

情報専門学科におけるカリキュラム標準 J 0 7 に対するコメント（論点別分類）

1. J 0 7 全体について

(1) 全体構成

▪ ① 各領域内容の全体を捉えた整合化の確保

a. 作成方針・方向性、構成・表記の統一化（項目のコア・非コアの基準、領域間コア時間の差）

委員名	コメント原文（ただし論点ごとに分解）
A 委員	<p>(1) カリキュラム作成方針・方向性等</p> <p>① IS は習熟度（レベル感）定義の訂正を検討しているのに対し、その他は時間数の整合性について検討しているなど、各領域でカリキュラム作成方針・方向性に対する統一性の検討が不足していると考えます。5 領域で取組方針を共通化し J07 としての整合性を確保する必要があると考えます。</p>
B 委員	<p>② ……また、資料の表現方法の不統一な点があるため、まとめ方を統一するなど、わかりやすさの向上のためには全般的な整理が望まれます。今後、プロジェクト全体の整合性や表記の統一を図るための労力が非常にかかるのではないかと思います。それを推進する組織や仕組みを検討されては如何でしょうか。</p>
B 委員	<p>③ 各学習項目がコアかコアでないかの判断を行っている基準が明確ではないように思われます。基礎知識と応用知識、社会で利用度、日本の強み、先端的か否か、一般的か専門的かなど色々な切り口があるはずなので、それを示すことがよいのではないのでしょうか。</p>
H 委員	<p>1. カリキュラム全体の統一性について</p> <p>今回発表された資料について、5 領域それぞれの独自性が十分現れていると思われれます。しかし、その反面、コンセプトや表記方法に関する共通性への配慮が不足していると感じられます。このため、「情報専門学科カリキュラム標準 J 0 7」を検討する側がそれぞれの内容を検討する際に、不要な手間が必要と感じます。ぜひ表記の統一をお願いします。</p> <p>特に、授業時間に関して、コア時間の差（240 時間～360 時間）や、大項目、詳細項目の時間提示の有無など、5 領域における統一性への配慮が必要と感じます。</p>
H 委員	<p>1. カリキュラム全体の統一性について</p> <p>……</p> <p>特に、授業時間に関して、コア時間の差（240 時間～360 時間）や、大項目、詳細項目の時間提示の有無など、5 領域における統一性への配慮が必要と感じます。</p>
H 委員	<p>2. 目標とする人材像について</p> <p>……</p> <p>人材像・学習目標・学習項目・学習時間などに関する記述を統一し、明確化することで、大学として、どの学科を採用すべきかの判断を容易にできます。また、産業界での正確な認識につながり、ひいては、採用後に個人の能力が生かされやすくなるという効果も期待できるのではないのでしょうか。</p>

b. 領域間で重複する部分の明確化、共通化または相互参照による整理

委員	コメント原文（ただし論点ごとに分解）
A 委員	<p>(1) カリキュラム作成方針・方向性等</p> <p>② 各領域の知識体系の重複部分の明確化など、領域別ではなく、全体を捉えた整理が必要と考えます。</p> <p>情報システムの専門家(ITSS 等)として領域間で共有する知識項目を明確化した上で、領域毎の特徴を際立たせて重複を減らすことが考えられます。例えば、SE における「cmp.cf.11 データベースの基礎」、IS における「1.3 データベース」、IT における「IM 情報管理」などは、共有化あるいは共通化することが望ましいと考えます。</p> <p>一方で、各領域における技術者の育成目標の達成のためには、重複が避けられないこともあり、具体的には、CS と CE、IS と SE と IT の 2 つの領域間で共有する項目を定め、それぞれの上に各領域の専門知識を積み上げるといった整理が可能と考えます。</p>
B 委員	<p>④ 各セクション（委員会）間で重複したカテゴリーが一部見受けられます。自領域で習得すべき知識と他領域を参照すればよい知識とに分けて整理する構成ははいかがでしょうか。・・・</p>
C 委員	<p>1.・・・また、各領域に共通／固有の知識・スキルを整理・体系化したい。</p>
C 委員	<p>1. 全体を捉えにくい。自明かもしれないが、各領域の定義と位置づけを明確に言葉で定義した方が良い。・・・</p>
E 委員	<p>(1) 総合</p> <p>④複数領域に重複しているカリキュラムを整合化し、日本の情報工学教育に関与する方々の情報共有を図ることが必要である。</p>
E 委員	<p>(2) 構成</p> <p>①基礎部分、共通部分のカリキュラムを設置した上で、5つの領域の専門コースを位置付ける構成にすべきである。</p> <p>その場合、共通部分と各領域の専門部分の整理と強調点の明示が必要である。</p> <p>・・・</p>
E 委員	<p>(2) 構成</p> <p>・・・</p> <p>基礎／共通部分の教材は、情報工学関係学科以外の理系学科および文系学生にも活用できるのではないか。</p>
F 委員	<p>・ カリキュラムの重複</p> <p>5領域間で重複しているカリキュラムが多数存在するので、各分野間で連携を取るべきと考える。</p> <p>また、それらは情報専門学科の必須内容であると思われるので、抽出し、「共通科目」として専門教育前に一括して教育をしたほうが効率的と思われる。</p>
G 委員	<p>①CS、IS、SE、CE、IT の領域毎に知識体系が検討されているが、領域毎に考え方が異なっており、領域間に重複が多い。用語の統一を図り、共通部分については粒度に凸凹がないように調整した上で、それぞれの領域の特徴的な部分が明らかになるよう整理していただきたい。また、整理のしかたについては統一していただきたい。</p>
G 委員	<p>③5つの領域毎に知識体系を定義する場合は、下図のように領域間の知識項目の関連をわかりやすく説明していただきたい。</p>

