

情報学を専門とする学科対象の  
教育カリキュラム標準の策定及び提言

SE

目次

1. はじめに	1
2. J07-SE以降の状況	1
3. 旧版からの主な変更点	3
4. J17-SEの全体像	3
5. J17-SEの知識項目と科目	6
6. ACM/IEEE-CSへの報告	13
7. おわりに	13
付録1 J17-SEの知識項目	14
付録2 J17-SEの情報科学基礎科目（抜粋）	31
付録3 J17-SEのSE科目（抜粋）	35

WG 構成

主査 井上 克郎（阪大）

副主査 鷺崎 弘宜（早大）

アドバイザー 鵜林 尚靖（九大）

幹事 沢田 篤史（南山大），松下 誠（阪大）

委員

青木 利晃（北陸先端大），大久保 隆夫（情報セキュリティ大），大平 雅雄（和歌山大）

菊地 奈穂美（沖電気工業），白銀 純子（東京女子大），高田 眞吾（慶応大）

新田 直也（甲南大），野田 夏子（芝浦工大），長谷川 勇（スクウェア・エニックス）

伏田 享平（NTT DATA），道浦 康貴（JAMSS），横川 智教（岡山県立大）

## 1. はじめに

J17-SE は、情報専門学科カリキュラム標準 J17 におけるソフトウェアエンジニアリング領域のカリキュラムモデルである。これは情報処理学会ソフトウェアエンジニアリング教育委員会において、2007 年度に策定を行った J07-SE をもとにして策定したものである。

本概要では、J07-SE 策定以降のソフトウェアエンジニアリング教育を取り巻く状況とともに J17-SE カリキュラムモデルについて述べ、今後の取り組みを概説する。

## 2. J07-SE 以降の状況

改めて述べるまでもなく、企業や政府ならびに地方公共団体といった日本を支える主要な組織における情報システム（エンタープライズ系システム）や、ソフトウェアが高度に組み込まれた製品（組込み系システム）などの普及によって、ソフトウェアは今日の社会になくしてはならない存在となっている。我々の生活も企業の活動も、ソフトウェアを利用せずに行うことは不可能となってきている。しかしその一方で、ソフトウェアに起因するシステム障害や製品リコールなど、数多くの問題が起きているのも事実である。特に最近では、銀行間取引システムのような重要な社会インフラや、人命に関わる製品にも及んでいる。

これらのソフトウェアに起因した問題の原因については様々な議論があるが、質の高い人材を生み出すためのシステムが依然機能していないことが一因であろう。我が国のソフトウェアの多くは、きちんとソフトウェアエンジニアリング（以降 SE）を学んでおらず、また企業でも十分な体系的教育を受けていない技術者によって作りだされている。

ソフトウェア工学の専門技術者の育成を目的として、世界的には欧米を中心に一定数の大学でソフトウェア工学専門の学科が設置されているが、我が国には従来ソフトウェア工学を専門とする学科が存在せず、従ってカリキュラム標準 J07-SE を、全面的に大学のカリキュラム開発へと適用する機会は存在しなかった。そのような中、2009 年 4 月に南山大学において我が国で初めて、学部・学科再編の過程においてソフトウェア工学を専門に体系的に扱うソフトウェア工学科が設置された。そのソフトウェア工学科のカリキュラム開発においては、J07-SE、および、J07-SE の元となっているカリキュラム標準 SE2004 が参照モデルとして採用している。

前述の通り、J07-SE 開発終了後、SE2004 の後継にあたるカリキュラム標準 SE2014 が米国で 2015 年 2 月に制定された。SE2014 は SE2004 の文書構造をおおむね引き継ぐ一方、10 年間に起きた SE 分野の広がり並びに変化を反映するために、内容の改変が行われている。さらに、SE2014 ではソフトウェアの要求ならびにセキュリティに関する内容をより明示的

にし、内容の充実が図られているが、これらは SE に対して外部から求められている視点であると言えよう。

情報処理学会ソフトウェアエンジニアリング教育委員会においては、このような流れを受けて、2016 年より J07-SE の改定作業として J17-SE に関する調査ならびに議論を行ってきた。具体的にはまず、現在の標準である SE2014 の構成を把握し、J07-SE 策定時の標準である CCSE2004 と SE2014 で述べられている SEEK (Software Engineering Education Knowledge) 内各項目の差異について精細な調査を行い、それに基づいて J17-SE で定める知識項目に関して検討を行った。J17-SE は、前身である J07-SE の構成を基本とすることによって変更内容の明確化を行うこととし、最近の標準である SE2014 を参照した上で、カリキュラム標準を再設計することによって制定することとした。

### 3. 旧版からの主な変更点

J07-SE と J17-SE では、元となったカリキュラム標準である SE2004 と SE2014 との違いが大きいため、まず SE2004 と SE2014 の違いについて述べる。

#### 3.1. SE2004 と SE2014 の相違点

SE2004 ならび SE2014 ではソフトウェア工学教育に関する項目を SEEK として定めており、概念の上位から順に Knowledge Areas, Units, Topics という 3 階層で構成される。SE2014 では、最上位の Knowledge Areas レベルにおいて SE2004 と比べ主に以下の変更が行われている。

- (1) EVO (Software Evolution) が削除された。ただし、ここに含まれている多くの Units 以下については、他の場所で再定義されているため (PRO. evo; Evolution processes and activities など)、概念として削除されたわけではない。
- (2) MGT (Software Management) も削除された。ただし、プロジェクトの計画やソフトウェア構成管理といった、この Knowledge Area に含まれる Units のいくつかは、他の場所で再定義されている。
- (3) SEC (Security) が新設された。セキュリティは情報、システム並びにネットワークを保護するために必要な知識であり、それと直行する考え方として、ソフトウェア開発サイクル中のすべての要素においてどのように保護するかを考慮する知識でもある。
- (4) REQ (Requirements analysis and specification) が新設された。要求自体は、ユーザ、顧客、その他のステークホルダといった実社会からの需要を表現するものであり、

ステークホルダの需要を引き出して分析を加え、望まれるシステム振る舞いや品質に関する適切な記述を作成するものである。

このほか SE2004 と SE2014 では、Units, Topics 単位で見ていった場合、いくつかの項目が消滅、ないし統廃合されている点、並びに、Topics 単位での細かい修正等が行われている。

### 3.2. J07-SE と J17-SE の相違点

J17-SE は、上記 SE2004 と SE2014 の相違点を J07-SE に取り込むことによって策定することとした。そのため、J07-SE と J17-SE の相違点は SE2004 と SE2014 の相違点に由来している。具体的な違いは以下の通りである。

- (1) J17-SE の知識項目を整理するにあたり、SE2014 で新設された SEC (Security) を主に扱うことを目的として「セキュアソフトウェア開発概論」という科目を 1 つ追加し、知識項目の整理の際に用いている。
- (2) J07-SE の知識項目は、存在するものに関してはすべて SE2014 で定められた項目で置き換えられ、J17-SE の知識項目とした。従って、知識項目の ID が J07-SE と J17-SE で同一であっても、その ID が指し示す内容が異なる場合、ないし、軽微な変更が行われている場合がある。なお、原則として J07-SE で扱われている科目の内容は、連続性を担保するために変更していないが、SE2014 での変更を取り込むために科目単位での割り当てや構成を変更して整合性を担保した部分がある。
- (3) J17-SE の各科目間にある依存関係を整理し、極力科目間の依存関係を持たせないようにした。
- (4) SAS 科目の内容を更新し、最新のトピックを許容するようにした。

