

情報専門学科におけるカリキュラム標準(J07)  
- BOK -  
に対する産業界コメントへの対応

情報処理学会情報処理教育委員会  
J07プロジェクト連絡委員会

## J07(中間報告)-BOKIに対するコメントおよびその対応

コンピュータ科学 (CS: computer science)		コア	変更	C委員	変更	F委員	変更	H委員
DS	情報の基礎となる数学など	41		数学系は、全領域共通基礎知識として、別分類としたほうが良いのでは。				
DS1	関数, 関係, 集合	6						
DS2	論理	6						
DS3	グラフ	4						
DS4	証明技法	8						
DS5	数え上げと離散確率の基礎	7						
DS6	オートマトンと正規表現	6						
DS7	計算論概論	4						
DS8	計算論	-						
PF	プログラミングの基礎	38						
PF1	プログラミングの基本的構成要素	9						
PF2	アルゴリズムと問題解決	6						
PF3	基本データ構造	14						
PF4	再帰	5						
PF5	イベント駆動プログラミング	4						
AL	アルゴリズムの基礎	18						
AR	アーキテクチャと構成	33						
OS	オペレーティングシステム	15		組込みシステムへの適用は、日本として推進すべきではないか。				
OS1	オペレーティングシステムの概要	1						
OS2	オペレーティングシステムの原理	2						
OS3	プロセスの構造とスケジューリング	2						
OS4	並行性	4						
OS5	メモリ管理	4						
OS6	デバイス管理と入出力	-						
OS7	ファイルシステム	1						
OS8	認証とアクセス制御	1						
OS9	セキュリティと高信頼化	-						
OS10	リアルタイムシステムと組込みシステム	-	+					
OS11	並列・分散処理のためのオペレーティングシステムの機	-						
OS12	オペレーティングシステム構成法	-						
OS13	システム性能評価	-						
NC	ネットワークコンピューティング	14						
PL	プログラミング言語	19						
HC	ヒューマンコンピュータインタラクション	8						

初等解析や線形代数は、理系共通基礎の数学として、共通の内容を想定していますが、離散構造DSはCS領域で学習すべき専門のエリアとしています。他の領域でDSをエリアとして含んでいる場合も、それに含まれる各ユニットの内容やコアの範囲は異なっていて、領域間で共通にすることは難しいと思われます。

このカリキュラムでは具体的な教え方までは論じません。また、コア時間の合計をあまり増やすわけにはいかないという制約があります。

アルゴリズムは、プログラム全般の基礎になるものと思われる。アルゴリズムの設計手法・設計例を学ぶものとなっているが、アルゴリズムで必要とされるものは、トレースだと思える。トレースを中心に+10単位ほど必要なのではないか。

カリキュラム設計の際に、取り上げる可能性のあるテーマとして真剣に議論しましたが、まだコアとするレベルには達していないと判断しました。今後の検討課題の一つだと考えます。

コンピュータ科学 (CS: computer science)			コア	変更	C委員	変更	F委員	変更	H委員	
MR		マルチメディア表現	3		情報理論として習得させたい。 MR全体を情報理論の観点から習得させることは教え方の問題で、ここでは扱いませんが興味深いと思われます。最終報告ではMR3に情報理論が入っています。					
	MR1	情報理論	2	+						
	MR2	文字コード	1							
	MR3	標本化・量子化と圧縮	-	+						
	MR4	マルチメディア機器	-							
	MR5	オーサリング	-							
	MR6	メディア・インタラクション	-							
GV		グラフィックスとビジュアル・コンピューティング	3						検討課題とします。	
IS		インテリジェントシステム	3							
IM		情報管理	14		中間報告書を出した後で SE のコア時間に大きな見直しが入りました。中間報告書時点のコア時間は 20 時間でしたが、現在の案では 32 時間です。プロジェクト管理もコアとして入っています。要求分析という作業は顧客から言われた要求を受け入れるという作業ではなく、さまざまな関係者、文書、既存システム、法令などから、あるべき要求を決定してプロセスで、その中には当然業務分析も含まれます。				業務系アプリケーションでは、データベースのスキーマ構造を理解することから始まり、システムの拡張性やパフォーマンスを考慮した設計をすることが重要になる。その意味で、「IM6:関係データベース設計」で教える内容は、コアの位置づけで扱うべきものとする。また、「IM7:トランザクション処理」についても、業務系アプリケーションでは必須となる事項なので、概念と必要性だけでも、コアの中で多少の時間を割いて説明しておいていただきたいと思う。	
	IM1	情報モデルとシステム	2							
	IM2	データベースシステム	2							
	IM3	データモデリング	4							
	IM4	関係データベース	3							
	IM5	データベース間合わせ言語	3							
	IM6	関係データベース設計	2							
	IM7	トランザクション処理	1							
	IM8	分散データベース	-							
	IM9	データベースの物理設計	-							
	IM10	データマイニング	-							
	IM11	情報格納と検索	-							
	IM12	ハイパーテキストとハイパーメディア	-							
	IM13	マルチメディア情報とシステム	-							
	IM14	電子図書館	-							
SP		社会的視点と情報倫理	11							
SE		ソフトウェア工学	20		CSと言えども、学生を受け入れる立場としては、プロジェクト管理を外してほしくない。 中間報告書を出した後で SE のコア時間に大きな見直しが入りました。中間報告書時点のコア時間は 20 時間でしたが、現在の案では 32 時間です。プロジェクト管理もコアとして入っています。要求分析という作業は顧客から言われた要求を受け入れるという作業ではなく、さまざまな関係者、文書、既存システム、法令などから、あるべき要求を決定してプロセスで、その中には当然業務分析も含まれます。				「SE5:ソフトウェア要求及び仕様」については、要求分析の前に、現状の業務分析を行なうことを盛り込んだほうが良いと思う。お客様から言われた要求を受け入れることも重要だが、新システム導入のタイミングで、BPRができることはさらにシステムの価値を生む。「SE6:ソフトウェア妥当性検査」がコア科目に含まれていることは高く評価する。開発能力が高くて、ソフトウェアの品質に対する意識が低いと、ソフトウェアに対する信頼性を損ねる恐れが大きいからであ	
	SE1	ソフトウェア設計	5							
	SE2	APIの使用	2							
	SE3	ソフトウェアツールおよび環境	3							
	SE4	ソフトウェアプロセス	2							
	SE5	ソフトウェア要求および仕様	5							
	SE6	ソフトウェア妥当性検査	3							
	SE7	ソフトウェアの進化	-							
	SE8	ソフトウェアプロジェクト管理	-	+						
	SE9	コンポーネントベース開発	-							
	SE10	形式手法	-							
	SE11	ソフトウェアの信頼性	-							
	SE12	専用システムの開発	-							
CN		計算科学と数値計算	-							

