。</li> <li>● 更に理解を深めるためには、オープンシステム工学演習も履修することが望ましい。</li> </ul>
研修方法	講義中心（講義内容に関する理解度をみるショートレポートと各テーマに応じたロングレポートを提出する）
期間	13回（週1講時）
到達目標	<ul style="list-style-type: none"> <li>● オープンシステムそのものの研究開発、及び、オープンシステム上のソフトウェア研究開発に必要な知識を習得する。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>1. オープンシステム概論(3回) <ul style="list-style-type: none"> <li>・ オープンシステム概念</li> <li>・ アーキテクチャ</li> <li>・ 歴史</li> <li>・ 得失</li> <li>・ 動向など</li> </ul> </li> <li>2. オープンシステム構成論(3回) <ul style="list-style-type: none"> <li>・ ソフトウェアの外部仕様</li> <li>・ 相互運用性</li> <li>・ オープンソース</li> <li>・ オープン標準など</li> </ul> </li> <li>3. オープンシステム要素論(4回) <ul style="list-style-type: none"> <li>・ OS</li> <li>・ Web アプリケーション <ul style="list-style-type: none"> <li>- WWW サーバの基本動作、HTTP、HTML、動的コンテンツの生成技術、WWW ブラウザの基本動作 など</li> </ul> </li> <li>・ Web サービス</li> <li>・ XML、アプリケーションサーバ</li> <li>・ アプリケーションフレームワーク</li> <li>・ セキュリティ</li> <li>・ Grid コンピューティングなど</li> </ul> </li> <li>4. オープンシステム実例(5回) <ul style="list-style-type: none"> <li>・ Linux</li> <li>・ Java/J2EE</li> <li>・ .NET Framework</li> <li>・ Apache AXIS</li> <li>・ XML アプリケーション</li> <li>・ アプリケーションサーバなど</li> </ul> </li> </ol>
関連コース	<ul style="list-style-type: none"> <li>● オープンシステム工学演習</li> </ul>
備考	講義資料を適宜配布し、その都度参考文献、参考書を示す。

## オープンシステム工学演習

コース概要	<ul style="list-style-type: none"> <li>オープンシステムを構成する主要な要素技術について具体的な解説と演習を組み合わせることで、各要素技術の本質的な理解を促し、実践的・創造的活用能力を養うことを目標とする。</li> </ul>
受講前提	<ul style="list-style-type: none"> <li>「オープンシステム工学」の受講を条件とする。</li> </ul>
研修方法	演習中心（演習成果報告(プレゼンテーション、デモなど)およびレポートを作成する)
期間	13回（週1講時）
到達目標	<ul style="list-style-type: none"> <li>オープンシステムを構成する主要な要素技術を習得する。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>オープンシステム演習概要(1回) <ul style="list-style-type: none"> <li>演習で扱う要素技術の位置づけ及び概要</li> <li>演習の進め方</li> <li>演習環境の説明など</li> </ul> </li> <li>オープンシステム基盤技術演習(9回) <ul style="list-style-type: none"> <li>プロダクトやサーバの設定なども各演習の中で実施していく</li> <li>Linux カーネル</li> <li>Java/J2SE/J2EE、.NET Framework</li> <li>Web サービス</li> <li>XML アプリケーション</li> <li>アプリケーションサーバなど</li> </ul> </li> <li>オープンシステム応用技術演習(5回) <ul style="list-style-type: none"> <li>オープンソース</li> <li>インターネットセキュリティ</li> <li>ユビキタス技術</li> <li>モバイルアプリケーションなど</li> </ul> </li> <li>討論(1回) <ul style="list-style-type: none"> <li>全体討論による獲得知識の再確認とまとめ</li> </ul> </li> </ol>
関連コース	<ul style="list-style-type: none"> <li>オープンシステム工学</li> </ul>
備考	<ul style="list-style-type: none"> <li>演習資料を適宜配布し、その都度参考文献、参考書を示す。</li> <li>各要素技術の詳細や実用について、講師や専門家と質疑応答する機会を設ける。</li> </ul>

## ソフトウェア開発方法論の理論と実践

コース概要	<ul style="list-style-type: none"> <li>システム開発におけるライフサイクルや工程および標準化の考え方を学ぶ。</li> </ul>
受講前提	<ul style="list-style-type: none"> <li>いずれかの言語で簡単なプログラミングができること。</li> <li>システム開発工程についての基礎的な知識を有すること。</li> </ul>
研修方法	講義および演習
期間	13回（週1講時）
到達目標	<ul style="list-style-type: none"> <li>システム開発における標準的な工程とWBS、およびライフサイクルについて理解している。</li> <li>開発標準(規準)とその必要性を知っている。</li> <li>部品化によるソフトウェア再利用を知っている。</li> <li>プログラムの自動生成、テストの自動実行といった支援ツールの考え方を理解している。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>システム開発の工程とプロセス改善 <ul style="list-style-type: none"> <li>システム開発の各工程、ライフサイクル（SLCP）、プロセス改善（CMMI）、工程と品質の作りこみ（プロセスQA）</li> </ul> </li> <li>工程と手順の標準化 <ul style="list-style-type: none"> <li>ソフトウェアの部品化と共通化による再利用、フレームワーク、開発標準（規準）、手順と成果物の規定（WBS）</li> </ul> </li> <li>演習 <ul style="list-style-type: none"> <li>初歩的な開発標準（ネーミングルールなど）の作成</li> <li>既存プログラムからの部品化と共通化</li> </ul> </li> <li>開発プロセス <ul style="list-style-type: none"> <li>ウォーターフォール、プロトタイプ、アジャイル（RUP、XP、TDD等）</li> </ul> </li> <li>開発支援ツール <ul style="list-style-type: none"> <li>プログラムの自動生成（CASEツール）、テストツール（Junit等）</li> </ul> </li> <li>演習 <ul style="list-style-type: none"> <li>テストツールを用いたテストファーストに基づいたプログラム開発体験</li> </ul> </li> </ol>
関連コース	
備考	