## 4. J17-SE の全体像

### 4.1. ミニマムセットのカリキュラムモデル

J17-SE は、大学などの高等教育機関の情報専門学科における SE 教育のためのカリキュラムモデルである。まず、カリキュラムそのものではない点に注意が必要である。本カリキュラムモデルは、カリキュラムを開発するために最低限必要な知識項目、それらを教えるコア科目、演習やインターンシップの仕様、学科やコースの特色を出すための SAS 科目 (System and Application Specialties: システム応用・特化型科目) 科目の例から構成

される。学科やコースにおいてソフトウェアエンジニアリングのカリキュラムを開発するためには、本カリキュラムモデルを基にして、知識項目の把握、コア科目の理解、演習やインターンシップの実装、SAS 科目の選択や開発などの策定などを行うことが必要となる。

次に、J17-SE は SE の学科やコースにおけるミニマムセットである点に注意が必要である。本カリキュラムモデルは、日本の大学が置かれている競争的環境を踏まえ、学科やコースの特色を出す余裕を持たせるために、最低限必要な知識項目に絞ってコア科目を列挙している。そのため、学科やコースの特色を反映した科目を追加しなくてはならない。例として、エンタープライズ系システム向け SE カリキュラムや組込みシステム向け SE カリキュラム、Web システム向け SE カリキュラムなどの適用ドメイン的な特色を反映した学科やコース、要求開発やアーキテクチャ設計、実装、検証と妥当性確認、マネジメントなど技術的な特色を反映した学科やコースなどが考えられる。J17-SE では、カリキュラム開発者のために SAS 科目として、学科やコースの特色を反映させるための追加科目を例示している。また特色を出しやすいように、演習などについては内容の詳細に踏み込まず、仕様を提示するに留めている。

また真に実践的で充実した教育を行うためには、学部教育に許された時間数では少ないと言わざるを得ない。大学院との連携教育が可能な学科やコースは、ミニマムセットである J17-SE を拡張することで大学院修士課程も含めた 6 年間のカリキュラムモデルを構築するとよい。

#### 4.2. コンセプト

SE は、その名の通り「エンジニアリング」である。エンジニアリング教育には、主に 2 つの特性が求められる。一つは“実践的”であり、もう一つは“骨太”である。従来から大学で行われている情報系の教育は、その出自や理念などにより、理論的なものが多いと言われている。しかし産業界からは、開発現場であり役に立たないという批判も多い。そこで J17-SE は、産業界のニーズに応えるために従来の基礎一辺倒のカリキュラムから、実践的内容にも踏み込んだカリキュラムモデルを指向することとした。まず、プログラミング言語の習得に留まらず、開発の全体像である要求開発から検証と妥当性確認、保守、プロセスやマネジメントといった開発ライフサイクルを網羅するカリキュラムモデルとなっている。次に、理論の理解に留まらず、開発の目的である品質・生産性・コストといった要因を重視するカリキュラムモデルとなっている。また、個人に閉じた作業に留まらず、開発に必要なマネジメントやコミュニケーション、チームダイナミクスなどを学び実践するカリキュラムモデルとなっている。さらに、開発者のモチベーションや、不具合につながる開発者のヒューマンエラーといった心理的側面も取り扱っている。

一方で J17-SE は、SE の技術進化の速さを鑑みるに近視眼的な技術訓練では不十分であると考え、骨太なカリキュラムモデルを指向した。特に開発現場寄りの産業界からは、すぐに現場で使える技術を求められることもある。しかし、SE において重要なのは技術の使い方よりも“ものの考え方”そのものである。そこで、モデル化を習得することで「捉える力」や「考える力」、「表現する力」などを、検証と妥当性確認やプロセス改善を習得することで「問題発見能力」や「問題解決能力」などを、プロセスやマネジメントを習得することで「段取り力」や「調整力」などを涵養することを指向した。またソフトウェアに留まらない一般的な工学原則も学ぶこととした。これによって、技術を単に理解するだけでなく、臨機応変に応用でき中長期的に付加価値を生み続けられる技術者の育成が期待できる。

J17-SE の特徴は、こうした内容を PBL (Project Based Learning) のみに担わせるべきではないと意図している点にある。PBL はその性格上、基本概念や技術を習得する際に、PBL で採用した開発やマネジメントのスタイルに基づく一面的な理解になりかねない。そこで J17-SE では、講義においてさまざまな基本概念や技術を提示することで、PBL が採用するスタイル以外のさまざまな開発やマネジメントのスタイルの特徴を多面的に深く理解することを指向した。

#### 4.3. カリキュラムモデルの構造

J17-SE は、講義科目と実習科目、SAS 科目から構成される。講義科目はさらに、情報科学基礎科目と SE 科目に分けられる。情報科学基礎科目とは、論理と計算理論、離散数学、オペレーティングシステム基礎・データベース基礎といった SE の基礎となる科目である。SE 科目とは、ソフトウェア構築やソフトウェア設計、検証と妥当性確認、開発マネジメントといった SE 技術を扱う科目である。

実習科目は、プログラミング基礎実習、プログラミング応用実習、ソフトウェア開発実習から構成される。プログラミング基礎実習では、概略設計内容に対する個人でのプログラム開発と単体・統合テストを行う。プログラミング応用実習では、要求仕様に対する個人ないしグループでのアーキテクチャ設計、ソフトウェア設計、プログラム開発、単体・統合・システムテストを行う。ソフトウェア開発実習では、システムへの要望に対するグループでのシステム開発とプロジェクトマネジメントを行う。エンタープライズ系や組込み系など学科やコースの特色に合わせて、かなり具体的なシステムを想定した実習となる。

SAS (System and Application Specialties) 科目は、学科やコースの特色を反映させるための追加科目である。J17-SE では、システムのドメイン、システムの特長、システムの構造、開発技術による SAS 科目を例示している。ただし科目名の提示に留め、詳細は学科

やコースで定める必要がある。

## 5. J17-SE の知識項目と科目

### 5.1. 知識項目

カリキュラムモデルにまず必要となるのは、教えるべき知識項目の整理である。SE 分野には SWEBOK (Software Engineering Body of Knowledge) などいくつかの知識項目が定義されているが、J17-SE では、SE2014 を参照し科目として取り扱いやすい単位として再体系化を行っている。表 1 に知識項目の第 2 カテゴリまでを示す。付録 1 に J17-SE の知識項目を示す。知識項目の ID は SE2014 と一致させている。

表 1 J17-SE の知識項目 (第 2 カテゴリまで)

コンピュータとソフトウェアの基礎		
CMP		コンピュータ基礎
	CMP.cf	コンピュータ科学基礎
DES		ソフトウェア設計
	DES.str	設計のパラダイム
MAA		ソフトウェアのモデリングと分析
	MAA.tm	モデルの種類
確率統計		
FND		数理基礎・工学基礎
	FND.ef	ソフトウェアのための工学基礎
	FND.mf	数理基礎
離散数学		
FND		数理基礎・工学基礎
	FND.mf	数理基礎
プログラミング基礎		
CMP		コンピュータ基礎
	CMP.cf	コンピュータ科学基礎
	CMP.ct	構築技術
	CMP.tl	構築のためのツール
論理と計算理論		
FND		数理基礎・工学基礎