## JO 7 (中間報告) -BOKに対するコメントおよびその対応

## &lt;対応について&gt;

ISでは、個々のBOKを取り上げて教えるのではなく、教育目的と学習レベルを明確に示したラーニングユニットを設計し、そこに関連するBOKを紐づけるという方法を採用しています。そのため、BOKは複数のラーニングユニットに組み込まれています。

科目はラーニングユニットを組み合わせて編成し、したがって、カリキュラムもラーニングが基になります。

この思想は最終報告書に反映されています。

コメントをいただいた中間報告書は、まだBOKだけしか提示できていませんでした。ご指摘いただいた教育内容については、ラーニングユニットの構成の中でできる限り反映できるよう努めました。

インフォメーションシステム (IS: information systems)			最低 レベ ル	変更	A委員	B委員	C委員	F委員	H委員	
IS1		情報技術								
	IS1.1	コンピュータアーキテクチャ	1			これらの基礎的な内容は、OSなどの別セクションとの重複が見受けられるため、各セクションの共通事項としてまとめるほうが学習者にはわかりやすいと思われます。				
	IS1.1.1	基本的なデータの表現	2							
	IS1.1.2	デジタル化された情報の物理的な表現	1							
	IS1.1.3	CPUの構成	1							
	IS1.1.4	コンピュータシステムの構成要素	1							
	IS1.1.5	マルチプロセッサアーキテクチャ	1							
	IS1.1.6	デジタル論理とシステム	1							
	IS1.2	アルゴリズムとデータ構造	2			IS1.1と同様				
	IS1.3	プログラミング言語	2		オブジェクト指向言語の基礎知識は必要。オブジェクト指向言語を使って手続き型プログラムを書いている例があとを絶たないため。	IS1.1と同様	コンポーネント指向の方が適すると考える。			
	IS1.3.1	基本的なプログラミング言語の構造	2							
	IS1.3.2	機械語とアセンブリレベルの言語	0							
	IS1.3.3	手続き型言語	2							
	IS1.3.4	非手続き型言語	0							
	IS1.3.5	第4世代言語	0							
	IS1.3.6	言語のオブジェクト指向への拡張	1+							
	IS1.3.7	プログラミング言語、設計、実装と比較	0							

インフォメーションシステム (IS: information systems)			最低 レベル	変更	A委員	B委員	C委員	F委員	H委員
IS1.4		オペレーティングシステム (OS)	2			IS1.1と同様			
IS1.5		通信	3+		通信については、データベースに準じて重要度が高まっている。連携が高まり、個々の情報システムが単独では機能しなくなって来ているため。(小項目への配分はお任せします。)	IS1.1と同様			
	IS1.5.1	国際通信標準、モデル、傾向	1						
	IS1.5.2	データの伝送	1						
	IS1.5.3	回線構成	1						
	IS1.5.4	ローカルエリアネットワーク	2						
	IS1.5.5	広域ネットワーク	1						
	IS1.5.6	ネットワークアーキテクチャとプロトコル	2						
	IS1.5.7	インターネット接続	2						
	IS1.5.8	ネットワーク設定、性能解析および監視	2						
	IS1.5.9	ネットワークのセキュリティ	2						
	IS1.5.10	高速ネットワーク	2						
	IS1.5.11	ネットワーク技術の最新の話題	2						
	IS1.5.12	通信アプリケーション	2						
	IS1.5.13	オープンシステムのプロトコル	1						
	IS1.5.14	情報分散	1						
IS1.6		データベース	4		IS1.1と同様			レベル4の内容に、時代が古すぎて不適切なものがある。再度検討していただきたい。 →1.6.6.7 DAO	
	IS1.6.1	DBMS	1						
	IS1.6.2	データモデル	4						
	IS1.6.3	トランザクション	2						
	IS1.6.4	整合性	4						
	IS1.6.5	データ定義言語	4						
	IS1.6.6	アプリケーションインタフェース	4						
	IS1.6.7	知識問合せプロセッサと問合せ機構、OLAPツール	1						
	IS1.6.8	分散型データベース、リポジトリとデータウェアハウス	1						
	IS1.6.9	DBMS製品：データベースシステムの最新状況	1						
	IS1.6.10	データベースマシンとサービス	1						
	IS1.6.11	データとデータベースの管理	2						
	IS1.6.12	データ辞書、事典、リポジトリ	2						
	IS1.6.13	情報検索	4						
IS1.7		人工知能	1		IS1.1と同様				

インフォメーションシステム (IS: information systems)			最低レベル	変更	A委員	B委員	C委員	F委員	H委員
IS2		組織と管理概念							
	IS2.1	組織理論一般	3	++	情報システムへの変更要求は、組織変更によるものが多い。情報システムを設計する際、これを活用する組織の現状だけではなく組織の構造と経営の要求に基づく変更の可能性を見抜いた構造化を行っておくことで、後の変更要求への対応の可能性が高まる。また、組織のミッションを理解することにより、情報要求の理解が高まり、実装形態に左右され難い有用な情報が設計できる。(小項目はどれも重要と認識します。)	内部統制に関するカリキュラムを明確にしたほうがよいのではないのでしょうか。			
		IS2.1.1 組織の階層とフローモデル	3	++					
		IS2.1.2 組織上の作業グループ	3	++					
		IS2.1.3 組織のスパン(単一ユーザ、作業グループ、チーム、企業、グローバル)	3	++					
		IS2.1.4 企業内でのISの役割	3	++					
		IS2.1.5 組織構造におけるISの影響	3	++					
		IS2.1.6 組織の構造(集権型、分権型、マトリクス型)	3	++					
		IS2.1.7 組織でのソフトウェア使用に関する組織的問題	3	++					
	IS2.2	情報システム管理	5		この中項目を「5」として いる点に共感します。	ISの位置づけを説明する部分、と捉えれば適切な内容と思われます。ただ内容によってはIS3以降のカリキュラムと重複することが懸念されます。  また、セキュリティに関する項目が複数に分かれています。情報セキュリティとしてまとめはいいのでしょうか。			
		IS2.2.1 IS計画	5						
		IS2.2.2 IS機能のコントロール	3						
		IS2.2.3 スタッフ配置と人的資源管理	3						
		IS2.2.4 ISの機能構造(企業内対アウトソーシング)	2						
		IS2.2.5 IS組織の目的と目標の決定	3						
		IS2.2.6 ビジネスとしてのIS管理	3						
		IS2.2.7 CIOとスタッフの機能	1						
		IS2.2.8 サービス機能としてのIS	3						
		IS2.2.9 ISの財政管理	2						
		IS2.2.10 ISの戦略的な使用	2						
		IS2.2.11 知的作業、エンドユーザコンピューティング	3						
		IS2.2.12 ISの方針、運用手順の公式化およびコミュニケーション	3						
		IS2.2.13 バックアップ、災害対策、および復旧の計画	2						
		IS2.2.14 新しい技術の管理	1						
		IS2.2.15 サブ機能の管理	1						
		IS2.2.16 セキュリティと管理、ウイルスとシステムの安全性	2						
		IS2.2.17 コンピュータ運用の管理	2						
	IS2.3	意思決定理論	3		情報システムの専門化がユーザ組織(経営層から末端の担当者を含む)を相手に意思決定の支援を行うことは必須です。プロとしてサービスできるレベルの訓練を充実させるべきと考えます。	経営的な視点からの内容であるならば、その視点を明確にした内容にすることで、学習者は立場の相違による見方の違いを理解しやすいかと思われます。			
		IS2.3.1 計測とモデル化	3						
		IS2.3.2 確実性、不確実性およびリスクの下での意思決定	3						
		IS2.3.3 情報のコスト/価値、ISの競争可能な価値	3						
		IS2.3.4 意思決定モデルとIS(最適化、満足化)	3						
		IS2.3.5 グループの意思決定プロセス	4	+					