委員	コメント原文 (ただし論点ごとに分解)
G 委員	②情報通信技術(IT)の各領域(CS、CE、IS、SE、IT、IS)毎に J07 カリキュラム標準を定義することは有意義であるが、各領域を統合したリファレンスモデルがあると適用し易いのではないかと考える。学部学生は、まだ IT のどの分野に進むが決まらないことがほとんどであり、ある領域に特化するということではなく、すべての領域を統合したカリキュラムの方が利用価値が高い。
D 委員	<p>(1) 5領域の分類について</p> <p>日本経団連が、従来から提起してきた課題の一つが、(無論 CS 分野の研究の重要性は認識した上で)日本の情報系教育が CS 分野に偏りすぎている点です。今回、J07 において、CS、IS、SE、CE、IT の 5 つの領域の区別が提示されたことは、社会が求める情報系教育の多様性を広く認識させ、わが国の情報教育を考える上での鳥瞰図として非常に有効といえます。</p> <p>一方で、説明会において、それら 5 つの領域が、個別の学科設立にも繋がりととの可能性が言及されておりました。学部時代から、専門性の高い教育を行うことに対しては賛同いたしますが、情報系学科学士の総数が限られること、そして学生が進むキャリアの多様性確保を考慮すれば、5 分野個別の専門分化、高度化を追及するよりも、共通領域の整理を行っていくことが有用と考えます。</p> <p>それにあたっては、主に科学的な基礎や理論を礎とする CS と CE、コンピュータの活用を基盤とする SE と IS と IT、という区別のもとでの整理が可能ではないかと考えます。</p> <p>説明会においても、5 領域の相互参照については十分な時間を割けておらず、本元の CC2005 でも整合性は不十分とのコメントがありました。容易な作業ではありませんが、今後、整理を強く希望いたします。</p>
H 委員	<p>3. カリキュラム内容について</p> <p>5 領域に渡り非常に広範囲なエリアをカバーできていると思われれます。しかし、一般の大学であれば、一つだけの領域の学科ではなく、複数の領域を組み合わせることが考えられますが、それを教える情報系教員の人材の課題が考えられます。また、各領域が独自体系であるため、組み合わせ自体が大変困難であることも予想されます。</p> <p>産業界からの基礎知識向上のニーズも踏まえ、情報技術者全般に必要とされる共通の基礎部分については、5 領域で学習内容を統一し、それぞれの領域に特化した専門科目や選択科目を抽出して各領域の知識体系を整理することで、基礎レベルの向上とともに、大学側のカリキュラムの組み合わせ選択が容易になることが期待できます。</p> <p>また、実際の教育現場での 5 つのカリキュラムの捉え方や組み合わせ方などについても適切な指導が必要と考えられます。</p>

c. その他の表記方法・構成についての意見

委員	コメント原文 (ただし論点ごとに分解)
A 委員	<p>(3) 教育効果の問題</p> <p>③ 今後の取り組みにおいて、それぞれの学習項目に CC2005 のようにトピックスと学習の目標を明示することが望まれます。</p>
C 委員	2. BOK として範囲は分かるが深さがイメージできない。時間数でなく、到達レベルで表現できないか。