	FND.mf	数理基礎
オペレーティングシステム基礎・データベース基礎		
CMP		コンピュータ基礎
	CMP.cf	コンピュータ科学基礎
ネットワーク基礎		
CMP		コンピュータ基礎
	CMP.cf	コンピュータ科学基礎
SEC		セキュリティ
	SEC.net	コンピュータとネットワークのセキュリティ
	SEC.sfd	セキュリティ基礎
工学基礎		
CMP		コンピュータ基礎
	CMP.cf	コンピュータ科学基礎
FND		数理基礎・工学基礎
	FND.ef	ソフトウェアのための工学基礎
MAA		ソフトウェアのモデリングと分析
	MAA.af	モデルの分析の基礎
PRF		プロフェッショナルプラクティス
	PRF.pr	プロフェッショナルリズム
REQ		要求の分析と仕様
ソフトウェア構築		
CMP		コンピュータ基礎
	CMP.tl	構築のためのツール
DES		ソフトウェア設計
	DES.con	設計に用いられる概念
MAA		ソフトウェアのモデリングと分析
	MAA.af	モデルの分析の基礎
PRO		ソフトウェア開発プロセス
	PRO.con	プロセスの基礎
VAV		検証と妥当性確認
	VAV.rev	レビューと静的解析
モデル化と要求開発		

MAA		ソフトウェアのモデリングと分析
	MAA.md	モデリングの基礎
	MAA.tm	モデルの種類
PRO		ソフトウェア開発プロセス
	PRO.pp	プロジェクトの計画
REQ		要求の分析と仕様
	REQ.er	要求の獲得
	REQ.rfd	要求分析の基礎
	REQ.rsd	要求の仕様化と文書化
	REQ.rv	要求の評価
VAV		検証と妥当性確認
	VAV.tst	テスト
ソフトウェアアーキテクチャ		
DES		ソフトウェア設計
	DES.ar	アーキテクチャ設計
	DES.con	設計に用いられる概念
	DES.ev	設計の評価
	DES.str	設計のパラダイム
ソフトウェア設計		
DES		ソフトウェア設計
	DES.con	設計に用いられる概念
	DES.dd	詳細設計
	DES.str	設計のパラダイム
検証と妥当性確認		
VAV		検証と妥当性確認
	VAV.fnd	V&V の用語と基礎
	VAV.par	不具合の分析と報告
	VAV.rev	レビューと静的解析
	VAV.tst	テスト
形式手法		
CMP		コンピュータ基礎
	CMP.ct	構築技術

DES		ソフトウェア設計
	DES.dd	詳細設計
	DES.ev	設計の評価
MAA		ソフトウェアのモデリングと分析
	MAA.af	モデルの分析の基礎
	MAA.md	モデリングの基礎
	MAA.tm	モデルの種類
REQ		要求の分析と仕様
	REQ.rv	要求の評価
VAV		検証と妥当性確認
	VAV.rev	レビューと静的解析
ソフトウェアプロセスと品質		
FND		数理基礎・工学基礎
	FND.ec	ソフトウェアのためのエンジニアリングエコノミクス
PRF		プロフェッショナルプラクティス
	PRF.pr	プロフェッショナリズム
PRO		ソフトウェア開発プロセス
	PRO.cm	ソフトウェア構成管理
	PRO.con	プロセスの基礎
	PRO.evo	進化のプロセスと活動
	PRO.imp	プロセスの実装
	PRO.pp	プロジェクトの計画
QUA		ソフトウェア品質
	QUA.cc	ソフトウェア品質の概念と文化
	QUA.pca	プロセスの保証
	QUA.pda	プロダクトの保証
ヒューマンファクター		
CMP		コンピュータ基礎
	CMP.cf	コンピュータ科学基礎
DES		ソフトウェア設計
	DES.hci	ヒューマン・コンピュータ・インタフェース (HCI) 設計
開発マネジメント		

DES		ソフトウェア設計
	DES.ar	アーキテクチャ設計
PRF		プロフェッショナルプラクティス
	PRF.com	(SE に特化した) コミュニケーションスキル
	PRF.pr	プロフェッショナルリズム
	PRF.psy	グループダイナミクスと心理学
PRO		ソフトウェア開発プロセス
	PRO.con	プロセスの基礎
	PRO.imp	プロセスの実装
	PRO.pp	プロジェクトの計画
QUA		ソフトウェア品質
	QUA.pca	プロセスの保証
セキュアソフトウェア開発概論		
SEC		セキュリティ
	SEC.dev	安全なソフトウェアの開発

## 5.2. 情報科学基礎科目

J17-SE では前述した知識項目を基に、8つの情報科学基礎科目を提示した。表2に情報科学基礎科目の一覧を示す。また付録2に情報科学基礎科目シラバスの一部を示す（シラバス全体は資料に掲載している）。なお確率統計については教養系で実施する講義を想定しているが、各種SE技術の背景・基礎という観点で習得の必要な内容をカリキュラムモデルに含めて策定した。プログラミング基礎については、学科やコースの特色に応じたプログラミング言語を対象として科目を開発すべきである。本カリキュラムモデルでは、例としてC言語を用いた科目を提示している。

表2 J17-SE の情報科学基礎科目の一覧

コンピュータとソフトウェアの基礎
確率統計
離散数学
プログラミング基礎
論理と計算理論
オペレーティングシステム基礎・データベース基礎
ネットワーク基礎

### 5.3. SE 科目

J17-SE では前述した知識項目を基に、10 の SE 科目を提示した。表 3 に SE 科目の一覧を示す（シラバス全体は資料に掲載している）。また付録 3 に SE 科目シラバスの一部を示す。

表 3 J17-SE の SE 科目の一覧

ソフトウェア構築
モデル化と要求開発
ソフトウェアアーキテクチャ
ソフトウェア設計
検証と妥当性確認
形式手法
ソフトウェアプロセスと品質
ヒューマンファクター
開発マネジメント
セキュアソフトウェア開発概論

### 5.4. 実習科目

J17-SE では、4 つの実習科目を提示した。表 4 に実習科目の一覧を示す（シラバス全体は資料に掲載している）。ソフトウェア開発実習については、例としてエンタープライズ系と組込み系の 2 種を示す。

表 4 J17-SE の実習科目

プログラミング基礎実習
プログラミング応用実習
エンタープライズ開発実習
組込みソフトウェア開発実習

### 5.5. SAS 科目

J17-SE では SAS 科目の例を提示した。SAS 科目の例は、システムのドメイン、システム  
の特性、システムの構造、開発技術などに分類している。ただし J17-SE では各 SAS 科目の

内容は定めていない。表 5 に SAS 科目の例の一覧を示す。

表 5 J17-SE の SAS 科目の例の一覧

システムのドメインによる SAS 科目
Web システム (ネットワーク中心・Web システム)
企業情報システム・ERP (情報システムとデータ処理)
金融・電子商取引システム
物流・小売システム
通信・ネットワークシステム
エンターテインメントシステム (ゲーム・CG・音声)
ユビキタスシステム (モバイルシステム)
航空・車システム
工業プロセス制御システム
生命医学システム (バイオメディカルシステム)
科学技術計算システム
XR 利活用システム (AR・VR・MR)
システムの特性による SAS 科目
高信頼性・高可用性システム/組込み(ディペンダブルシステム)
高信頼性・高可用性システム/エンタープライズ(ディペンダブルシステム)
高セキュリティ情報システム
高安全性組込みシステム
システムの構造による SAS 科目
ハードウェア制御・リアルタイムシステム (組込み・リアルタイムシステム)
トランザクションシステム
制御モデル開発
経営情報システム
レガシーシステムとプロダクトライン
OSS によるシステム開発
エージェント・人工知能システム
IoT・サイバーフィジカルシステム
クラウド・モバイルシステム
開発技術による SAS 科目