インフォメーションシステム (IS: information systems)			最低レベル	変更	A委員	B委員	C委員	F委員	H委員
IS2.4		組織行動	3		IS2.4で3がついている小項目をIS2.3.5と併せて学習するとよいと考えます。	IS2.3と同様			
	IS2.4.1	ジョブ設計理論	1						
	IS2.4.2	文化の多様性	3						
	IS2.4.3	グループダイナミクス	1						
	IS2.4.4	チームワーク、リーダーシップおよび権限委譲	3	IS2.6は、プロジェクト管理の基礎になりますので、情報システムを提供するプロセスを理解する上で非常に重要と考えます。業務設計を行う上でのスケジューリングであったとしても、業務システムとしての情報システム設計のためには「時間」の概念は非常に重要です。					
	IS2.4.5	影響力、権限、政策の行使	1						
	IS2.4.6	認知スタイル	1						
	IS2.4.7	交渉と交渉スタイル	1						
	IS2.4.8	合意の形成	3						
IS2.6		スケジューリングの理論と概念	3++		IS2.3と同様				
	IS2.6.1	スケジューリングの目的	3++						
	IS2.6.2	スケジューリングの方法	4+++						
	IS2.6.3	TOC (制約理論)	3++						
IS2.7		変革プロセスの管理	2		この中項目も非常に重要と考えますが、配分を高めるのは難しいでしょうか。	IS2.3と同様			
	IS2.7.1	変革に抵抗する理由	1						
	IS2.7.2	変革を動機づける戦略	1						
	IS2.7.3	変革の計画づくり	1						
	IS2.7.4	変革の管理	1						
	IS2.7.5	モデル化プロセスとシステム	1						
	IS2.7.6	データ取得手段の実験	1						
	IS2.7.7	プロセスや関係するソフトウェアの改良でのリーダーシップ	1						
	IS2.7.8	対処方略	1						
	IS2.7.9	グループおよびチーム学習	2						
	IS2.7.10	変革者の特性	1						
IS2.8		ISの法的、倫理的側面	2		個人情報保護や情報セキュリティに関するカリキュラムを明確にしたほうがよいのではないのでしょうか。				契約の基礎で、準委任型の契約と請負型の契約の特徴をきちんと教えるとともに、情報システム開発における「モデル取引・契約」の概要を教えるべきである(可能であれば、どのようなトラブルがあるのかを紹介する)
	IS2.8.1	ソフトウェアの販売・使用許諾および取次ぎ	1						
	IS2.8.2	契約の基礎	2						
	IS2.8.3	プライバシー法	2						
	IS2.8.4	取次ぎと規制集団	1						
	IS2.8.5	知的財産権の保護と倫理	2						
	IS2.8.6	倫理と法律、倫理モデル、倫理的社会的分析	2						
	IS2.8.7	計算機アプリケーションのリスク、損失および責任	2						
	IS2.8.8	保証	2						
	IS2.8.9	コンピュータ犯罪	2						

インフォメーションシステム (IS: information systems)			最低レベル	変更	A委員	B委員	C委員	F委員	H委員
IS2. 9		プロフェッショナリズム	2			学術的な側面だけでなく、業界標準があるものに関してはそれらについても触れておくのはいかがでしょうか。			
	IS2. 9. 1	現行の定期的、専門的、学術的刊行物	1						
	IS2. 9. 2	証明書の発行	1						
	IS2. 9. 3	専門組織（学協会など）	1						
	IS2. 9. 4	専門家会議	1						
	IS2. 9. 6	IS産業	2						
	IS2. 9. 7	コンピューティングの歴史的社会的な文脈	1						
IS2. 10		対人関係の能力	3		IS2. 10の重要性は痛いほどわかります。しかしこの領域は、頭でわかって口で説明できても何の役にも立ちません。教室での知識提供が中心となる大学教育において、この領域をきちんと理解させるには時間が足りないのではないのでしょうか。できればレベル4を目指すべきですが、教室では、諦めて1に留めることになるでしょう。この領域の教室での講義や学習は1に留めておき、他の項目の演習や教授・講師の態度を通じて、4年間かけて伝えていくということではいかがでしょうか。そういう意味では、1年生の早い時期に知識として与え、日々の学習	この分野は机上の知識等では習得できない内容が多いものです。ここでは基本的な手法や理論を紹介した上で、別カリキュラムでケーススタディやロールプレイなども実施すると、社会での活躍が早く期待できるのではないのでしょうか。米国のMBAでも専門知識に加えてこのような実務擬似体験を重視していると聞きます。			
	IS2. 10. 1	コミュニケーション能力	3						
	IS2. 10. 2	インタビュー、質問、傾聴	1	-					
	IS2. 10. 3	プレゼンテーションの技能	1	-					
	IS2. 10. 4	コンサルティングの能力	1	-					
	IS2. 10. 5	執筆能力	1	-					
	IS2. 10. 6	積極的な態度と取組み	1	-					
	IS2. 10. 7	個人の目標の設定、意思決定、時間管理	1	-					
	IS2. 10. 8	原則を中心としたリーダーシップ	1	-					
	IS2. 10. 9	交渉の原則	1	-					
	IS2. 10. 10	創造性と機会発見力	1	-					
	IS2. 10. 11	批判思考	1	-					
	IS2. 10. 12	データの測定と分析	1	-					
	IS2. 10. 14	個人の問題解決	1	-					
	IS2. 10. 15	発想法	3						