## プログラミング演習

コース概要	<ul style="list-style-type: none"> <li>与えられたプログラム設計書(ソフトウェア詳細設計書)に基づいて、アルゴリズムを意識したコード作成～テストまでのプログラム開発実習を行う(4～5日)。</li> </ul>
受講前提	<ul style="list-style-type: none"> <li>「ソフトウェア開発方法論の理論と実践」を修了していること。</li> <li>「オブジェクト指向概説」を修了している、または UML/DFD で記述された仕様書が読めること。</li> <li>各種レビュー手法について基本的な知識があること。</li> </ul>
研修方法	講義および演習
期間	13回(週1講時)
到達目標	<ul style="list-style-type: none"> <li>プログラム開発手順を理解し、プログラム設計書(ソフトウェア詳細設計書)に応じたプログラムを開発することができる。</li> <li>適切なアルゴリズム、データ構造を選択したプログラムを開発できる。</li> <li>テストケースに基づいて、単体テスト・結合テストを行うことができる。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>プログラム開発手順(プログラム設計～単体テスト・結合テスト) <ul style="list-style-type: none"> <li>モジュール仕様の作成(モジュール・レベル・デザイン)→コーディング→単体テストの一連の流れの解説・理解</li> <li>例題の仕様解説・理解</li> <li>プログラム設計書(ソフトウェア詳細設計書)の解説・理解</li> <li>プログラムレビュー・テスト技法の解説</li> <li>モジュール・レベル・デザイン <ul style="list-style-type: none"> <li>IPO、擬似コーディング等の技法の演習</li> </ul> </li> <li>ウォークスルー、インスペクションなど</li> <li>制御フローテスト、データフローテスト、ドメインテストなど</li> </ul> </li> <li>プログラム作成(演習) <ul style="list-style-type: none"> <li>開発支援ツールの紹介、机上演習 <ul style="list-style-type: none"> <li>Eclipse など</li> </ul> </li> <li>コーディング</li> <li>コーディングレビュー(机上デバッグ)</li> <li>テスト実行、工程・品質管理用数値/データの収集、評価</li> </ul> </li> <li>成果報告(プレゼンテーション)</li> </ol>
関連コース	<ul style="list-style-type: none"> <li>ソフトウェア開発方法論の理論と実践</li> <li>オブジェクト指向概説</li> </ul>
備考	<ul style="list-style-type: none"> <li>本コースの演習を行うにあたり、次の機能を使えるようにする。関連コースと内容を調整し、不足分については本コースで講義を行う。 <ul style="list-style-type: none"> <li>ライブラリの利用</li> <li>例外処理</li> <li>ファイル入出力</li> <li>並行処理(マルチスレッド)のプログラミング</li> <li>ネットワークプログラミング</li> </ul> </li> <li>開発するプログラムは、数千ステップぐらいになるものを用意する。</li> <li>標準化や開発環境の整備などの実務者と、質疑応答する機会を設けていくのが望ましい。</li> </ul>

## ソフトウェア開発工学

コース概要	<ul style="list-style-type: none"> <li>大規模化、複雑化する情報システム構築には多くの要素技術のみならず、アーキテクチャやインテグレーションについての高度な知識と技術を要する。また、開発の側面においても、品質保証、開発プロジェクトの運営、契約など多岐にわたる課題がある。本講義では、これら課題に対し、大きく4つのテーマで講義を行い、次世代の情報技術と構築プロジェクトに対処できる実践的知識を習得する。</li> </ul>
受講前提	<ul style="list-style-type: none"> <li>基本的なソフトウェア工学の知識を有すること。</li> </ul>
研修方法	講義中心（講義内容に関する理解度をみるショートレポートと各テーマに応じたロングレポートを提出する）
期間	13回（週1講時）
到達目標	<ul style="list-style-type: none"> <li>次世代の情報技術と構築プロジェクトに対処できる実践的知識を習得する。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>大規模システム構築論(4回) <ul style="list-style-type: none"> <li>ソフトウェア開発プロジェクト</li> <li>ソフトウェアの見積もり</li> <li>統一プロセスなど</li> </ul> </li> <li>ソフトウェア要求分析 <ul style="list-style-type: none"> <li>ソフトウェア要求の調査と分析</li> <li>要求分析のための各種ダイアグラミング技法(UML、DFD、ERD等)</li> <li>システム要求の定義</li> </ul> </li> <li>ソフトウェア/システム品質論(4回) <ul style="list-style-type: none"> <li>ソフトウェア品質保証 <ul style="list-style-type: none"> <li>変更管理、障害管理、成果物管理 など</li> </ul> </li> <li>ソフトウェアテスト（総合テスト）</li> <li>テスト技法</li> <li>システム運用・保守 <ul style="list-style-type: none"> <li>インフラ管理（障害管理、ストレージ管理、ネットワーク管理など）</li> <li>サービス管理(可用性管理、自律性管理など)</li> <li>アプリケーション管理（パフォーマンス管理、トランザクション分析など）</li> <li>セキュリティ管理</li> <li>ソフトウェア資産管理</li> </ul> </li> <li>セキュリティなど</li> </ul> </li> <li>ITインテグレーション論(3回) <ul style="list-style-type: none"> <li>システム/ソフトウェアアーキテクチャ</li> <li>OLTPと無停止システム</li> <li>RASIS</li> <li>開発事例など</li> </ul> </li> <li>開発プロジェクトク運営論(4回) <ul style="list-style-type: none"> <li>ソリューション営業技術</li> <li>提案技術</li> <li>契約/ソフトウェア会計基準など</li> </ul> </li> </ol>
関連コース	<ul style="list-style-type: none"> <li>ソフトウェア開発工学演習</li> </ul>
備考	<ul style="list-style-type: none"> <li>更に理解を深めるためには、オープンシステム工学演習も履修することが望ましい。</li> <li>講義資料を適宜配布し、その都度参考文献、参考書を示す。</li> </ul>

## ソフトウェア開発工学演習

コース概要	<ul style="list-style-type: none"> <li>● 実際の社会で利用される情報システムの開発を題材とし、チーム開発による実践的な一貫型の開発実習を行う。本実習では、プロジェクトチームが仮想の顧客から RFP を受け取ることから始まり、ソリューション提案、工程/スケジュール策定、分析・テスト、プロジェクト運営、を各人の役割とその連携によって進める。</li> </ul>
受講前提	<ul style="list-style-type: none"> <li>● 「ソフトウェア開発工学」の受講を条件とする。</li> <li>● 「プログラミング演習」の受講が望ましい。</li> <li>● 「リーダーシップ」の受講が望ましい。</li> </ul>
研修方法	演習中心（テーマ毎にショートレポートを作成する。また演習成果として設計・開発成果物の作成や、プレゼンテーション・デモンストレーションを実施する）
期間	13 回（週 1 講時）
到達目標	<ul style="list-style-type: none"> <li>● 実際の開発現場で直面する種々の問題への対処能力を実体験として習得し、次世代の情報技術者に必要な創造的、実践的なスキルを習得する。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>1. 大規模システム分析・設計・実装・テスト(6回) <ul style="list-style-type: none"> <li>・ ソフトウェア開発の各工程（要求分析、設計、実装、テスト）における要素技術、ダイアグラミング技術（主にUML、DFD、ERD）について簡単な導入演習後、各人が主体的に実践する。</li> </ul> </li> <li>2. IT インテグレーション(4回) <ul style="list-style-type: none"> <li>・ ハードウェア/ソフトウェア/ミドルウェアの高度利用技、システム全体のアーキテクチャ設計/実装、及び開発環境構築を実践する。</li> </ul> </li> <li>3. プロジェクト運営(3回) <ul style="list-style-type: none"> <li>・ チーム内での役割（プロジェクトマネージャ/要求定義/設計/実装/品質保証）に基づく責任/作業分担により、全体のゴールを達成すべくプロジェクト運営を実践する。</li> </ul> </li> <li>4. コミュニケーション/プレゼンテーション(2回) <ul style="list-style-type: none"> <li>・ チーム内あるいは対顧客とのコミュニケーション、提案プレゼンテーション、成果物報告会を実践する。</li> </ul> </li> </ol>
関連コース	<ul style="list-style-type: none"> <li>● ソフトウェア開発工学</li> </ul>
備考	<ul style="list-style-type: none"> <li>● 本演習は IT 企業の第一線技術者の協力の下に実施する。</li> <li>● 演習資料を適宜配布し、その都度参考文献、参考書を示す。</li> <li>● 各要素技術の詳細や実用について、講師や専門家と質疑応答する機会を設ける。</li> </ul>