ソフトウェア構築・応用
モデル化と要求開発・応用
ソフトウェアアーキテクチャ・応用
ソフトウェア設計・応用
検証と妥当性確認・応用
形式手法・応用
ソフトウェアプロセスと品質・応用
ヒューマンファクター・応用
開発マネジメント・応用
アジャイル・DevOps・応用

## 6. ACM/IEEE-CS への報告

4章および5章で述べた本カリキュラム標準 J17-SE の策定にあたっては、ACM、IEEE-CS によるカリキュラム標準である SE2014、前身となったカリキュラム標準である J07-SE、また日本国内の情報専門科目の状況や、ソフトウェア工学の専門技術の現状を踏まえて行った。

J17-SE 策定にあたっての検討内容は、英文でまとめたうえで、ACM、IEEE-CS など参照したカリキュラム標準を策定した団体等にも報告する。具体的には、SE2004 と SE2014 の違い、ならびに、我々が策定した J07-SE と J17-SE の違いなど、主に3章の内容について報告を行う。

## 7. おわりに

本概要では、ソフトウェアエンジニアリングを取り巻く状況、J17-SE カリキュラムモデルの全体像、知識項目および科目の概説を行った。今後の取り組みとしては、J17-SE を基にしたカリキュラム例の策定、複数の大学や大学院の SE カリキュラムの比較が考えられる。多面的な開発やマネジメントのスタイルを理解できるような PBL の例を提示することも必要だろう。また産業界や学会からの意見を募り、さらに実践的かつ骨太なカリキュラムを目指して改善していくとともに、大学院修士課程を含む6年でのカリキュラムモデルの検討も視野に入れたい。

付録1 J17-SE の知識項目

コンピュータとソフトウェアの基礎			
CMP			コンピュータ基礎
	CMP.cf		コンピュータ科学基礎
		CMP.cf.1	プログラミング基礎（制御とデータ、型付け、再帰）
		CMP.cf.2	アルゴリズムとデータ構造、データ表現（静的・動的）、複雑性
		CMP.cf.5	コンピュータの構造
		CMP.cf.8	プログラミング言語の基礎
		CMP.cf.9	オペレーティングシステムの基礎
		CMP.cf.11	ネットワークプロトコル
DES			ソフトウェア設計
	DES.str		設計のパラダイム
		DES.str.1	機能指向による設計
		DES.str.2	オブジェクト指向による設計
MAA			ソフトウェアのモデリングと分析
	MAA.tm		モデルの種類
		MAA.tm.5	エンタープライズシステムのモデリング（ビジネスプロセス、組織、ゴール、ワークフローなど）
		MAA.tm.6	組込みシステムのモデリング（リアルタイムスケジューリング分析、外部インタフェース分析など）
確率統計			
FND			数理基礎・工学基礎
	FND.ef		ソフトウェアのための工学基礎
		FND.ef.2	統計解析（検定と推定、回帰分析、相関など）
	FND.mf		数理基礎
		FND.mf.6	離散確率
		FND.mf.9	数値誤差と精度

離散数学			
FND			数理基礎・工学基礎
	FND.mf		数理基礎
		FND.mf.4	数え上げ基礎
		FND.mf.5	グラフとツリー
		FND.mf.10	数論
プログラミング基礎			
CMP			コンピュータ基礎
	CMP.cf		コンピュータ科学基礎
		CMP.cf.1	プログラミング基礎（制御とデータ、型付け、再帰）
		CMP.cf.2	アルゴリズムとデータ構造、データ表現（静的・動的）、複雑性
		CMP.cf.4	抽象化（カプセル化や階層化など）
	CMP.ct		構築技術
		CMP.ct.2	コードの再利用とライブラリ
		CMP.ct.4	パラメータ化と汎化
		CMP.ct.5	アサーション、契約による設計(DbC)、防御的プログラミング
		CMP.ct.6	エラーハンドリング、例外処理、フォールトトレラント
	CMP.tl		構築のためのツール
		CMP.tl.1	開発環境
		CMP.tl.2	GUI 構築ツール
		CMP.tl.3	単体テストツール
		CMP.tl.4	プロファイリング・パフォーマンス分析のツール
論理と計算理論			
FND			数理基礎・工学基礎
	FND.mf		数理基礎
		FND.mf.1	関数、関係、集合
		FND.mf.2	論理学基礎（命題、述語）

		FND.mf.3	証明技法
		FND.mf.7	有限状態機械と正規表現
		FND.mf.8	文法
オペレーティングシステム基礎・データベース基礎			
CMP			コンピュータ基礎
	CMP.cf		コンピュータ科学基礎
		CMP.cf.9	オペレーティングシステムの基礎
		CMP.cf.10	データベースの基礎
ネットワーク基礎			
CMP			コンピュータ基礎
	CMP.cf		コンピュータ科学基礎
		CMP.cf.11	ネットワークプロトコル
SEC			セキュリティ
	SEC.net		コンピュータとネットワークのセキュリティ
		SEC.net.1	ネットワークセキュリティの脅威と攻撃
		SEC.net.2	ネットワークセキュリティのための暗号理論の使用
		SEC.net.3	保護と防衛のメカニズムとツール
	SEC.sfd		セキュリティ基礎
		SEC.sfd.1	情報の保証の概念（秘匿、完全性、可用性）
		SEC.sfd.2	脅威の起源（自然、意図的、事故など）
		SEC.sfd.3	暗号化、デジタル署名、メッセージの認証、ハッシュ関数
		SEC.sfd.4	暗号プロトコル（アプリケーション、長所、短所）
		SEC.sfd.5	非技術的なセキュリティ問題（ソーシャルエンジニアリングなど）
工学基礎			
CMP			コンピュータ基礎
	CMP.cf		コンピュータ科学基礎
		CMP.cf.3	問題解決技法
FND			数理基礎・工学基礎

	FND.ef		ソフトウェアのための工学基礎
		FND.ef.1	統計的技法と実験的技法（CPU やメモリの利用に対する測定法）
		FND.ef.2	統計解析（検定と推定、回帰分析、相関など）
		FND.ef.3	測定とメトリクス
		FND.ef.4	システム特性（セキュリティ、安全性、パフォーマンス、スケーラビリティ、機能競合など）
		FND.ef.5	工学的設計の基本概念（問題の定式化、別解、フィージビリティスタディなど）
		FND.ef.6	測定の理論（意味のある測定の基準など）
MAA			ソフトウェアのモデリングと分析
	MAA.af		モデルの分析の基礎
		MAA.af.1	まとまりの分析（完全性、一貫性、ロバスト性など）
		MAA.af.2	正当性の分析（静的解析、シミュレーション、モデルチェックなど）
		MAA.af.3	依存性の分析（フォールトモード分析、フォールトツリー解析など）
PRF			プロフェッショナルプラクティス
	PRF.pr		プロフェッショナリズム
		PRF.pr.2	倫理綱領とプロフェッショナルとしての行動
REQ			要求の分析と仕様
	REQ.rfd		要求分析の基礎
		REQ.rfd.5	品質特性（非機能特性）の分析（安全性、セキュリティ、ユーザビリティ、性能など）
		REQ.rfd.8	トレーサビリティ
		REQ.rfd.9	優先順位付け、トレードオフ分析、要求のためのリスク分析および影響解析
ソフトウェア構築			
CMP			コンピュータ基礎
	CMP.ct		構築技術
		CMP.ct.1	API の設計と利用