インフォメーションシステム (IS: information systems)			最低レベル	変更	A委員	B委員	C委員	F委員	H委員
IS2.11		基本的な組織の機能	4	++	<p>基礎的なビジネスモデルの理解は、ビジネスアプリケーションを提案・設計するシステムエンジニアには必須の基礎知識であると考えます。社会人経験のない学生に企業のビジネスモデルを理解させることは至難のわざと思われませんが、努力を放棄してしまっはいけない領域だと考えます。よろしくお願いします。</p>	<p>要件定義に関するカリキュラムとしてのまとめがあると思われるかと考えられます。</p>		<p>・ ビジネスモデルを用いた講義時間を確保していることに賛成である。</p>	<p>この中項目の内容は、モデルとした米国版の影響を受け過ぎていると思われる。→ 発送 (Shipping) を船積みと訳したところなど。もう少し一般的な業務形態とフローをとりいれ、作り直す必要がある。</p>
	IS2.11.1	支払い	2	+					
	IS2.11.2	ビジネスの関係 (例: CtoB、BtoB、BtoGなど)	2	+					
	IS2.11.3	ビジネスのモデル、伝統的/電子的商取引	2	+					
	IS2.11.4	バリューチェーン (VC) の概念	2	+					
	IS2.11.5	サプライチェーンマネジメント (SCM) の概念	2	+					
	IS2.11.6	アテンション	2	+					
	IS2.11.7	マーケティングと広告	2	+					
	IS2.11.8	小売り	2	+					
	IS2.11.9	製造業と製品	2	+					
	IS2.11.10	人的資源管理とコンプライアンス	2	+					
	IS2.11.11	在庫管理	2	+					
	IS2.11.12	発送 (船積み)	2	+					
	IS2.11.13	調達	2	+					
	IS2.11.14	注文票と顧客サービス	2	+					
	IS2.11.15	会計監査と管理	2	+					
IS3		システムの理論と開発			<p>業務システム (=業務の仕組み) (例: 金融システム)、電子化された信号だけを扱うコンピュータシステム、更にその一部であるアプリケーションプログラムといったものが入れ子構造に組みあがった仕組みの全体像と、その中で流通する「情報」(≠画面・帳票)に関する正しい理解は、IS学士の必須学習項目と考えます。レベル2であっても、深い理解を促す講義をお願いします。</p>	<p>他のセクション等との重複感があるので、いずれかでまとめて示すことがよいのではないのでしょうか。</p>			
IS3.1		システムと情報の概念	2						
	IS3.1.1	一般システム理論	2						
	IS3.1.2	システム概念	2						
	IS3.1.3	開放系と閉鎖系	1						
	IS3.1.4	システムの構成要素と関係	2						
	IS3.1.5	システム管理	1						
	IS3.1.6	情報システムの特徴	2						
IS3.2		システム開発のアプローチ	5		<p>巷には、何とかアプローチというのが溢れています。学生が「新しい(提唱・実用化が遅かった)もの=良いもの」という誤謬に陥る前に、正しい知識を与えておく事は非常に重要と考えます。作り上げる構体の利用目的と規模によって適切なアプローチを選ぶという考え方を理解することは必須です。</p>	<p>この中に含めないとしても、具体的な事例教材があると実感としてわかりやすいかと思われます。</p>		<p>なぜ、システム開発モデルで「プロトタイプング」だけが取り上げられているのか。(欠陥のあるモデルであるものの)日本で最も利用されている「ウォーターフォール・モデル」や「スパイラル」、最近注目されている「アジャイル」などを教えるべきである。</p>	
	IS3.2.1	システム開発モデル	5						
	IS3.2.2	パッケージの取得と実装	3						
	IS3.2.3	ソフトウェア構成要素の統合	1						
	IS3.2.4	エンドユーザ開発のシステム	1						
	IS3.2.5	システム開発アプローチの選択	2	+					

インフォメーションシステム (IS: information systems)			最低レベル	変更	A委員	B委員	C委員	F委員	H委員
IS3.3		システム開発の概念と方法論	5						
	IS3.3.1	組織のモデル化及びソフトウェアプロセスのモデル化	1						
	IS3.3.2	データモデリング	4						
	IS3.3.3	データ指向方法論	1						
	IS3.3.4	プロセス指向方法論	1						
	IS3.3.5	行動指向方法論	1						
	IS3.3.6	オブジェクト指向方法論	2+						
	IS3.3.7	ソフトウェア工学のプロセスとプロダクト	4						
IS3.4		システム開発ツールと技術	5						
	IS3.4.1	CASE	1						
	IS3.4.2	グループベースの方式	1						
	IS3.4.3	ソフトウェア実装の概念とツール	4						
		リポジトリの概念と活用	4+						
IS3.5		アプリケーション計画	5						
	IS3.5.1	インフラストラクチャ計画	1						
	IS3.5.2	ISアーキテクチャの計画	2+						
	IS3.5.3	運用のための計画	4						
	IS3.5.4	システム規模、ファンクションポイント、複雑さ管理のメトリクス	1						
	IS3.5.5	ISセキュリティ、プライバシーおよび管理のための計画	1						
	IS3.5.6	発注管理	1						
IS3.6		リスク管理	2						
	IS3.6.1	実現可能性の評価	1						
	IS3.6.2	リスク管理の原則	1						
	IS3.6.3	不測事態対応計画	2						

インフォメーションシステム (IS: information systems)			最低レベル	変更	A委員	B委員	C委員	F委員	H委員
IS3.7		プロジェクト管理	3		IS3.7.1は、IS3.2.5のアプローチ選択に続いて、そのプロジェクト固有の制約や事情を加味してプロジェクトプロセスそのものを作り上げていく段階として切り分けないと、学生には、IS3.7.1とIS3.2.5の違いが分からないでしょう。小項目名も、例えば、「プロジェクト計画と適切なプロセスの組み立て」などにすると誤解が少ないと考えます。また、品質管理に関するカリキュラムは、プロジェクト管理の一環としての「システム開発の品質保証」だけではなく、独立して全フェーズを網羅するように位置づけるとよいのではないのでしょうか。	IS2.10と同様に、プロジェクト管理の基本的な考え方や方法を示した上で、実践を想定した演習が別途実施できれば、実務的には有効と思われる。			工数の見積もり手法はどこで教えるのか？
	IS3.7.1	プロジェクト計画と適切なプロセスモデルの選択	3						
	IS3.7.2	プロジェクトの組織、管理、原則、概念、問題	3						
	IS3.7.3	作業構造 (WBS) とスケジュール	3						
	IS3.7.4	プロジェクトスタッフの考え方	1						
	IS3.7.5	プロジェクトの管理	2						
	IS3.7.6	複数プロジェクトの管理	2						
	IS3.7.7	管理上の概念、ストレスと時間管理	1						
	IS3.7.8	システム文書の作成	1						
	IS3.7.9	ユーザ文書の作成	1						
	IS3.7.10	システムのメトリクス	1						
	IS3.7.11	スコープとスコープ管理	1						
	IS3.7.12	構成管理	1						
	IS3.7.13	システム開発の品質保証	1						
	IS3.7.14	プロジェクトの追跡	1						
	IS3.7.15	プロジェクトの完了	1						
IS3.8		情報とビジネスの分析	5		この領域をレベル5とした点に共感します。用語：「要求定義」であり、「要求開発」ではないですね。	順序として、現状分析と要件定義はIS3章のもう少し前に置くほうがわかりやすいかと思われます。			
	IS3.8.1	問題点と機会の認識	4						
	IS3.8.2	企業モデル	1						
	IS3.8.3	要求定義と仕様化	4						
IS3.9		情報システム設計	5		単純なマンマシンインターフェースという意味ではなく、経済面を加味した人とコンピュータの適切な分担の提案ができるという意味での理解が必要と考えます。	IS3.8と同様に、システム設計についてもIS3章のもう少し前に置くほうがわかりやすいかと思われます。			
	IS3.9.1	設計 (論理、物理)	4						
	IS3.9.2	設計方法論	5						
	IS3.9.3	設計目標	5						
	IS3.9.4	創造的な設計プロセスを促進する技術	1						
	IS3.9.5	情報表現の代替案、認知スタイル	4						
	IS3.9.6	人間とコンピュータの相互作用	3+						
	IS3.9.7	ソフトウェア開発	1						
	IS3.9.8	ソフトウェアアーキテクチャ	2						