## IT サービス管理

コース概要	<ul style="list-style-type: none"> <li>コンピュータシステムとその運用管理を、業務の遂行を手助けする「IT サービス」ととらえ、サービスを要求に応じて適切に提供し、また、高い投資効果で継続的に改善するための方法を解説する。</li> </ul>
受講前提	なし
研修方法	講義
期間	13回（週1講時）
到達目標	<ul style="list-style-type: none"> <li>コンピュータ運用管理業務の体系を知っている。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>ITIL 概要</li> <li>サービスサポートその1 <ul style="list-style-type: none"> <li>サービスデスクの役割とインシデント対応プロセス <ul style="list-style-type: none"> <li>サービスデスク</li> <li>インシデント管理</li> <li>問題管理(障害発生時の対応も含む)</li> </ul> </li> </ul> </li> <li>サービスサポートその2 <ul style="list-style-type: none"> <li>変更プロセスとリリース対応、システム構成の維持 <ul style="list-style-type: none"> <li>構成管理</li> <li>変更管理</li> <li>リリース管理</li> <li>品質保証・品質管理（縮退テストといった現行保証を中心に）</li> <li>ライブラリ管理</li> </ul> </li> </ul> </li> <li>サービスデリバリーその1 <ul style="list-style-type: none"> <li>ITサービスの要件定義とコスト管理 <ul style="list-style-type: none"> <li>サービスレベル管理</li> <li>ライフサイクル管理（EOSL など）</li> <li>ITサービス財務管理</li> </ul> </li> </ul> </li> <li>サービスデリバリーその2 <ul style="list-style-type: none"> <li>ITサービスインフラの管理と災害リスク対応 <ul style="list-style-type: none"> <li>キャパシティ管理</li> <li>可用性管理</li> <li>ITサービス継続性管理</li> </ul> </li> </ul> </li> <li>ITIL の導入と実践 <ul style="list-style-type: none"> <li>プロジェクトの理解</li> <li>ビジョンの設定</li> <li>現状の把握</li> <li>目標の設定</li> <li>実現手段の検討</li> <li>成果の測定</li> <li>継続的な改善</li> </ul> </li> <li>ITIL の具体的事例</li> </ol>
関連コース	
備考	

## 企業情報システム論

コース概要	<ul style="list-style-type: none"> <li>代表的な企業情報システムに関する概要と、代表的パッケージや企業での実例等について講義する。また、企業システムの責任者である CIO の役割についても解説する。</li> </ul>
受講前提	なし
研修方法	講義
期間	13 回 (週 1 講時)
到達目標	<ul style="list-style-type: none"> <li>主要な企業情報システムと、その役割・構成・活用法を知る。</li> <li>CIO の役割、業務、責任を知る。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>経営と IT の関連について <ul style="list-style-type: none"> <li>IT 経済とは</li> </ul> </li> <li>e ビジネス時代の情報システムの動向</li> <li>基幹業務系システム <ul style="list-style-type: none"> <li>販売管理</li> <li>在庫／購買管理</li> <li>生産計画／管理</li> <li>財務会計</li> <li>管理会計</li> <li>人事管理</li> <li>ERP(Enterprise Resorce Planning) など</li> </ul> </li> <li>企業間生産・流通システム <ul style="list-style-type: none"> <li>SCM(supply Chain Management)</li> <li>SCP(Supply Chain Planning),</li> <li>CALS(Commerece At Light Speed)</li> <li>EDI(Electronic Data Interchange) など</li> </ul> </li> <li>企業間電子商取引 <ul style="list-style-type: none"> <li>BtoB</li> <li>BtoC</li> <li>CtoC</li> <li>EAI(Enterprise Application Integration) など</li> </ul> </li> <li>営業関連情報システム <ul style="list-style-type: none"> <li>CRM(Customer Relationship Management)</li> <li>SFA(Sales Force Automation)</li> <li>KBS(Knowledge Based Sales) など</li> </ul> </li> <li>製造情報管理システム <ul style="list-style-type: none"> <li>PLM (Products Lifecycle Management)</li> <li>PDM(Product Data Management)</li> <li>CIM(Computer Integrated Manufacturing)</li> </ul> </li> <li>情報検索システム <ul style="list-style-type: none"> <li>DHW(Data WareHouse)</li> <li>DM(Date Mining)</li> </ul> </li> <li>知識共有システム <ul style="list-style-type: none"> <li>KM(Knowledge Management)</li> <li>GW(Groupware)</li> <li>EIP(Enterprise Information Portal)</li> </ul> </li> <li>CIO(Chief Information Officer)とは <ul style="list-style-type: none"> <li>CIO の役割・責任・権限</li> <li>システムの経済性</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>▪ リスクマネジメント</li> <li>▪ システム監査</li> <li>▪ 情報セキュリティ監査</li> <li>▪ BPR(Business Process Reengineering)</li> <li>▪ EA(Enterprise Architecture),</li> <li>▪ TCO(Total Cost of Ownership)</li> </ul>
関連コース	
備考	