		CMP.ct.2	コードの再利用とライブラリ
		CMP.ct.3	オブジェクト指向パラダイムにおける実行時のトピック（ポリモルフィズム、ダイナミックバイインディングなど）
		CMP.ct.4	パラメータ化と汎化
		CMP.ct.5	アサーション、契約による設計(DbC)、防御的プログラミング
		CMP.ct.6	エラーハンドリング、例外処理、フォールトトレラント
		CMP.ct.7	状態ベースおよびテーブル駆動の構築技法
		CMP.ct.8	実行時コンフィグレーションと国際化
		CMP.ct.9	文法ベースの入力処理（パース処理）
		CMP.ct.10	並列処理の基本要素（セマフォ、モニターなど）
		CMP.ct.11	分散ソフトウェアのための構築技術（クラウド、モバイルコンピューティングなど）
		CMP.ct.12	ハードウェアとソフトウェアシステムの構築
		CMP.ct.13	ホットスポット分析とパフォーマンスチューニング
	CMP.tl		構築のためのツール
		CMP.tl.1	開発環境
		CMP.tl.2	GUI 構築ツール
		CMP.tl.3	単体テストツール
		CMP.tl.4	プロファイリング・パフォーマンス分析のツール
DES			ソフトウェア設計
	DES.con		設計に用いられる概念
MAA			ソフトウェアのモデリングと分析
	MAA.af		モデルの分析の基礎
		MAA.af.2	正当性の分析（静的解析、シミュレーション、モデルチェックなど）
PRO			ソフトウェア開発プロセス

	PRO.con		プロセスの基礎
VAV			検証と妥当性確認
	VAV.rev		レビューと静的解析
		VAV.rev.3	静的解析（欠陥検出、形式仕様に対する検査など）
モデル化と要求開発			
MAA			ソフトウェアのモデリングと分析
	MAA.md		モデリングの基礎
		MAA.md.1	モデリングの原則（分解、抽象化、汎化、投影 / ビュー、形式的アプローチの利用など）
	MAA.tm		モデルの種類
		MAA.tm.1	情報やデータのモデリング（ERD、クラス図など）
		MAA.tm.2	振る舞いのモデリング（状態遷移図、ユースケース分析、インタラクション図、FMEA、FTA など）
		MAA.tm.3	アーキテクチャのモデリング（アーキテクチャパターン、コンポーネント図など）
		MAA.tm.4	ドメインのモデリング（ドメインエンジニアリング・アプローチなど）
		MAA.tm.5	エンタープライズシステムのモデリング（ビジネスプロセス、組織、ゴール、ワークフローなど）
		MAA.tm.6	組み込みシステムのモデリング（リアルタイムスケジュール分析、インタフェースプロトコルなど）
PRO			ソフトウェア開発プロセス
	PRO.pp		プロジェクトの計画
		PRO.pp.1	要求のマネジメント（プロダクトのバックログ、優先順位、依存関係、変更など）
REQ			要求の分析と仕様
	REQ.er		要求の獲得

		REQ.er.1	要求を獲得する対象（ステークホルダ、ドメインエキスパート、操作環境や組織環境など）
		REQ.er.2	要求獲得の技法（インタビュー、アンケート/調査、プロトタイピング、ユースケース、観察、参加型技法など）
	REQ.rfd		要求分析の基礎
		REQ.rfd.1	要求という概念の定義（プロダクト、プロジェクト、制約、システムの境界、システムの外部、システムの内部など）
		REQ.rfd.2	要求分析のプロセス
		REQ.rfd.3	要求のレベル/階層（ニーズ、ゴール、ユーザ要求、システム要求、ソフトウェア要求など）
		REQ.rfd.4	要求が備えるべき特性（検証可能性、非曖昧性、一貫性、正当性、トレーサビリティ、優先度など）
		REQ.rfd.5	品質特性（非機能特性）の分析（安全性、セキュリティ、ユーザビリティ、性能など）
		REQ.rfd.6	システムエンジニアリングにおけるソフトウェア要求
		REQ.rfd.7	要求の進化
		REQ.rfd.8	トレーサビリティ
		REQ.rfd.9	優先順位付け、トレードオフ分析、要求のためのリスク分析および影響解析
		REQ.rfd.10	要求のマネジメント（一貫性管理、リリースの計画、再利用など）
		REQ.rfd.11	要求とアーキテクチャの競合
	REQ.rsd		要求の仕様化と文書化
		REQ.rsd.1	要求の文書化の基礎（種類、読み手、構造、品質、属性、標準など）
		REQ.rsd.2	ソフトウェア要求の仕様化技術（計画による要求の文書化、ディビジョンテーブル、ユーザストーリー、振る舞いの仕様）
	REQ.rv		要求の評価

		REQ.rv.1	レビューとインスペクション
		REQ.rv.2	要求の評価のためのプロトタイピング（累積型プロトタイピング）
		REQ.rv.3	受け入れテストの設計
		REQ.rv.4	品質特性の評価
		REQ.rv.5	要求競合のための分析（機能競合など）
		REQ.rv.6	形式手法による要求の分析
VAV			検証と妥当性確認
	VAV.tst		テスト
		VAV.tst.8	システムテストと受け入れテスト
ソフトウェアアーキテクチャ			
DES			ソフトウェア設計
	DES.ar		アーキテクチャ設計
		DES.ar.1	アーキテクチャスタイル（パイプアンドフィルタ、レイヤード、トランザクション中心、ピアツーピア、publish/subscribe、イベント駆動、クライアントサーバなど）
		DES.ar.2	アーキテクチャで考慮すべき様々な特性間のトレードオフ
		DES.ar.3	ソフトウェアアーキテクチャで考慮すべきハードウェア
		DES.ar.4	アーキテクチャにおける要求のトレーサビリティ
		DES.ar.5	サービス中心のアーキテクチャ
		DES.ar.6	ネットワーク、モバイル、組込みシステムのアーキテクチャ
	DES.con		設計に用いられる概念
		DES.con.1	設計という概念の定義
		DES.con.2	基本的な設計の考慮事項（データの永続性、ストレージマネジメント、例外など）
		DES.con.3	複数のソフトウェア開発ライフサイクルにおける設計の関係
		DES.con.4	設計の原則（情報隠蔽、凝集度と結合度）