インフォメーションシステム (IS: information systems)			最低レベル	変更	A委員	B委員	C委員	F委員	H委員
IS3.10	システムの実装とテスト戦略	1			IS3.10.8はソフトウェアのインストールでしょうか。 ※システムとソフトウェアの違い(業務システムと情報システムとコンピュータシステムの違い、ソフトウェアとプログラムの違い)をはっきりさせて共有しないため、混乱の元になっています。(SLCP98と2007では定義が変わってしまいました。)どれをどうということは三輪が決める立場にはありませんが、広く共有された解釈がないことは問題です。よろしくお願ひします。	処理方式や方式設計、移行に関するカリキュラムも実務上は重要かと思われる。			
IS3.10.1	システムの構築	1							
IS3.10.2	ソフトウェアシステムの構築	1							
IS3.10.3	ソフトウェアの統合	1							
IS3.10.4	システム移行	1							
IS3.10.5	システム統合とシステムテスト	1							
IS3.10.6	訓練	1							
IS3.10.7	ソフトウェアプロジェクトの管理	1							
IS3.10.8	システムのインストール	1							
IS3.10.9	実装後のレビュー	1							
IS3.11	システムの運用と保守	1		経営者が投資する対象としての情報システムの価値を量るという概念だけは、理解させておく必要があると考えます。	企業等が利用する実際の情報システムでは、この分野に必要な経営資源はかなり多いので、それがわかるといふのはいかがでしょうか。				
IS3.11.1	サービス要請と変更管理	1							
IS3.11.2	リバースエンジニアリングとリエンジニアリング	1							
IS3.11.3	調整と均衡化	1							
IS3.11.4	システムとソフトウェア保守の概念	1							
IS3.11.5	情報システムの評価	2+							
IS3.12	さまざまな情報システムの開発	3	--	IS2.11の方が重要と考えます。あるいは、IS2.11の講義の中でIS3.12のような分類軸があることを添えて、IS2.11の各要素の属性として、先の分類を補足すれば済むようにも思えます。いかがでしょうか。	さまざまな適用例として、ISの冒頭でも概略を紹介するのはいかがでしょうか。				
IS3.12.1	トランザクション処理システム	1							
IS3.12.2	経営情報システム	1							
IS3.12.3	グループ支援システム	1							
IS3.12.4	意思決定支援システム/エキスパートシステム	1							
IS3.12.5	エグゼクティブ情報システム	1							
IS3.12.6	オフィスシステム	1							
IS3.12.7	協調作業システム	1							
IS3.12.8	画像およびワークフローシステム	1							
IS3.12.9	機能的な支援システム	1							
IS3.12.10	企業間連携システム	1							
IS3.12.11	生産管理システム	1							
IS3.13	IS教育と環境	3							
IS3.14	IS人材の育成	2							
IS3.15	教育方法論	2							
IS3.16	その他の参照学問	1							

## J07(中間報告)-BOKに対するコメントおよびその対応

ソフトウェアエンジニアリング (SE: software engineering)				時間数	C委員	F委員	G委員	H委員
確率・統計				0	数学系は、全領域共通基礎知識として、別分類としたほうが良いのでは。  その分野の基礎は全領域共通であっても各分野で再掲することとしています。			必要ならば、教養基礎科目に回すのではなく、コアとしての時間を設定すべきである。  その分野の基礎は全領域共通であっても各分野で再掲することとしています。
FND		数理基礎・工学基礎						
	FND.mf	数理基礎						
	FND.mf.6	離散確率						
	FND.mf.9	数値誤差と精度						
	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)						
離散数学				22.5	数学系は、全領域共通基礎知識として、別分類としたほうが良いのでは。	その分野の基礎は全領域共通であっても各分野で再掲することとしています。		
論理と推論・計算理論				22.5	数学系は、全領域共通基礎知識として、別分類としたほうが良いのでは。	その分野の基礎は全領域共通であっても各分野で再掲することとしています。		
コンピュータとソフトウェア工学の基礎				22.5				ソフトウェア工学の基礎も、きちんとBOKとして大中小の項目IDを付与すべきである。
CMP		コンピュータ基礎						
	CMP.cf	コンピュータ科学基礎						
	CMP.cf.5	コンピュータの構造						
	CMP.cf.6	システムの基礎						
***	***.***	***.***.***	ソフトウェア工学の基礎					
オペレーティングシステム基礎・データベース基礎				22.5		異なる2つの意見があるように、SE委員会の中でも意見が分れました。ある分野の特徴を出したい学科はSAS科目の中で、その部分を強化できるということで、このようになりました。	OS・DB・NW基礎の目標が「開発に必要な程度の知識を得る」であることを考慮すると、他項目と比較して若干時間を割きすぎな感がある。	データベース関係の科目がこれだけしかないというのは、産業界では受け入れがたいことである。トランザクション処理なども業務用ソフトウェアでは必須であり、何らかの項目として取り上げる必要がある。
CMP		コンピュータ基礎						
	CMP.cf	コンピュータ科学基礎						
	CMP.cf.10	オペレーティングシステムの基礎						
	CMP.cf.11	データベースの基礎						
ネットワーク基礎				22.5				
一般工学基礎				22.5	SE以外の領域でも共通として必要ではないか。  J07委員会の方で検討を依頼します。			
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						
	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						