委員	コメント原文（ただし論点ごとに分解）
E 委員	<p>(2) 構成</p> <p>②企業における関連人材像と彼らが担う業務およびキャリアパス等と本カリキュラムとの関係を図示し、各単元、項目の意味を明示する必要がある。</p> <p style="text-align: center;">大学教育では、共通部分と専門部分の整理、 企業に入ってからの流れ、キャリアパスの見える化を</p>
E 委員	<p>(3) 内容</p> <p>②システムのライフサイクルを意識し、今、学んでいる内容が、システム全体にどのように機能したり影響したりするかを明示できるようにした方がよい。</p>
E 委員	<p>(3) 内容</p> <p>③上流工程部分の項目については、粒度を揃え、不足項目を整理して、流れを把握できるように設定した方がよい。</p>
E 委員	<p>(4) 学習方法</p> <p>②項目と講義時間数のバランスだけでなく、演習や実技も組み合わせ、より理解が深まるような実施方法も提示した方がよい。</p>

▪ ② 育成目標となる人材像の提示

委員	コメント原文（ただし論点ごとに分解）
C 委員	3. 教育のゴールが不明確である。例えば、ねらう人材像と求めるスキル・知識を明示し、これに対する BOK の充足度合いを定義してみたいか。
E 委員	<p>(1) 総合</p> <p>③各カリキュラムを受講（修了）すると、どんな人材が育成されるのかをイメージできるような、具体的な目的や成果に関する記述があるとよい。</p>
F 委員	<p>・ J O 7 と育成すべき人材像</p> <p>J O 7 は、国家及び産学として育成すべき I T 人材像（質・量）を実現するためのカリキュラム標準となることを期待。グローバルスタンダードに加え、日本の特性・目標を付加したものが望ましいと考える。</p>

委員	コメント原文 (ただし論点ごとに分解)
H 委員	<p>2. 目標とする人材像について</p> <p>今回は5つの分野でのカリキュラム策定が行われていますが、それぞれの分野における期待される人材像が不明瞭です。その学科を卒業した人物がどのように社会に期待される人物であるのかについてももう少し具体的な検討と明示が必要と感じます。</p> <p>人材像・学習目標・学習項目・学習時間などに関する記述を統一し、明確化することで、大学として、どの学科を採用すべきかの判断を容易にできます。また、産業界での正確な認識につながり、ひいては、採用後に個人の能力が生かされやすくなるという効果も期待できるのではないのでしょうか。</p>

(2) カリキュラムに設定・追加・対応付けすべき項目

▪ ① カリキュラム策定方針の設定

委員	コメント原文 (ただし論点ごとに分解)
C 委員	4. まずは基礎知識・スキルを習得させることがねらいだとは思いますが、その次のステップについての方向性が感じられない。世界と対等に渡り合えるエンジニアを育成するには、例えば、技術的な展望を示すことも重要ではないか。
E 委員	<p>(1) 総合</p> <p>① I T の分野で世界をリードすることを前提にした明確な目標設定をすべきである。</p> <p>(C C の翻訳では C C の域を出ない。目標達成に向け、我が国流の差別化を施したカリキュラム編成が必要である。)</p>
E 委員	<p>(1) 総合</p> <p>② 産業界における I T 関連人材 (システムベンダ / ユーザーを問わず) の現状分布と我が国が目指す I T 関連人材ポートフォリオを描き、そのギャップを埋めるためのカリキュラムとして編成すべきである。</p> <p style="text-align: center;">IT人材構成 ToBe を示す</p> <p>The diagram illustrates the current state (AsIs) and the target state (ToBe) of IT talent. On the AsIs side, there are three pyramids representing talent levels: Information System Graduates (情報系卒業生), IT Technicians (AsIs) (IT技術者(AsIs)), and IT Upper Level Technicians (IT上級技術者) with 158,000 people. On the ToBe side, there are five circles representing target roles: IT Strategist (ITストラテジスト系人材) 000,000, System Architect (システムアーキテクト) 000,000, Project Manager (プロジェクトマネージャー) 000,000, Security Specialist (セキュリティスペシャリスト) 000,000, and Embedded System Technicians (組み込み系技術者) 000,000. A thought bubble asks 'What is the future image?' (将来像は?).</p>