		DES.con.5	設計と要求との競合
		DES.con.6	品質特性の設計（信頼性、ユーザビリティ、性能、テスト容易性、フォールトトレラント性）
		DES.con.7	設計におけるトレードオフ
	DES.ev		設計の評価
		DES.ev.1	設計上の特性（結合度、凝集度、情報隠蔽、関心事の分離など）
		DES.ev.2	設計のメトリックス
		DES.ev.3	形式手法による設計の分析
	DES.str		設計のパラダイム
		DES.str.1	機能指向による設計
		DES.str.2	オブジェクト指向による設計
		DES.str.3	データ構造を中心とした設計
		DES.str.4	アスペクト指向による設計
ソフトウェア設計			
DES			ソフトウェア設計
	DES.con		設計に用いられる概念
		DES.con.1	設計という概念の定義
		DES.con.2	基本的な設計の考慮事項（データの永続性、ストレージマネジメント、例外など）
		DES.con.3	複数のソフトウェア開発ライフサイクルにおける設計の関係
		DES.con.4	設計の原則（情報隠蔽、凝集度と結合度）
	DES.dd		詳細設計
		DES.dd.1	デザインパターン
		DES.dd.2	データベースデザイン
		DES.dd.3	ネットワークとモバイルシステムのデザイン
		DES.dd.4	設計の記法（クラス図とオブジェクト図、UML、状態遷移図、形式仕様など）
	DES.str		設計のパラダイム
		DES.str.1	機能指向による設計
		DES.str.2	オブジェクト指向による設計

		DES.str.3	データ構造を中心とした設計
検証と妥当性確認			
VAV			検証と妥当性確認
	VAV.fnd		V&V の用語と基礎
		VAV.fnd.1	V&V の目的と制約
		VAV.fnd.2	V&V の計画
		VAV.fnd.3	V&V の戦略のドキュメント化（テストなど）
		VAV.fnd.4	メトリクスと測定（信頼性、ユーザビリティ、性能など）
		VAV.fnd.5	V&V に関連する活動
	VAV.par		不具合の分析と報告
		VAV.par.1	不具合報告書の分析
		VAV.par.2	デバッグ/不具合切り分けの技法
		VAV.par.3	欠陥分析
		VAV.par.4	不具合の追跡
	VAV.rev		レビューと静的解析
		VAV.rev.1	個人レビュー（デザイン、コードなど）
		VAV.rev.2	ピアレビュー（インスペクション、ウォークスルーなど）
		VAV.rev.3	静的解析（欠陥検出、形式仕様に対する検査など）
	VAV.tst		テスト
		VAV.tst.1	単体テストとテスト駆動開発
		VAV.tst.2	例外のハンドリング（境界条件と範囲条件のテスト）
		VAV.tst.3	カバレッジ分析と構造ベースのテスト
		VAV.tst.4	ブラックボックスの技法
		VAV.tst.5	結合テスト
		VAV.tst.6	ユースケースや顧客シナリオによるテストケースの設計
		VAV.tst.7	操作プロファイルによるテスト
		VAV.tst.8	システムテストと受け入れテスト

		VAV.tst.9	品質特性に関連するテスト（ユーザビリティ、セキュリティ、互換性、アクセシビリティなど）
		VAV.tst.10	回帰テスト
		VAV.tst.11	テストツールと自動化
		VAV.tst.12	ユーザインタフェースのテスト
		VAV.tst.13	ユーザビリティテスト
		VAV.tst.14	性能テスト
形式手法			
CMP			コンピュータ基礎
	CMP.ct		構築技術
		CMP.ct.5	アサーション、契約による設計(DbC)、防御的プログラミング
DES			ソフトウェア設計
	DES.dd		詳細設計
		DES.dd.4	設計の記法（クラス図とオブジェクト図、UML、状態遷移図、形式仕様など）
	DES.ev		設計の評価
		DES.ev.3	形式手法による設計の分析
MAA			ソフトウェアのモデリングと分析
	MAA.af		モデルの分析の基礎
		MAA.af.1	まとまりの分析（完全性、一貫性、ロバスト性など）
		MAA.af.2	正当性の分析（静的解析、シミュレーション、モデルチェッキングなど）
		MAA.af.4	形式手法による分析（定理証明など）
	MAA.md		モデリングの基礎
		MAA.md.1	モデリングの原則（分解、抽象化、汎化、投影 / ビュー、形式的アプローチの利用など）
		MAA.md.2	事前条件、事後条件、不変表明
		MAA.md.3	数理モデルと形式記述の紹介
	MAA.tm		モデルの種類

		MAA.tm.1	情報やデータのモデリング（ERD、クラス図など）
		MAA.tm.2	振る舞いのモデリング（構造化分析、状態遷移図、ユースケース分析、インタラクション図、FMEA、FTA など）
REQ			要求の分析と仕様
	REQ.rv		要求の評価
		REQ.rv.6	形式手法による要求の分析
VAV			検証と妥当性確認
	VAV.rev		レビューと静的解析
		VAV.rev.3	静的解析（欠陥検出、形式仕様に対する検査など）
ソフトウェアプロセスと品質			
FND			数理基礎・工学基礎
	FND.ec		ソフトウェアのためのエンジニアリングエコノミクス
		FND.ec.1	ソフトウェアライフサイクルを通じた価値の考慮
		FND.ec.2	費用対効果の評価（利益実現、トレードオフ分析、コスト分析、ROI 分析など）
PRF			プロフェッショナルプラクティス
	PRF.pr		プロフェッショナリズム
		PRF.pr.7	雇用形態・雇用契約
PRO			ソフトウェア開発プロセス
	PRO.cm		ソフトウェア構成管理
		PRO.cm.1	リビジョン管理
		PRO.cm.2	リリース管理
		PRO.cm.3	支援ツール
		PRO.cm.4	ビルド
		PRO.cm.5	構成管理のプロセス
		PRO.cm.6	保守に関する考慮事項
		PRO.cm.7	分散環境とバックアップ

	PRO.con		プロセスの基礎
		PRO.con.1	プロセスの概念と用語
		PRO.con.2	プロセスのインフラストラクチャ（開発者、ツール、教育など）
		PRO.con.3	プロセスのモデリングと仕様化
		PRO.con.4	プロセスの測定と分析
		PRO.con.5	プロセスの改善（個人、チーム、組織の改善）
		PRO.con.6	品質の分析とコントロール（プロセスと経験を改善するための、欠陥予防、レビュー、品質特性、致命的欠陥の根本原因分析など）
		PRO.con.7	システムエンジニアリングのライフサイクルモデル
	PRO.evo		進化のプロセスと活動
		PRO.evo.1	進化や保守の基礎
		PRO.evo.2	レガシーシステムの扱い
		PRO.evo.3	リファクタリング
	PRO.imp		プロセスの実装
		PRO.imp.1	プロセス定義のレベル（組織、プロジェクト、チーム、個人など）
		PRO.imp.2	ライフサイクルモデルの特性（計画ベース、インクリメンタル、繰り返し、アジャイルなど）
		PRO.imp.3	個人によるソフトウェアプロセス（モデル、定義、測定、分析、改善）
		PRO.imp.4	チームによるソフトウェアプロセス（モデル、定義、組織、測定、分析、改善）
		PRO.imp.5	システムエンジニアリングにおけるプロセスの実装
		PRO.imp.6	プロセスのテーラリング
		PRO.imp.7	プロセスの外的要因の影響（契約と法的要求、標準、調達など）
	PRO.pp		プロジェクトの計画