## 法規・会計入門

コース概要	<ul style="list-style-type: none"> <li>システム開発と知的財産権、著作権と特許、問題を起こさない契約方法、プロジェクト・マネジメントとコンプライアンス、情報漏洩などについて、事例をもとに体系と概要を学ぶ。</li> <li>プロジェクトを計数管理するための基礎知識として、簿記・会計の基本概念を学ぶ。</li> </ul>
受講前提	なし
研修方法	講義中心
期間	13回（週1講時）
到達目標	<ul style="list-style-type: none"> <li>ソフトウェアの契約に関して、どのような法律が関係しているのか知っている。</li> <li>ソフトウェア開発に従事するメンバーに対してどのような労務関係の法律が関係しているのか知っている。</li> <li>セキュリティ関連に関して、どのような法律が関係しているのか知っている。</li> <li>知的財産権の基本的な考え方および関連法規を知っている。</li> <li>簿記・会計の基本概念を知っている。</li> </ul>
コース内容	<p><b>【法規入門】</b></p> <ol style="list-style-type: none"> <li>システム開発とコンプライアンス</li> <li>契約関連の法規</li> <li>労務関連の法規</li> <li>知的財産権関連の法規             <ul style="list-style-type: none"> <li>知的財産とは</li> <li>特許</li> <li>商標</li> <li>実用新案</li> <li>意匠</li> <li>著作権</li> <li>不正競争防止法</li> </ul> </li> </ol> <p><b>【会計入門】</b></p> <ol style="list-style-type: none"> <li>会計と会計学</li> <li>会計の基本概念（資産、負債、資本、収益、費用、利益）</li> <li>簿記の仕組み（取引、勘定科目、仕訳、元帳）</li> <li>財務諸表の作成             <ul style="list-style-type: none"> <li>貸借対照表</li> <li>損益計算書</li> </ul> </li> <li>製造原価の計算</li> <li>財務諸表の分析</li> <li>管理会計</li> </ol>
関連コース	
備考	

## ビジネスリテラン

コース概要	<ul style="list-style-type: none"> <li>● ビジネスに欠くことのできないパーソナルスキルとして、論理的な思考や表現、コミュニケーション、交渉、問題解決のための基礎的なフレームワークと手法を学ぶ。</li> </ul>
受講前提	なし
研修方法	講義と演習
期間	13回（週1講時）
到達目標	<ul style="list-style-type: none"> <li>● 論理的な思考方法と、その方法を活用した問題解決の一連の流れを知っている。</li> <li>● コミュニケーションの重要性を理解し、コミュニケーションのポイントを知っている。</li> <li>● プレゼンテーションに必要とされる技術とその使い方を知っている。</li> <li>● ネゴシエーションに必要とされる技術とその使い方を知っている。</li> <li>● ビジネスにおける問題解決と、その一連の流れを知っている。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>1. クリティカル・シンキング <ul style="list-style-type: none"> <li>・ 論理の組み立て</li> <li>・ 論理の構造化</li> <li>・ ケーススタディ</li> <li>・ 事象の構造化</li> <li>・ 複雑な事象の構造化</li> <li>・ ケーススタディ</li> </ul> </li> <li>2. コミュニケーション <ul style="list-style-type: none"> <li>・ コミュニケーションプロセスとポイント</li> <li>・ コミュニケーション実習</li> <li>・ 双方向コミュニケーション</li> <li>・ 組織のコミュニケーション</li> <li>・ ケーススタディ</li> </ul> </li> <li>3. プレゼンテーション <ul style="list-style-type: none"> <li>・ プレゼンテーションプログラムの作成 <ul style="list-style-type: none"> <li>- 目的と意図</li> <li>- 内容の絞込みと組み立て</li> <li>- 時間配分</li> </ul> </li> <li>・ プレゼンテーション実習</li> </ul> </li> <li>4. ネゴシエーション <ul style="list-style-type: none"> <li>・ 交渉・議論の基本</li> <li>・ 交渉の戦術</li> <li>・ 傾聴</li> <li>・ 質問</li> <li>・ 検証</li> <li>・ 議論</li> <li>・ 意思伝達</li> </ul> </li> <li>5. 問題解決力 <ul style="list-style-type: none"> <li>・ 問題の発見から解決策、意思決定まで <ul style="list-style-type: none"> <li>- 問題の発見</li> <li>- 解決策の考案</li> <li>- 意思決定</li> <li>- 問題の掘り下げ</li> </ul> </li> <li>・ 解決策の考案</li> </ul> </li> </ol>
関連コース	
備考	

## リーダーシップ

<p>コース概要</p>	<ul style="list-style-type: none"> <li>● まず、リーダーシップを発揮するための基本スキルとして、ファシリテーションとコーチングについて学ぶ。そして、演習を中心に、「共通の目標」「協働の意思」「コミュニケーション」「信頼と挑戦」などの重要性や具体的なアクションを理解し、チームとして全員が協働する意識をもって活動し、達成感を得る経験をする。</li> <li>● チームで仕事をする難しさやリーダーシップを理解した上で、人的資源管理の要素を学び、様々なケーススタディを通して、管理・監督者として現場を統率するための実践的なスキルを磨く。</li> </ul>
<p>受講前提</p>	<ul style="list-style-type: none"> <li>● 「ビジネスリテラシ」の受講が望ましい。</li> </ul>
<p>研修方法</p>	<p>講義と演習</p>
<p>期間</p>	<p>13回（週1講時）</p>
<p>到達目標</p>	<ul style="list-style-type: none"> <li>● チームで仕事をする難しさを知っている。</li> <li>● チームメンバーとして適切な行動をとることができる。</li> <li>● リーダに要求されるスキルを知っている。</li> <li>● リーダーシップを発揮するための基本スキルを習得している。</li> <li>● 小規模なチームのリーダを務めることができる。</li> </ul>
<p>コース内容</p>	<ol style="list-style-type: none"> <li>1. ファシリテーション             <ul style="list-style-type: none"> <li>・ 基本スキル（質問・発言・要約、話を聴く・話を引き出す、記録、グループ調整、コンセンサス・意思決定）</li> <li>・ ファシリテーションの実践（キックオフ、進捗会議など）</li> </ul> </li> <li>2. コーチング             <ul style="list-style-type: none"> <li>・ 基本スキル（関わり方、傾聴、質問、強化）</li> <li>・ コーチングの実践</li> </ul> </li> <li>3. チームビルディング             <ul style="list-style-type: none"> <li>・ チームワーク演習</li> <li>・ チームメンバーの多様性のプラス/マイナス効果</li> <li>・ チームビルディングを効果的に進める秘訣</li> <li>・ 効果的なチーム・ミーティングの在り方</li> <li>・ チームでの意思決定</li> <li>・ チームによる問題解決</li> <li>・ チームリーダとしての働き方</li> </ul> </li> <li>4. 人的資源管理             <ul style="list-style-type: none"> <li>・ 戦略、環境と組織構造</li> <li>・ 人材開発とキャリア設計</li> <li>・ 報酬とインセンティブ</li> <li>・ 業績評価</li> <li>・ 組織学習</li> <li>・ 組織変革と人的資源</li> </ul> </li> <li>5. リーダーシップ開発             <ul style="list-style-type: none"> <li>・ 目標管理能力</li> <li>・ 問題解決能力</li> <li>・ 組織・チームワーク能力</li> <li>・ コミュニケーション能力</li> <li>・ 部下指導・動機付け・育成能力</li> <li>・ 自己啓発と自己管理問題の発見</li> </ul> </li> </ol>