▪ ② 非情報系スキルやヒューマンスキル等の取り込み検討

委員	コメント原文（ただし論点ごとに分解）
A 委員	(1) カリキュラム作成方針・方向性等 ③ コミュニケーション能力等は本来初等中等教育において養成されるべき項目であります。これらの項目を大学教育に求めることは、本来の目的である専門教育の質・量を低下させることに繋がることを認識して、検討を継続する必要がありますと考えます。
A 委員	(1) カリキュラム作成方針・方向性等 ④ 組込システムのように日本に優位性のある分野に関しては、各領域に独自の項目を追加するよう総合的な検討が必要と考えます。
E 委員	(3) 内容 ①ヒューマン・スキル系（セルフコントロール、対人折衝、コミュニケーションなど）のカリキュラムを盛り込むべきである。
E 委員	(3) 内容 ④考え方、メソッド、ツールにも目を向けた項目をカリキュラムに組み込むとよい。
E 委員	(3) 内容 ⑤事例研究や業界の構造、ビジネスの仕組み、ITの歴史なども採り入れて、学生が目指すべき姿を描けるようにするとよい。
E 委員	(3) 内容 ⑥経営やビジネス分野の項目も盛り込むとよい。
E 委員	(3) 内容 ⑦組み込むべきコンテンツ 運用、保守、品質管理、システム要求把握、活用技術、ロジックの構築力 など。
E 委員	(6) ユーザー企業/JUASとしての支援 ①ビジネスモデルのケースを提供するので、事例研究を行い、新しいビジネスモデル創出能力を身に付けさせる方法も考えて欲しい。
E 委員	(7) その他 ③情報系業務に従事するためには、関連の法律や経営等の知識も必要なので、情報工学系の学生に、法学部や経営学部等の関連単位の修得も課す仕組みも検討すべきである。
G 委員	その他①若干、専門知識の詰め込み偏重になっている印象を受ける。開発現場では、自身の考えを他者へ適格に説明するための論理力やプレゼンテーションスキル、コミュニケーションスキル、あるいは事象をモデル化する等の抽象化スキルを備えた人材が不足していると感じている。専門知識のみならず、これらのスキルを養うようなカリキュラムを取り込んでいただきたい。
G 委員	その他⑨現状では、情報系出身の卒業生総数が需要に対して不足しており、企業側も「夢」のある職場を提案する必要性を感じているが、大学においても、「情報系への関心」とともに「人間力」「チーム力」の育成にも力を入れていただきたい。

委員	コメント原文（ただし論点ごとに分解）
D 委員	<p>(4) 大学教育としての観点</p> <p>・・・</p> <p>繰り返しになりますが、BOKの策定から着手したことが、システム開発のリアリティを伴わない学生への知識中心の教育に繋がってしまわないことを切望します。加えて、こうした高い専門性を駆使し社会にインパクトを与えられる人材となるには、論理的思考やコミュニケーション力といった技術者としての基礎的な能力養成も不可欠です。BOKに表現するのは難しいですが、学習方法の工夫によって、相当のレベルアップが可能であることは経団連の支援コースでも実感している所ですので、J07を使った指導上のアドバイスや指針等を別途策定していくことが必要です。</p>

③ 他制度との関係の明示

委員	コメント原文（ただし論点ごとに分解）
A 委員	<p>(2) ITスキル標準 (ITSS) との関係整理</p> <p>①単位取得とは別に業界標準 (共通) の技術レベルの達成度を客観的に判断できるように、在学中に業界標準の技術資格の取得を目標に加えることが考えられます。</p> <p>その意味で、ITSS レベル 1 に相当する IT パスポート試験は、学生のアウトカム評価の指標の 1 つになり得ると考えます。</p> <p>各領域の技術分野と ITSS の技術分野 (あるいは IT パスポート試験の出題範囲) と同じである必要はありませんが、双方のどの技術分野が相対するか、もしくはカバーするのか、分かりやすくマップなどを作り表示することが、産学の連携を深める上で必要と考えます。</p>
E 委員	<p style="text-align: center;">共通知識エリアマップの作成を</p> <p>(2) 構成</p> <p>③共通キャリアスキルフレームワーク、IT関連のスキル標準（情報処理試験、ITSS、ETSS、UISS）と本カリキュラムが、それぞれカバーする知識エリアの関係をマッピングし、整理体系化すべきである。</p>