		PRO.pp.6	プロジェクト追跡のためのメトリクスと技術 (データの取得、速度、バーンダウンチャート、欠陥追跡、技術的負債のマネジメント)
QUA			ソフトウェア品質
	QUA.cc		ソフトウェア品質の概念と文化
		QUA.cc.1	品質という概念の定義
		QUA.cc.2	品質に対する社会の関わり方
		QUA.cc.3	低い品質によって発生するコストと影響
		QUA.cc.4	品質モデルのコスト
		QUA.cc.5	ソフトウェアの品質特性 (信頼性、ユーザビリティ、安全性など)
		QUA.cc.6	人、プロセス、技法、ツール、技術の役割
	QUA.pca		プロセスの保証
		QUA.pca.1	プロセス保証の起源
		QUA.pca.2	品質計画
		QUA.pca.3	プロセス保証技術
	QUA.pda		プロダクトの保証
		QUA.pda.1	プロダクト保証の起源
		QUA.pda.2	保証と V&V の違い
		QUA.pda.3	プロダクト品質モデル
		QUA.pda.4	根本原因分析と欠陥予防
		QUA.pda.5	プロダクト品質のメトリクスと測定
		QUA.pda.6	品質特性のアセスメント (ユーザビリティ、信頼性、アベイラビリティなど)
ヒューマンファクター			
CMP			コンピュータ基礎
	CMP.cf		コンピュータ科学基礎
		CMP.cf.6	ヒューマンファクターの基礎 (ユーザ側: 入力、エラーメッセージ、障害対応)
		CMP.cf.7	ヒューマンファクターの基礎 (開発者側: コメント、構造、可読性)
DES			ソフトウェア設計

	DES.hci		ヒューマン・コンピュータ・インタフェース (HCI) 設計
		DES.hci.1	一般的な HCI 設計の原則
		DES.hci.2	モードやナビゲーションの使用
		DES.hci.3	コード化の技法とビジュアルデザイン (色、アイコン、フォントなど)
		DES.hci.4	応答時間とフィードバック
		DES.hci.5	デザインのリテラシー (直接操作、メニュー駆動、フォーム、問題回答型、コマンドなど)
		DES.hci.6	ローカライゼーションと国際化
		DES.hci.7	HCI 設計の技法
		DES.hci.8	インタフェースのリテラシー (音声と自然言語、音楽/映像、触感など)
		DES.hci.9	メタファとコンセプトモデル
		DES.hci.10	HCI の心理学
開発マネジメント			
DES			ソフトウェア設計
	DES.ar		アーキテクチャ設計
		DES.ar.7	プロダクトのアーキテクチャと開発組織および市場の構造との関係
PRF			プロフェッショナルプラクティス
	PRF.com		(SE に特化した) コミュニケーションスキル
		PRF.com.1	読解、理解、要約 (ソースコードやドキュメントなど)
			記述 (職務記述書、報告書、評価報告書、理由書など)
		PRF.com.2	チームとグループのコミュニケーション (口頭、文書、電子メールなど)
		PRF.com.4	プレゼンテーションスキル
	PRF.pr		プロフェッショナリズム
		PRF.pr.1	アクレディテーション、資格認定、免許制度
		PRF.pr.2	倫理綱領とプロフェッショナルとしての行動

		PRF.pr.3	社会的、法的、歴史的およびプロフェッショナルとしての考慮事項
		PRF.pr.4	プロフェッショナル・ソサエティ（学会や協会、コミュニティなど）の起源と役割
		PRF.pr.5	標準の起源と役割
		PRF.pr.6	ソフトウェアの経済的重要性
		PRF.pr.7	雇用形態・雇用契約
	PRF.psy		グループダイナミクスと心理学
		PRF.psy.1	チームやグループでの作業の際のダイナミクス
		PRF.psy.2	個人の認知（制限など）
		PRF.psy.3	認知的問題の複雑性
		PRF.psy.4	ステークホルダとの対話
		PRF.psy.5	不確実性と曖昧性の取り扱い
		PRF.psy.6	多国籍・多文化環境の取り扱い
PRO			ソフトウェア開発プロセス
	PRO.con		プロセスの基礎
		PRO.con.2	プロセスのインフラストラクチャ（開発者、ツール、教育など）
	PRO.imp		プロセスの実装
		PRO.imp.1	プロセス定義のレベル（組織、プロジェクト、チーム、個人など）
		PRO.imp.2	ライフサイクルモデルの特性（計画ベース、インクリメンタル、繰り返し、アジャイルなど）
		PRO.imp.4	チームによるソフトウェアプロセス（モデル、定義、組織、測定、分析、改善）
		PRO.imp.5	システムエンジニアリングにおけるプロセスの実装
		PRO.imp.7	プロセスの外的要因の影響（契約と法的要求、標準、調達など）
	PRO.pp		プロジェクトの計画
		PRO.pp.1	要求のマネジメント（プロダクトのバックログ、優先順位、依存関係、変更など）

		PRO.pp.2	工数の見積り（過去データの利用、コンセンサスに基づく見積もり技術など）
		PRO.pp.3	作業の細分化とタスクスケジューリング
		PRO.pp.4	リソースの割り当て
		PRO.pp.5	リスクマネジメント（識別、緩和、修復、状況のトラッキングなど）
		PRO.pp.6	プロジェクト追跡のためのメトリクスと技術（データの取得、速度、バーンダウンチャート、欠陥追跡、技術的負債のマネジメントなど）
		PRO.pp.7	チームの自己管理（進捗追跡、動的負荷割り当て、緊急事態への反応など）
QUA			ソフトウェア品質
	QUA.pca		プロセスの保証
		QUA.pca.2	品質計画
セキュアソフトウェア開発概論			
SEC			セキュリティ
	SEC.dev		安全なソフトウェアの開発
		SEC.dev.1	セキュリティを組み込んだソフトウェア開発ライフサイクル
		SEC.dev.2	要件分析と仕様のセキュリティ
		SEC.dev.3	セキュアデザインの原則とパターン
		SEC.dev.4	安全なソフトウェア開発の技術
		SEC.dev.5	セキュリティに関連した検証と妥当性確認

付録2 J17-SE の情報科学基礎科目（抜粋）

授業科目名	<b>コンピュータとソフトウェアの基礎</b>		
単位数	2		
開設学期	1 年生前期		
目的	コンピュータの構成と動作原理やシステムの基本的な概念、さらに、コンピュータ上で動作するソフトウェアおよびその開発技術の概念と意義について理解することを目的とする。		
概要	コンピュータの概念、基本構造、データのメモリ上での表現、アセンブリ言語の構成、メモリシステム、入出力インタフェースといった一連のコンピュータの構成とシステムの基本的な概念、ならびにソフトウェア開発の全体像を学ぶ。		
目標	コンピュータの構成と動作原理、システムの基本的な概念、ソフトウェアおよびその開発技術の基本的な概念を習得する。		
先修科目			
関連科目			
授業方法			
評価方法・基準			
授業回数	授業展開	トピック	学習目標
1	SE コースを概説し、工学的アプローチについて述べる。また本講義の概要と目的および進め方について説明する（※講義 0.3h）。	コースの全体説明 工学的アプローチについて	
2	コンピュータの概念と基本的な構造を説明する	コンピュータの構成、バイトとビット、アドレス	CMP.cf.5
		メモリ、CPU、I/O	CMP.cf.5
		ノイマン型コンピュータの仕組み	CMP.cf.5