ソフトウェアエンジニアリング (SE: software engineering)				時間数	C委員	F委員	G委員	H委員
データ構造とアルゴリズム・プログラミング言語基礎				22.5				プログラマにとっては、プログラミング言語の文法を理解することが手始めにはなるが、応用していくためには、構造を見抜いて分解する能力や、パターン化できる能力が必要になると考える。 離散数学など、上記のFNDの項目を基礎として、構造を把握する能力を身に付けさせることで、プログラミング能力も向上するものと考ええる。
CMP			コンピュータ基礎					<div style="border: 1px solid black; padding: 5px; width: fit-content;">           論点別分類(SE領域)の方でお答えしました。         </div>
	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.cf.13	プログラミング言語の意味論					
	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)、防衛的プログラミング					
		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	アプリケーション指向言語(スクリプト言語、ビジュアル言語、ドメイン特化言語、マークアップ言語、マクロな					
		CMP.tl.5	プロファイリング・パフォーマンス分析・スライシングのツール					
ソフトウェア構築				22.5		ビジネスモデルを用いた講義時間も確保すべきと考える。 <div style="border: 1px solid black; padding: 5px; width: fit-content;">論点別分類(SE領域)の方でお答えしました。</div>		
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					
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					
ソフトウェアモデリングと要求開発				22.5	非機能要件についても扱ってほしい。 <div style="border: 1px solid black; padding: 5px; width: fit-content;">論点別分類(SE領域)の方でお答えしました。</div>	モデリングという用語はソフトウェア開発の重要な概念であり、この用語を冠した授業名を置くべきという議論でした。「形式手法」は設計以前の工程の用語ではないため、ここは「ソフトウェアモデリングと要求開発」としたいと思います。「要求開発」という用語については更に検討の必要があると考えています。		この内容は「形式手法」とまとめて一つの講義とすべきである。
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					
	MAA.rv		要求の評価					

ソフトウェアエンジニアリング (SE: software engineering)			時間数	C委員	F委員	G委員	H委員
ソフトウェアアーキテクチャ			22.5				ソフトウェア (モジュール、コンポーネントetc) の再利用に関する教育が必要、できればWebサービスについても。DES.con.6品質特性の設計の中に、セキュリティを含めるべきである。(FND.ef.4やVAV.tst.9Iには含まれている)
DES		ソフトウェア設計					<p><b>ソフトウェア構築</b> CMP.ct.11,DES.dd.3などでコンポーネントが触れられます。この中で再利用性などについて言及されるようにしたいと思います。Software Engineeringの一部の教科書ではComponent-based developmentはこれまでの開発方法と違う形で明記されるなど、その重要性は高まっていると考えますので、今後その扱いをWebサービスを含めて検討します。</p> <p>セキュリティの重要性の高まりを受けて、カリキュラムモデルではSAS科目(選択科目)の一例として、セキュリティの扱いに特化した科目「高セキュリティ情報システム」を挙げています。しかしながらご指摘のように、コア科目群中の品質関連科目において共通にセキュリティについても言及すべきことが望ましく、今後、科目「ソフトウェアアーキテクチャ」内における時間的な実現可能性も含めて言及を検討します。</p>
	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.con.8	アーキテクチャのスタイル、パターン、再利用					
	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					
	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					
	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					
	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	ドメイン特化アーキテクチャおよびプロダクトライン開発					
	DES.ar.6	アーキテクチャのための記法(アーキテクチャ上のビューポイントと表現、コンポーネント図など)					
	DES.ste	設計の支援ツールと評価					
	DES.ste.1	設計支援ツール(アーキテクチャ、静的解析、動的評価など)					
	DES.ste.2	設計上の特性の測定(結合度、凝集度、情報隠蔽、関心事の分離など)					
	DES.ste.3	設計のメトリックス(アーキテクチャ上の要因、変換、一般的な使い方におけるメトリックス)					
	DES.ste.4	フォーマルメソッドによる設計の分析					
ソフトウェア設計			22.5				
ソフトウェアV&V			22.5				

ソフトウェアエンジニアリング (SE: software engineering)			時間数	C委員	F委員	G委員	H委員
形式手法			22.5			形式手法は重要であり、また要求分析、定義工程だけに限られるものではないので、このような形としました。	この内容は「ソフトウェアモデリングと要求開発」とまとめて一つの講義とすべきである。
CMP		コンピュータ基礎					
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					
開発プロセスと保守			22.5				
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					
EVO		ソフトウェアの進化や保守					
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					
QUA		ソフトウェア品質			「ソフトウェア品質関連科目を減らさざるを得なかった」とのことだが、その分野はむしろ企業側で補完できるので、問題はない。	ありがとうございます。	
	QUA.cc	ソフトウェア品質の概念と文化					
	QUA.std	ソフトウェア品質に関する標準					
	QUA.pro	ソフトウェア開発プロセスの改善					
	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			22.5				
ソフトウェア開発マネジメント			22.5				
FND		数理基礎・工学基礎					
PRF		プロフェッショナルプラクティス					
MAA		ソフトウェアのモデリングと分析					
MGT		ソフトウェア開発のマネジメント					ソフトウェア開発の契約に関する教育は必要ないか？
	MGT.con	マネジメントの基礎					
	MGT.pp	プロジェクトの計画					
	MGT.per	プロジェクトのメンバと組織					
	MGT.ctl	プロジェクトのコントロール					
	MGT.cm	ソフトウェア構成管理					

SE領域はソフトウェア開発ライフサイクルを網羅するカリキュラムモデルとし、各工程の理解、品質・生産性・コストといった要因の理解、また、グループ作業作業、マネジメント、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力を持たずこととしております。「契約」に関することは明確に出ていませんが、PRF.psy.4 ステークホルダーとの対話や、MGT.con.5 調達マネジメントで触れられることとなります。「契約」についての扱いの重要性等については再度検討いたします。