関連コース	
備考	

## セキュリティ基礎と情報資産保護概要

コース概要	<ul style="list-style-type: none"> <li>● コンピュータを利用する上での脅威と、その対処方法に関する知識を習得する。</li> <li>● コンピュータを利用する上で守るべきモラルに関する知識の修得を習得する。</li> </ul>
受講前提	なし
研修方法	講義中心
期間	13回（週1講時）
到達目標	<ul style="list-style-type: none"> <li>● コンピュータウイルスへの対策として必要なことを理解している。</li> <li>● 不正アクセスへの対策として必要なことを理解している。</li> <li>● 情報を扱う上で守るべきモラルを理解している。</li> <li>● セキュリティポリシーの意義を理解している。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>1. 情報セキュリティの重要性 <ul style="list-style-type: none"> <li>・ 情報化の進展と情報セキュリティ</li> <li>・ コンピュータ活用上の脅威</li> <li>・ 実施すべきセキュリティ対策</li> </ul> </li> <li>2. コンピュータウイルス対策 <ul style="list-style-type: none"> <li>・ コンピュータウイルスとは</li> <li>・ 代表的なワクチン</li> <li>・ 予防対策ガイドライン</li> <li>・ ウイルス発生時の対処</li> </ul> </li> <li>3. 不正アクセス対策 <ul style="list-style-type: none"> <li>・ 不正アクセスの脅威</li> <li>・ 不正アクセスへの対策</li> <li>・ パスワードの活用</li> <li>・ ファイルアクセス権の活用</li> <li>・ 暗号技術によるセキュリティ対策</li> <li>・ ソーシャルエンジニアリングへ</li> </ul> </li> </ol>
関連コース	
備考	<ul style="list-style-type: none"> <li>● NECラーニング(株) 自習 CD-ROM 教材「必須！！セキュリティ基礎知識」使用  <a href="http://www.sw.nec.co.jp/el/teach_material/course2002/CB033.html">http://www.sw.nec.co.jp/el/teach_material/course2002/CB033.html</a> </li> </ul>

## データベース

コース概要	<ul style="list-style-type: none"> <li>現在一般的に使用されているリレーショナルデータベースを例に、データベース管理システムの基本的な機能を理解し、その操作方法・設計方法・管理方法を講義および演習を通して習得する。</li> </ul>
受講前提	なし
研修方法	講義および演習
期間	13回（週1講時）
到達目標	<ul style="list-style-type: none"> <li>データベース管理システムの基本機能を理解する。</li> <li>データベースシステム構築に必要な各フェーズ（論理設計、物理設計、実装など）の作業を行うことができる。</li> </ul>
コース内容	<p>※講義による概念・機能の理解に加え、実際に自ら環境を構築し操作を行うことで実践的な技術を習得する。また論理設計においては具体的な題材を取り上げ、自ら考えて設計することを主眼とする。</p> <ol style="list-style-type: none"> <li>データベース管理システムの基本機能 <ul style="list-style-type: none"> <li>トランザクション制御</li> <li>排他制御</li> <li>障害回復</li> <li>アクセス制御</li> </ul> </li> <li>データの操作方法(SQLの基礎) <ul style="list-style-type: none"> <li>DDL</li> <li>DML</li> </ul> </li> <li>リレーショナルデータベースのモデルと設計方法（概念設計、論理設計） <ul style="list-style-type: none"> <li>リレーショナルデータベースモデルの概念</li> <li>エンティティの抽出</li> <li>リレーションシップの作成</li> <li>正規化</li> <li>論理設計とER図</li> </ul> </li> <li>リレーショナルデータベースの実装 <ul style="list-style-type: none"> <li>物理設計</li> <li>インデックス</li> <li>データベースのパフォーマンス</li> <li>データのバックアップ/リカバリ</li> </ul> </li> </ol>
関連コース	
備考	

## ネットワーク

<p>コース概要</p>	<ul style="list-style-type: none"> <li>実践的な TCP/IP ネットワークの知識・スキルの習得を目指す。第一段階は、インターネットの歴史と構造を示し、ネットワーク通信の仕組みを理解する。第二段階では、レイヤ 2 ネットワークを理解する上で必要な基本的な知識や基本プロトコルについて理解する。また、知識の定着を図るため、単一ネットワークセグメント(レイヤ 2)による実際のネットワークを構築し、TCP/IP や arp プロトコルを理解する。第三段階は、経路制御などのレイヤ 3 ネットワークの設計・構築に必要な基本的な知識を習得し、その実践として実際にルータを使用したレイヤ 3 ネットワークを設計・構築する。実習を通じて RIP、OSPF などのルーティングプロトコルの仕組みを理解し、LAN-WAN の接続形態の知識定着をはかる。最後にネットワークの新しい技術について学習する。</li> </ul>
<p>受講前提</p>	<ul style="list-style-type: none"> <li>情報通信についての基礎的な知識を有すること。</li> </ul>
<p>研修方法</p>	<p>講義および演習</p>
<p>期間</p>	<p>13 回 (週 1 講時)</p>
<p>到達目標</p>	<ul style="list-style-type: none"> <li>ネットワークアーキテクチャの原理還俗を理解している。</li> <li>小規模 LAN 環境(レイヤ 2 およびレイヤ 3)の設計、構築、トラブルシューティングができる。</li> </ul>
<p>コース内容</p>	<ol style="list-style-type: none"> <li>インターネットとネットワーク通信の仕組み             <ul style="list-style-type: none"> <li>インターネットの歴史</li> <li>インターネットの構造(AS について)</li> <li>IP アドレス(ネットマスク、グローバルアドレス、プライベートアドレス)</li> <li>名前解決(ネームサーバ(DNS))の仕組み</li> <li>ネットワークアプリケーションの動作の理解 ※例えば、Web ブラウザで HP にアクセスした場合、FQDN が IP アドレスへ名前解決されてアクセスされること</li> </ul> </li> <li>レイヤ 2 ネットワークを理解する。             <ul style="list-style-type: none"> <li>基本知識、基本プロトコル(MAC アドレス、TCP/IP)</li> <li>レイヤ 2 プロトコル(arp)</li> <li>ネットワークを構成する機器、配線のそれぞれの役割・特徴</li> <li>(演習)単一ネットワークセグメントネットワークを構築し、TCP/IP や arp などのプロトコルの動作を理解する</li> </ul> </li> <li>レイヤ 3 ネットワークの設計の基本的知識の習得             <ul style="list-style-type: none"> <li>一般的な経路制御(RIP、OSPF 等のルーティングプロトコル)</li> <li>ネットワークのトポロジー(レイヤ 3)</li> <li>現在使われているネットワークサービスについて(ADSL、光回線等)</li> <li>LAN-WAN 接続の設計</li> <li>プロバイダの経路制御(BGP、経路集約)</li> <li>(演習)ルータを利用し、レイヤ 3 ネットワークの設計・構築を実際に行い、機器やルーティングプロトコルの動作を理解する</li> <li>(演習)冗長化や負荷分散を目的としたネットワークの設計・構築を行い、その動作を確認する</li> </ul> </li> <li>新しいネットワーク技術について学習する。             <ul style="list-style-type: none"> <li>NAT</li> <li>IP v 6</li> <li>IPsec</li> <li>無線 LAN</li> <li>QoS(優先制御)</li> <li>VoIP</li> </ul> </li> </ol>