3	コンピュータの命令語の構成と種類、割り込みについて説明する	アセンブリ言語の命令の構成、基本的な命令の種類、割り込み	CMP.cf.5
4	2進数とN進数、数の計算機上での表現、各種文字コード、データの表現方法について説明する	2進数の演算、N進数、2進数と8進数、10進数、16進数の相互変換	CMP.cf.5
		BCD、EBCDIC、ASCII、漢字コード	CMP.cf.5
		計算機上での各種データの表現	CMP.cf.5
5	メモリ階層について紹介し、仮想記憶方式とキャッシュの仕組みについて説明する	仮想記憶方式の仕組み	CMP.cf.5
		キャッシュの仕組み	CMP.cf.5
6	入出力、割込み、インタフェース、ネットワーク技術とネットワーク階層について説明する	入出力、割込み、インタフェース	CMP.cf.5
		ネットワーク技術、ネットワーク階層	CMP.cf.11
7	プログラムの概念と定義、特長、言語処理系、各種のプログラム言語について説明する	プログラムの概念と定義、利点	CMP.cf.1
		コンパイラ、インタプリタ	CMP.cf.9
		手続き型、オブジェクト指向型、関数型、論理型など各種プログラム言語	CMP.cf.8
8	プログラムの基本となる制御構造、アルゴリズムの記法、構造化プログラムについて説明する	順接、選択、繰り返し	CMP.cf.1
		アルゴリズムとその記法	CMP.cf.2

9	構造化プログラム、プログラムとソフトウェアについて説明する	構造化プログラムの概念と利点、モジュール特性とメトリクス	CMP.cf.5
		プログラムとソフトウェアの共通点と差異	CMP.cf.5
10	ソフトウェア工学の背景、定義、成果について説明する	ソフトウェア工学の背景(歴史的経緯、実例、工学の必要性)	
		ソフトウェア工学の定義(プロダクト、プロセス、ピープル、プロジェクト、QCD)	
11	ソフトウェア開発プロセスとプロセスモデルについて説明する	プロダクトの要求、分析、設計、構築、V&V、保守・再利用	
		ソフトウェアライフサイクルプロセス、プロセスモデル	
12	開発プロジェクトとプロジェクト管理、要員、ソフトウェア品質、コスト、納期について説明する	プロジェクト	
		ピープル(人間系)	
		品質・コスト・納期(QCD)	
13	システムエンジニアリングと各種システムについて説明する	システムとは、システムの概念と利点、システムの分類	
		情報システムとコンピュータシステムの違い	
		システムとハードウェアとソフトウェアの関わり	

		組み込みシステムの紹介	
		企業情報システムの紹介	
14	重要事項のまとめ		
15	期末試験		
教科書・参考書	ロジャー・プレスマン: 実践ソフトウェアエンジニアリング, 日科技連, 2005. エリック・プロディ: ソフトウェアエンジニアリング, 翔泳社, 2004		
備考			

付録 3 J17-SE の SE 科目 (抜粋)

授業科目名	<b>ソフトウェア構築</b>		
単位数	2		
開設学期			
目的	ソフトウェアを構築するにあたり必要とされる過程を理解し、構築の際に用いられる技術について学習することを目的とする。		
概要	ソフトウェアの構築とは何を行うことかを習得する。ソフトウェアを構築するにあたり、必要とされる過程とその重要性、および構築のための要素技術を習得する。		
目標	作成されるソフトウェアの設計内容が与えられた際、自分で、あるいは多人数で共同してソフトウェアを構築する能力を身につける。ソフトウェアを構築する他の作業者に、ソフトウェアを構築するために必要な作業が何かを指示し管理する能力を身につける。		
先修科目			
関連科目	ソフトウェア設計とHCI, 開発プロセスと保守, ソフトウェア開発マネジメント, ソフトウェアアーキテクチャ		
授業方法	講義		
評価方法・基準	期末試験の成績により決定する。		
授業回数	授業展開	トピック	学習目標
1	授業の概要と目的および進め方について説明する。		
	ソフトウェア開発の全体的な流れについて説明する。また、その流れの中でソフトウェア構築がどのような位置づけであるか説明する。	プロセスの基礎	PRO.con

	ソフトウェア開発の要素としてどのような作業があるかを理解する.	プロセスの基礎	PRO.con
2	ソフトウェア開発の全体的な流れの中で、ソフトウェア構築がどのような位置づけであるか、ソフトウェア設計との関連から説明する.	設計に用いられる概念	DES.con
3	ソフトウェア構築における開発環境の役割について説明する.	開発環境	CMP.tl.1
		開発環境の認識	CMP.tl.1
4	ソフトウェア設計時に用いるプログラミング言語としてオブジェクト指向言語を取り上げ、その特徴について説明する.	プログラミング言語基礎	CMP.cf.8
		オブジェクト指向パラダイムにおける実行時のトピック(ポリモルフィズム, ダイナミックバインディングなど)	CMP.ct.3
5	ソフトウェア設計時に用いるプログラミング言語としてオブジェクト指向言語を取り上げ、その特徴について説明する.	パラメータ化と汎化	CMP.ct.4
		アサーション, 契約による設計(DbC), 防御的プログラミング	CMP.ct.5
		エラーハンドリング, 例外処理, フォールトトレラント	CMP.ct.6
6	ソフトウェア設計時に用いるプログラミング言語としてオブジェクト指向言語を取り上げ、その特徴について説明する.	ユーザーインタフェースフレームワーク, ツール	CMP.tl.2
	単体テストツールを用いることにより, ソフトウェア構築をどのように行うことが出来るかを説明する.	単体テストツール	CMP.tl.3
		単体テストツールの利用	CMP.tl.3
7	複数の実行が並列・並行に動作するソフトウェアの動作原理について説明する. また, そのような	並列処理の基本要素(セマフォ, モニターなど)	CMP.ct.10

	ソフトウェアを構築するための技術について説明する.	分散ソフトウェアのための構築技術	CMP.ct.11
8	ソフトウェア設計の際に状態やテーブルが用いられた際、それをソフトウェア構築時どのようにして実装することが出来るかを説明する.	状態ベースおよびテーブル駆動の構築技法	CMP.ct.7
		状態ベースおよびテーブル駆動を用いたプログラムの作成	CMP.ct.7
9	ソフトウェアの振舞いをその実行時に決定する方法について説明する. また、その一例としてソフトウェアの国際化を取り上げ、国際化がどのようにして行われるかを説明する.	実行時コンフィグレーションと国際化	CMP.ct.8
10	文字列等を入力とするソフトウェアを対象とし、入力内容をソフトウェア内部で利用するために必要となる処理について説明する.	文法ベースの入力処理(パース処理)	CMP.ct.9
		文法ベースの入力処理を行うプログラムの作成	CMP.ct.9
11	実世界のソフトウェアにおける特定のドメインに関するソフトウェアについて説明する.	組込みシステムの構築とハードウェア・ソフトウェア協調設計	CMP.ct.11
			CMP.ct.12
12	ソフトウェアを構築する段階における、ソフトウェア分析手法について説明する.	ホットスポット分析とパフォーマンスチューニング	CMP.ct.13
		プロファイリング・パフォーマンス分析のツール	CMP.tl.4
		静的解析と動的解析	MAA.af.2, VAV.rev.3 (動的解析に関しては対応なし)

13	ソフトウェアを構築する際、既存のソフトウェアを流用することによって効率的にソフトウェアを作成する方法について説明する。また、既存のソフトウェアを流用することによる効果について説明する。	API の設計と利用	
			CMP.ct.1
		コードの再利用とライブラリ	CMP.ct.2
			CMP.tl.4
14	重要事項のまとめ		
15	期末試験		
教科書・参考書			
備考			