J07(中間報告)-BOKに対するコメントおよびその対応

コンピュータエンジニアリング (CE: computer engineering)			コア	変更	C委員	変更	F委員	G委員
CE-ALG		アルゴリズム	25		CE-PRF「プログラミング」とセットとした方がよいのでは。アルゴリズムは、設計等、他の工程でも活用する知識なので独立しているかもしれないが、活用場面との関連性を持たせた方が理解が深まると考える。また、プログラミングこそ、アルゴリズムを注意深く検討すべきであると考え。		BOK(知識項目の整理)としては、別項目としていますが、具体的な科目を構成するときにはご指摘のように同じ科目の中に組み込む、同時期に平行した科目として設置するなどの工夫をすることがよいと考えています。	
CE-ALG0		歴史と概要	1					
CE-ALG1		基本アルゴリズムの分析	2					
CE-ALG2		アルゴリズム戦略	8					
CE-ALG3		アルゴリズムの複雑性	2					
CE-ALG4		アルゴリズムと問題解決	4					
CE-ALG5		データ構造	6					
CE-ALG6		再帰	2					
CE-ALG7		基本的計算可能性理論	-					
CE-ALG8		コンピューティングアルゴリズム	-					
CE-ALG9		分散アルゴリズム	-					
CE-CAO		コンピュータのアーキテクチャと構成	38					
CE-CSG		回路および信号	18					
CE-DBS		データベースシステム	5					
CE-DIG		デジタル論理	29					
CE-DSP		デジタル信号処理	17					
CE-ESY		組込みシステム	30		組込みならではの設計、実装に力を入れてほしい。コンパイラの原理についても触れてほしい。 BOKコアは、CEを対象とするすべて学科で与えるべき最低限を示すものです。設計・実装にどれだけの力を注ぐかは、それぞれの学科が判断し決定することになります。 コンパイラの原理を含め、ご指摘のCE-ESY14~CE-ESY17は、最終報告書でコアとして時間割当を行いました。			
CE-ESY0		歴史と概要	1					
CE-ESY1		低電力コンピューティング	2					
CE-ESY2		高信頼性システムの設計	2					
CE-ESY3		組込み用アーキテクチャ	6					
CE-ESY4		開発環境	2					
CE-ESY5		ライフサイクル	1					
CE-ESY6		要件分析	1					
CE-ESY7		仕様定義	1					
CE-ESY8		構造設計	1					
CE-ESY9		テスト	1					
CE-ESY10		プロジェクト管理	1					
CE-ESY11		並列設計(ハードウェア、ソフトウェア)	1					
CE-ESY12		実装	2					
CE-ESY13		リアルタイムシステム設計	8					
CE-ESY14		組込みマイクロコントローラ	-	+				
CE-ESY15		組込みプログラム	-	+				
CE-ESY16		設計手法	-	+				
CE-ESY17		ツールによるサポート	-	+				
CE-ESY18		ネットワーク型組込みシステム	-					
CE-ESY19		インタフェースシステムと混合信号システム	-					
CE-ESY20		センサ技術	-					
CE-ESY21		デバイスドライバ	-					
CE-ESY22		メンテナンス	-					
CE-ESY23		専門システム	-					
CE-ESY24		信頼性とフォールトトレランス	-					

コンピュータエンジニアリング (CE: computer engineering)			コア	変更	C委員	変更	F委員	G委員
CE-HCI		ヒューマンコンピュータインタラクション	7					
CE-NWK		テレコミュニケーション	22					
CE-OPS		オペレーティングシステム	22		RTOSを作る人をイメージすると、時間が短いのでは。また、RTOSを活用する人として、設計の原則はコアとした。いずれを目指すのか？	CEを対象とするすべての学科で最低限教えるべきものを定めたのがコアです。コアの上に、学科が目指すものに依りて、内容を深めたり、演習・実習をしたりしてカリキュラムを構成することになります。コアの割当は継続して見直していきますが、すべての学科共通という性格から大幅な時間増はできません。J07をベースとした具体的なカリキュラムの提示等で特色を出したカリキュラム構成例を示す努力をしていく予定です。		
	CE-OPS0	歴史と概要	1					
	CE-OPS1	並行性	6					
	CE-OPS2	スケジューリングとディスパッチ	3					
	CE-OPS3	メモリ管理	3					
	CE-OPS4	セキュリティと保護	2					
	CE-OPS5	ファイル管理	2					
	CE-OPS6	リアルタイムOS	3					
	CE-OPS7	OSの概要	2					
	CE-OPS8	設計の原則	-					
	CE-OPS9	デバイス管理	-					
	CE-OPS10	システム性能評価	-					
CE-PRF		プログラミング	14		コーディング作法の中身には触れなくとも、コーディング作法の必要性だけはコアとした。項目を整理し直し、コーディング作法もコアに含めた形としました。その結果は最終報告書に示してあります。			
	CE-PRF0	歴史と概要	1					
	CE-PRF1	プログラミングの構成	7					
	CE-PRF2	オブジェクト指向プログラミング	1					
	CE-PRF3	OSのシステムコールの使用	4					
	CE-PRF4	機器制御プログラミング	1					
	CE-PRF5	プログラミングのパラダイム	-					
	CE-PRF6	イベント駆動プログラミングとコンカレントプログラミング	-					
	CE-PRF7	APIの使用	-					
	CE-PRF8	コーディング作法	-	+				
CE-SPR		社会的な観点と職業専門人としての問題	16				「技術者倫理」の項目が取り上げられている点は大変意義深いと思う。学生にとっては、その目的や真意をただちに理解することは難しいかもしれないが、その後の企業活動の中で必ず生きてくる内容だと思う。  賛同いただきありがとうございます。JABEEなど専門教育認定でも技術者倫理は必須の項目となっています。	
	CE-SPR0	歴史と概要	1					
	CE-SPR1	公的ポリシー	2					
	CE-SPR2	分析の方法およびツール	2					
	CE-SPR3	社会的な観点と職業専門人としての問題	2					
	CE-SPR4	リスクと責任	2					
	CE-SPR5	知的財産権	2					
	CE-SPR6	プライバシーと市民的自由	2					
	CE-SPR7	コンピュータ犯罪	1					
	CE-SPR8	コンピュータにおける経済問題	2					
	CE-SPR9	哲学的枠組み	-					
	CE-SPR10	個人情報保護	-					
	CE-SPR11	内部統制	-					
	CE-SPR12	人材育成	-					
	CE-SPR13	環境問題	-					
	CE-SPR14	ハイテク製品の輸出入規制	-					
	CE-SPR15	各国のハイテク関連法規	-					