関連コース	
備考	

## 情報セキュリティ

コース概要	<ul style="list-style-type: none"> <li>情報セキュリティを取り巻く環境や情報セキュリティシステムを構成する要素技術について解説する。後半部では、情報セキュリティにおける脅威とその対処法を体得するための演習を行う。</li> </ul>
受講前提	<ul style="list-style-type: none"> <li>インターネット／ネットワークの基礎知識を有すること。</li> <li>OS や計算機に関する基礎知識を有すること。</li> </ul>
研修方法	講義および演習
期間	13 回（週 1 講時）
到達目標	<ul style="list-style-type: none"> <li>情報セキュリティの基本的な知識を理解している。</li> <li>脅威から防御するために必要な一通りの知識・技術を習得している。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>情報セキュリティを取り巻く環境 <ul style="list-style-type: none"> <li>情報社会の流れ</li> <li>情報セキュリティの必要性</li> <li>関連法規</li> </ul> </li> <li>セキュリティとは <ul style="list-style-type: none"> <li>機密性</li> <li>完全性</li> <li>可用性</li> </ul> </li> <li>リスクとは <ul style="list-style-type: none"> <li>脅威</li> <li>脆弱性</li> <li>リスク分析</li> </ul> </li> <li>認証・アクセス制御 <ul style="list-style-type: none"> <li>パスワード</li> <li>証明書</li> <li>SSO</li> </ul> </li> <li>暗号技術 <ul style="list-style-type: none"> <li>共通鍵暗号方式</li> <li>公開鍵暗号方式</li> <li>PKI</li> </ul> </li> <li>物理セキュリティ <ul style="list-style-type: none"> <li>入退室管理</li> <li>バックアップ</li> <li>災害対策</li> </ul> </li> <li>システムセキュリティ <ul style="list-style-type: none"> <li>OS、セキュア OS</li> <li>パッチ</li> <li>アカウント・権限管理</li> </ul> </li> <li>ネットワークセキュリティ <ul style="list-style-type: none"> <li>VLAN、VPN、SSH、SSL/TLS、NAT、routing</li> <li>FW、IDS、IPS</li> <li>WLAN</li> </ul> </li> <li>アプリケーションセキュリティ <ul style="list-style-type: none"> <li>WWW、mail、DNS</li> <li>DB</li> <li>セキュアプログラミング</li> </ul> </li> </ol>

	<p>10. セキュリティ運用</p> <ul style="list-style-type: none"> <li>・ ログ監視</li> <li>・ 事故対応</li> <li>・ ポリシー見直し</li> </ul> <p>11. 組織のセキュリティ</p> <ul style="list-style-type: none"> <li>・ リスク分析</li> <li>・ セキュリティポリシー</li> <li>・ コンプライアンス</li> <li>・ 体制</li> </ul> <p><b>【演習】</b></p> <p>1. 1. 脅威の体験</p> <ul style="list-style-type: none"> <li>・ 各種ツールを使用し、暗号化されていないネットワーク、セキュリティ対策をしていないサーバに侵入し、情報収集、データの改ざんなどを行い、脅威が情報資産にどのような影響を及ぼすのかを体験する。</li> </ul> <p>2. 2. 脅威への対策</p> <ul style="list-style-type: none"> <li>・ 暗号化通信、アクセス制御、適切な権限の割り当てなどにより、上記脅威を低減できることを確認する。また、PKI を使用した簡単なシステム構築、セキュリティアプライアンス (IDS、 IPS など) を使用したセキュリティ対策も実践する。</li> </ul>
<p>関連コース</p>	
<p>備考</p>	

## オペレーティングシステム設計と実装

コース概要	<ul style="list-style-type: none"> <li>組み込みシステムに用いられるリアルタイム OS とその中心技術について解説する。また、Linux などの OS のソースコードを解析し、OS の挙動を理解する。</li> </ul>
受講前提	<ul style="list-style-type: none"> <li>オペレーティング・システムの基本概念を理解していること。</li> <li>「オペレーティングシステム」を受講していることが望ましい。</li> </ul>
研修方法	講義および演習
期間	13 回（週 1 講時）
到達目標	<ul style="list-style-type: none"> <li>リアルタイムシステムにおけるプロセス制御を理解している。</li> <li>デバイスドライバの概念や構成を理解している。</li> <li>特定の OS の実装(ソースコード)を知っている。</li> <li>特定の OS の実装を改変したり、OS を直接操作することが出来る。</li> </ul>
コース内容	<p>※OS の挙動を具体的に理解するために用いる特定 OS は、リアルタイム OS であることが望ましいが、MINIX など教育に適した規模や機能のものでもよい。</p> <ol style="list-style-type: none"> <li>システム L S I 構成技術</li> <li>リアルタイム OS             <ul style="list-style-type: none"> <li>概要・特徴</li> <li>リアルタイム OS の紹介                 <ul style="list-style-type: none"> <li>ITRON</li> <li>リアルタイム処理用 Linux など</li> </ul> </li> </ul> </li> <li>割り込み処理（システムコール）</li> <li>プロセス             <ul style="list-style-type: none"> <li>プロセスとは</li> <li>プロセス間通信</li> <li>プロセススケジューリング</li> <li>特定 OS におけるプロセスの概要</li> <li>特定 OS におけるプロセスの実現</li> </ul> </li> <li>入出力             <ul style="list-style-type: none"> <li>入出力ハードウェアの原理</li> <li>入出力ソフトウェアの原理</li> <li>デッドロック</li> <li>特定 OS における入出力の概要</li> <li>特定 OS におけるブロック型デバイス</li> <li>RAM ディスク</li> <li>ディスク</li> <li>クロック</li> <li>端末</li> <li>特定 OS におけるシステムタスク</li> </ul> </li> <li>メモリ管理             <ul style="list-style-type: none"> <li>特定 OS におけるメモリ管理の概要</li> <li>特定 OS におけるメモリ管理の実現</li> </ul> </li> <li>演習             <ul style="list-style-type: none"> <li>ソースコードやエミュレータを使用し、様々なデバイス・ドライバを制御する</li> <li>簡易 OS を作成する など</li> </ul> </li> </ol>
関連コース	<ul style="list-style-type: none"> <li>オペレーティングシステム</li> </ul>
備考	【参考文献】ピアソン・エディケーション(株)

『オペレーティングシステム』設計と理論および MINIX による実装

**組み込みソフトウェア開発工学演習**

コース概要	<ul style="list-style-type: none"> <li>● 実際の社会で利用される組み込みソフトウェアの開発を題材とし、チーム開発による実践的な一気通貫型の開発実習を行う。本実習では、プロジェクトチームが組み込みシステムの開発提案を受けとることから始まり、工程/スケジュール策定、分析-テスト、ハードウェア開発チーム（アーキテクト）との交渉、プロジェクトク運営、を各人の役割とその連携によって進める。</li> </ul>
受講前提	<ul style="list-style-type: none"> <li>● アセンブラ、C といったハードウェア寄りの言語で、メモリ操作のプログラムが記述できること。</li> <li>● 「ソフトウェア開発工学」の受講を条件とする。</li> <li>● 「チームビルディングと問題解決」の受講を条件とする。</li> <li>● 「プログラミング演習」の受講が望ましい。</li> </ul>
研修方法	演習中心（テーマ毎にショートレポートを作成する。また演習成果として設計・開発成果物の作成や、プレゼンテーション・デモンストレーションを実施する）
期間	13 回（週 1 講時）
到達目標	<ul style="list-style-type: none"> <li>● 実際の開発現場で直面する種々の問題への対処能力を実体験として習得し、次世代の情報技術者に必要な創造的、実践的なスキルを習得する。</li> </ul>
コース内容	<p>1. 組み込みソフトウェア分析・設計・実装・テスト(10 回)</p> <ul style="list-style-type: none"> <li>・ 組み込みソフトウェア開発の各工程（要求分析、設計、実装、テスト）における要素技術について講義する。また、開発支援ツールなどの紹介や机上演習も実施する。 <ul style="list-style-type: none"> <li>- 状態遷移</li> <li>- 設計パターン(省メモリ)</li> <li>- 開発環境（クロスプラットフォームでの開発など）</li> <li>- テスト手法</li> <li>- シミュレータ など</li> </ul> </li> <li>・ 組み込みソフトウェア特有の開発プロセス(ハードウェアとソフトウェアの並行開発)についても講義する。</li> <li>・ 各工程の講義受講後、複数名でチームを組み各自が主体的に開発プロジェクトを実践していく。</li> </ul> <p>2. プロジェクト運営(3 回)</p> <ul style="list-style-type: none"> <li>・ チーム内での役割（プロジェクトマネージャ/要求定義/設計/実装/品質保証）に基づく</li> </ul>
関連コース	<ul style="list-style-type: none"> <li>● ソフトウェア開発工学</li> </ul>
備考	<ul style="list-style-type: none"> <li>● 本演習は IT 企業の第一線技術者の協力の下に実施する。</li> <li>● 演習資料を適宜配布し、その都度参考文献、参考書を示す。</li> <li>● 各要素技術の詳細や実用について、講師や専門家と質疑応答する機会を設ける。</li> </ul> <p>※本演習は、ソフトウェア開発工学演習の組み込みソフトウェア版に相当する。</p>

## プロジェクトマネジメントの理論と実践

コース概要	<ul style="list-style-type: none"> <li>企業に入り即戦力としてプロジェクトに参画するにあたり、プロジェクトメンバーとして必要なプロジェクトマネジメントの基礎的な知識や体系的なマネジメント技術を学習する。</li> </ul>
受講前提	<ul style="list-style-type: none"> <li>システム開発の基本的な流れ(要件定義、設計、開発、テスト)は理解していることが望ましい。</li> </ul>
研修方法	講義および演習
期間	13回(週1講時)
到達目標	<ul style="list-style-type: none"> <li>グローバルな知識基盤に則って、モダンPMの内容と重要性を理解する。</li> <li>実務におけるプロジェクトの計画作成や実行、管理の基本を習得する。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>プロジェクトマネジメント概説 <ul style="list-style-type: none"> <li>モダンPMと経緯、PMの国際動向・国内動向、PMの知識エリアとプロセス、プロジェクトマネージャに必要なスキル、企業におけるPMコンピテンシ</li> </ul> </li> <li>プロジェクトの立ち上げとプロジェクト・スコープマネジメント <ul style="list-style-type: none"> <li>プロジェクトの選定、目標設定、スコープ計画、スコープ定義、WBS</li> <li>(演習)プロジェクトの立ち上げ <ul style="list-style-type: none"> <li>チーム編成、目標設定、スコープ決定、前提・制約の洗い出し、メンバーの役割分担</li> </ul> </li> </ul> </li> <li>プロジェクト・スケジューリングとタイムマネジメント <ul style="list-style-type: none"> <li>スケジューリング技法(PERT、CPM、クリティカル・チェーン等)、WBSの詳細化</li> <li>(演習)マスタースケジュールの作成、WBSの作成</li> </ul> </li> <li>プロジェクト・コストマネジメント <ul style="list-style-type: none"> <li>各種見積もり手法、コストの構成、コスト計画、トラッキングとコントロール</li> <li>(演習)開発規模の見積もり、ワークロードの見積もり</li> </ul> </li> <li>人的資源マネジメント <ul style="list-style-type: none"> <li>プロジェクトの組織形態、組織計画、必要スキルと要員計画、</li> <li>チームビルディングと要員育成</li> </ul> </li> <li>品質マネジメント <ul style="list-style-type: none"> <li>欠陥除去工程、問題管理と変更管理、ウォークスルーとインスペクション、テスト計画の作成</li> <li>(演習)ウォークスルー/インスペクションの実施</li> </ul> </li> <li>プロジェクト・リスクマネジメント <ul style="list-style-type: none"> <li>プロジェクト・リスクマネジメントのモデル、プロジェクト・リスクマネジメントのプロセス</li> <li>(演習)プロジェクト・リスクマネジメント <ul style="list-style-type: none"> <li>リスクの識別、対応の優先順位付け、対応策の検討</li> </ul> </li> </ul> </li> <li>アーンド・バリュー・マネジメント <ul style="list-style-type: none"> <li>EVMの概念、基本用語、計上法、効率指標、予測</li> <li>(演習)EVMと進捗マネジメント <ul style="list-style-type: none"> <li>進捗計画、実績入力、進捗報告、今後の見通し</li> </ul> </li> </ul> </li> <li>プロジェクト調達マネジメント <ul style="list-style-type: none"> <li>国際的なプロジェクト契約形態、動機付け型契約、日本でのプロジェクト調達の状況</li> </ul> </li> <li>(演習)プロジェクト計画書の作成 <ul style="list-style-type: none"> <li>プロジェクト計画書の作成</li> </ul> </li> <li>エンタープライズPMとPMが企業に与える効果</li> </ol>

	<ul style="list-style-type: none"> <li>・ PMが企業に与える影響、PMオフィス、プロジェクト・ポートフォリオ、PMスキル育成</li> </ul> <p>12. 全体のまとめ</p> <ul style="list-style-type: none"> <li>・ プロジェクトマネジメントのまとめ</li> <li>・ (演習) チーム発表</li> </ul> <p>13. クイズ</p> <ul style="list-style-type: none"> <li>・ 理解度の確認、評価</li> </ul>
関連コース	
備考	