コンピュータエンジニアリング (CE: computer engineering)			コア	変更	C委員	変更	F委員	G委員
CE-SWE		ソフトウェア工学	16				<p>・システム開発スキル向上のためのカリキュラムに従来よりも多くの時間を割り当てるべきと考える。</p> <p>スキル向上には、実習・演習が重要です。BOKの上に、具体的な教育手段・教育方法を組み合わせることが必要となります。J07普及活動の中で取り組んでいくべき課題だと認識しております。</p>	
	CE-SWE0	歴史と概要	1					
	CE-SWE1	ソフトウェアプロセス	2					
	CE-SWE2	ソフトウェアの要求と仕様	2					
	CE-SWE3	ソフトウェアの設計	2					
	CE-SWE4	ソフトウェアのテストと検証	2					
	CE-SWE5	ソフトウェアの保守	2					
	CE-SWE6	ソフトウェア開発・保守ツールと環境	2					
	CE-SWE7	ソフトウェアプロジェクト管理	3					
	CE-SWE8	言語翻訳	-					
	CE-SWE9	ソフトウェアのフォールトトレランス	-					
	CE-SWE10	ソフトウェアの構成管理	-					
	CE-SWE11	ソフトウェアの標準化	-					
						+		
CE-VLS		VLSIの設計および製造	8					
CE-DSC		離散数学	23		数学系は、全領域共通基礎知識として、別分類としたほうが良いのでは。			
	CE-DSC0	歴史と概要	1					
	CE-DSC1	関数、関係、集合	6					
	CE-DSC2	数え上げの基礎	4					
	CE-DSC3	グラフとツリー	4		その分野の基礎は全領域共通であっても各分野で再掲することとしています。			
	CE-DSC4	帰納法	2					
	CE-DSC5	推論	6					
	CE-DSC6	ファジー集合	-					
CE-PRS		確率・統計	15		数学系は、全領域共通基礎知識として、別分類としたほうが良いのでは。			
	CE-PRS0	歴史と概要	1					
	CE-PRS1	離散確率	4					
	CE-PRS2	連続確率	4					
	CE-PRS3	期待値	2					
	CE-PRS4	標本分布	2		その分野の基礎は全領域共通であっても各分野で再掲することとしています。			
	CE-PRS5	推定	2					
	CE-PRS6	確率過程	-					
	CE-PRS7	仮説検定	-					
	CE-PRS8	相関関係と回帰	-					
	CE-PRS9	待ち行列理論	-					
	CE-PRS10	状態遷移モデルとマルコフチェーン	-					
	CE-PRS11	モンテカルロ法	-					

\* +:項目・コアの増大(追加)  
 -:項目・コアの削減(削除)

インフォメーションテクノロジー (IT: information technology)				A委員	C委員	F委員	H委員
ITF	IT基礎	コア	変更*	変更*	変更*		
ITF1	ITの一般的なテーマ	33	28	-5	時間数の増減については、モデルカリキュラムで必要と思われる授業時間数を増やしています。IT基礎についてはモチベーションや社会における情報の役割などを説明する必要から、時間数を考えています。	・実践的なスキル向上にウェイトを置くことは大いに賛成だが、それと共に高等教育機関ならではの基礎理論領域、ハードウェアアーキテクチャ領域等のカリキュラムも必要ではないか。	コアカリキュラムとしては基礎やアーキテクチャは最小限にとどめています。
ITF2	組織の問題	17	17				
ITF3	ITの歴史	6	6				
ITF4	IT分野(学科)とそれに関係ある分野(学科)	3	3				
ITF5	応用領域	3	3				
ITF6	IT分野における数学と統計学の活用	2	2				
HCI	ヒューマンコンピュータインタラクション	20	15	-5	IT技術者には、ネットワーク管理の知識も必須であると考え、ネットワーク管理については、情報保証(IAS)とセキュリティ、およびシステム管理とメンテナンス(SA)に含まれています。	IT技術者には、ネットワーク管理の知識も必須であると考え、ネットワーク管理については、情報保証(IAS)とセキュリティ、およびシステム管理とメンテナンス(SA)に含まれています。	
IAS	情報保証と情報セキュリティ	20	23	+3			
IM	情報管理	24	28	+4			
IPT	技術を統合するためのプログラミング	24	28	+4			
NET	ネットワーク	20	24	+4			
NET1	ネットワークの基礎	3	3				
NET2	ルーティングとスイッチング	8	8				
NET3	物理層	6	6				
NET4	セキュリティ	2	2				
NET5	アプリケーション分野	1	1				
NET6	ネットワーク管理	-	-				
PF	プログラミング基礎	38	38				
PF1	基本データ構造	10	10				
PF2	プログラミングの基本的構成要素	9	9				
PF3	オブジェクト指向プログラミング	9	9				
PF4	アルゴリズムと問題解決	6	6				
PF5	イベント駆動プログラミング	3	3				
PF6	再帰	1	1				
PT	プラットフォーム技術	14	20	6	オペレーティングシステムについては、情報分野の基礎の一つとして学んでおく必要があると考えています。また、仕組みをある程度理解しておかないと十分なシステム設計や管理が行えないと考えています。逆にコアとして大幅に増やすよりは管理と運用に時間を割いた方がよいと考えています。	オペレーティングシステムは、「SA1:オペレーティングシステムの導入と運用」を理解できる程度で良いと考える。	
PT1	オペレーティングシステム	10	10				
PT2	アーキテクチャと機構	3	3				
PT3	コンピュータインフラストラクチャ	1	1				
PT4	デプロイメントソフトウェア	-	-				
PT5	ファームウェア	-	-				
PT6	ハードウェア	-	-				
SA	システム管理とメンテナンス	11	15	4	運用マニュアル等の作成が求められるため、ドキュメンテーション技術を入れたい(入れる場所はここでないかもしれないが)	ご指摘のとおりドキュメンテーションは重要であると考えています。カリキュラムの総合演習科目で設計書などのドキュメント作成を行わせる内容が盛り込まれています。	
SA1	オペレーティングシステムの導入と運用	4	4				
SA2	アプリケーションの導入と運用	3	3				
SA3	管理作業	2	2				
SA4	管理分野	2	2				

\* +: 項目・コアの増大(追加)  
 -: 項目・コアの削減(削除)

インフォメーションテクノロジー (IT: information technology)				A委員		C委員		F委員		H委員	
SIA		コア	コア	変更*		変更*		変更*			
	システムインテグレーションとアーキテクチャ	21	20	-1			IT技術者には、特に段取り力が求められる。リスク管理を含め、プロジェクト管理に力を入れたい。		・ビジネスモデルを用いた講義時間も確保すべきと考える。		アーキテクチャに割り当てられている時間が少なくないか？(1時間でCRM、ERPの主要な機能を教え、さらにEAを教えるのは無理)
SIA1	要求仕様	6	6								
SIA2	調達/手配	4	4								
SIA3	インテグレーション	3	3								
SIA4	プロジェクト管理	3	3			+	プロジェクト管理やビジネスモデルなどは重要と考えています。BOKには含めておりませんが、モデルカリキュラムでは総合演習で扱うようにしています。		ビジネスモデルの授業・演習は重要と考えていますが、BOKのコア(必修)に含めるのではなく、各学科が独自に取り入れることが推奨されるものだと考えています。		BOKコアはあくまで最低限履修すべきものとして作られていますので、CRMやERPなどは学科独自で取り入れることが望ましいと考えております。このため、システムインテグレーションなどの授業科目や発展学習に含めてあります。
SIA5	テストと品質保証(QA)	3	3								
SIA6	組織の特性	1	1								
SIA7	アーキテクチャ	1	1								