## プロジェクトリスクマネジメントとプロセス改善

コース概要	<ul style="list-style-type: none"> <li>システム開発プロジェクトのリスク低減に必要な、プロジェクトリスクマネジメントやプロセス改善の活動について、基本的な考え方を、座学と演習を通じて学ぶ。</li> </ul>
受講前提	<ul style="list-style-type: none"> <li>プロジェクトマネジメント知識体系(例えば PMBOK)についての基本的な理解ができていること。</li> <li>システム開発に関する法規および会計に関する基礎知識があることが望ましい。</li> </ul>
研修方法	講義および演習
期間	13回(週1講時)
到達目標	<ul style="list-style-type: none"> <li>プロジェクトリスクについて理解する。</li> <li>プロジェクトリスクマネジメントの考え方を理解する。</li> <li>組織レベルのリスク低減の取り組みとしてのプロセス改善の考え方を理解する。</li> <li>PMO(Project Management Office)や SEPG(Software Engineering Group)が組織に必要なことを理解する。</li> </ul>
コース内容	<ol style="list-style-type: none"> <li>プロジェクトリスクマネジメントの概念 <ul style="list-style-type: none"> <li>リスクの定義、リスクマネジメントの基本的な考え方</li> </ul> </li> <li>プロジェクトリスクケーススタディ(1) <ul style="list-style-type: none"> <li>問題プロジェクト事例をベースに、リスクの所在を検討するグループ演習</li> </ul> </li> <li>プロジェクトリスクの種類 <ul style="list-style-type: none"> <li>様々なプロジェクトリスクとそのヘッジの考え方</li> </ul> </li> <li>見積もりとプロジェクトリスク <ul style="list-style-type: none"> <li>システム開発における見積もりの考え方、見積もりとリスクの関係</li> </ul> </li> <li>PMO活動によるプロジェクトリスクマネジメントの実際 <ul style="list-style-type: none"> <li>PMO活動の実例</li> </ul> </li> <li>EVMとリスクコントロール <ul style="list-style-type: none"> <li>EVMの概念、EVMを利用したリスクコントロール</li> </ul> </li> <li>プロジェクトリスクケーススタディ(2) <ul style="list-style-type: none"> <li>ケーススタディ(1)の事例を再検討するグループ演習</li> </ul> </li> <li>ソフトウェアプロセス <ul style="list-style-type: none"> <li>ISO/IEC12207に基づくプロセス定義の概説</li> </ul> </li> <li>プロセス改善とアセスメントのモデル <ul style="list-style-type: none"> <li>CMM、CMMI、ISO/IEC15504の概説</li> </ul> </li> <li>プロセス改善の実際 <ul style="list-style-type: none"> <li>プロセス改善活動の実例</li> </ul> </li> <li>プロセスアセスメントの実際 <ul style="list-style-type: none"> <li>実際に行われるプロセスアセスメントの概要</li> </ul> </li> <li>&amp; 13 プロセスアセスメント演習(1)(2) <ul style="list-style-type: none"> <li>アセスメントケーススタディもしくはロールプレイ</li> </ul> </li> </ol>
関連コース	<ul style="list-style-type: none"> <li>プロジェクトマネジメントの理論と実践</li> <li>法規・会計入門</li> </ul>
備考	

## 拠点大学院入学者の想定スキル

本案は、

- 北海道大学、寄付講座「実ソフトウェア開発工学講座」「オープンシステム工学講座」の経験
- 「システム開発者に要求される知識」（情報処理学会、ア kredィテーション委員会）

を基にして策定した。

カテゴリ		知識・能力
大項目	中項目	
コンピテンシ	問題発見、解決	問題を発見する心構えがある
		問題を定式化し、解決策を探る意欲がある
	探求	強い探究心がある
社会常識	一般マナー	ルール、マナーを守る必要性を理解している
	教養	書籍、新聞を読む習慣がついている
	コミュニケーション	相手の話を理解し、必要な情報を引き出すことができる
		相手が受け取れる形で情報を発信することができる
		TOEIC600点以上(英語)
	文書作成	適正な日本語で文章が書ける
情報を整理し、テーマに適した構成を作ることができる		
プレゼンテーション	テーマに沿って、プレゼンおよび議論ができる	
コンピュータリテラシ	PC操作	PCの操作ができる
		タッチタイピングができる
	インターネット	電子メールが使える
		インターネットで情報収集ができる
		パスワード、ウイルス対策ソフトを管理できる
情報倫理	個人情報保護、情報引用のルールを知っている	
OAソフト	文書作成ソフト／表計算ソフト／プレゼンテーションソフトが使える	
数学的分野	微積分	微分方程式の解法を理解している
		多変数関数の微分、積分を理解している
		線積分、マクローリン展開、フーリエ解析を理解している
	線形代数	線形写像の概念を理解している
		ベクトル、行列、逆行列、固有値の計算ができる
	離散数学	集合の記法／演算、順列、組み合わせの計算ができる
		ブール代数の演算ができる
		グラフ理論の基礎を知っている
	数理計画法	線形計画法を理解している
		非線形計画法、整数計画法、動的計画法、分岐計画法を知っている
	確率統計学	初歩的な確率計算、データからの平均／分散／期待値の算出ができる
		二項分布、ポアソン分布、正規分布を理解している

別紙1 拠点大学院入学者の想定スキル

		重回帰を理解している
	数値計算	計算誤差について理解している
		ニュートン法、シンプソン法を使える
論理学	記号理論を理解している	
コンピュータ サイエンス	情報論	情報量、エントロピー、符号理論を知っている
		計算量論を知っている
		チューリングマシンを知っている
		オートマトンを理解している
	ハードウェア	コンピュータの構成要素とその役割を理解している
		数値、文字の計算機内表現法を知っている
		論理回路を理解している
	OS	OSの役割を理解している
	コンパイラ	コンパイラの動作概念を知っている
	ソフトウェア工学	開発工程を知っている
	ソフトウェア アーキテクチャ	クライアント/サーバ、層構造モデルを知っている
		WEBサーバの役割を知っている
	データ構造と アルゴリズム	リスト、スタック、キュー、ツリー、ハッシュを理解している
		再帰を理解している
		ソート、サーチ、圧縮について複数の手法を理解している
	プログラミング	C, C++, C#, Java のいずれかでプログラミングできる
		構造化手法、オブジェクト指向を理解している
	データベース	リレーショナル演算を理解している
		正規化の意義を理解している
		SQLを理解している
	セキュリティ	ファイアウォールを知っている
		暗号化理論を知っている
	ネットワーク	TCP/IPの基本を理解している
OSIの階層モデルを知っている		