委員	コメント原文 (ただし論点ごとに分解)
E 委員	(7) その他 ①コースの開発については、基礎／共通部分は国等の公的機関が、専門領域は大学等の各実施機関で、といった役割分担を行い、効率的に開発することも検討すべきである。
E 委員	(7) その他 ② I T 関連業務に従事する人材を増員するためには、情報工学系学科以外の学生も、必要単位数を修得すれば情報工学系の修了資格を得られる、ダブルメジャー制度の導入を検討すべきである。
F 委員	・ 新情報処理技術者試験との整合性 I S、S E、I T の各領域のカリキュラムと新情報処理技術者試験との間に整合性があるのが望ましいと考える。
G 委員	その他⑤履修すると、新情報処理技術者試験の I T パスポート試験などが試験免除となるという事が考えられても良い。
H 委員	4. 共通キャリアフレームワークとの関連性について 今回の「情報専門学科カリキュラム標準 J 0 7」は経済産業省の人材育成ワーキンググループの検討課題と関係していることを再確認する必要があります。ワーキンググループの検討の中で、3つの人材像と7つの人材類型が、今後の我が国が目指すべき高度 I T 人材として類型化がされました。その人材像および人材類型と「情報専門学科カリキュラム標準 J 0 7」とは関連しています。例えば、基本戦略系人材のストラテジスト育成には C C - I S カリキュラムでの学習が望まれるとされています。しかし、今回の発表資料にはその人材育成ワーキンググループの報告書との関連が明記されていません。人材育成の施策は重要であり、経済産業省も文部科学省も連携した施策の展開が重要と考えるので、その施策の共通部分を明確に示すべきと考えます。
G 委員	その他④本標準を策定するにあたって、アメリカのカリキュラム標準を参考にされているとのことであったが、アジア(特に中国、韓国、東南アジア諸国等)とのベンチマークも併せて行っていただきたい。日本と欧米には文化や習慣、気質、考え方に違いがあり、アメリカのカリキュラムを踏襲することが日本にとって一概に良いとはいえない面もあるように思える一方で、日本にこれらの傾向が比較的近いアジア諸国と比較することは意味があるものと思う。また、成長著しいアジア諸国を相手に、日本が今後競争してゆけるかを考える上でも重要ではないかと思う。

▪ **④ 関連カリキュラムの策定**

委員	コメント原文 (ただし論点ごとに分解)
A 委員	(3) 教育効果の問題 ①大学が、各領域において必要十分な教員を確保することは困難と考えます。その場合、各大学において5つの領域から項目を選択して配置することが想定されますが、教育効果について J07 との関係が不明確になることが想定されま す。

委員	コメント原文 (ただし論点ごとに分解)
G 委員	<p>その他⑦ITはIT業界だけでなく、金融、公共、製造、流通、行政、社会システムなどあらゆる業界の競争力強化のための必須な技術になりつつあり、また、これらの各業界のIT技術者として就職する人材は必ずしもIT専門学科の卒業生だけではない。今やIT教育は、情報系以外にもあらゆる学科に必要であり、情報系以外の理工系学生や、文科系学生にもITを教育する必要があると考える。情報系以外の理工系、文科系学科を想定しJ07のサブセットとしてカリキュラムモデルを作成していただきたい。</p>
D 委員	<p>(2) ISとITに対する認識</p> <p>ISおよびITは、対象となる組織や情報システムの課題と密接であり、一段階高度な教育(たとえば、大学院教育や社会人教育)として再整理すべきではないという一部意見がございました。内部での完全な合意には至っておりませんが、参考としてコメント提出させていただきます。</p> <p>「例えば、現在、ISやITの特徴的な部分は入社後3~5年の実務経験の後に担当している状況です。学部教育では、この入社後3~5年の実務経験に相当する部分を、如何に効率よく体系的に学ぶかという点に着目していただきたいと考えます。学習時間も限られておりますので、細分化して即戦力としての能力を高めるよりも、多様な実務に対応しうるに最低限必要な“共通的な知識・能力”を有していることの方が、企業・学生の双方にとって有用ではないかと考えます。」(一部の意見)</p>

(3) カリキュラム実施上の課題

■ ① 教育実施方法・スキームの検討

委員	コメント原文 (ただし論点ごとに分解)
A 委員	<p>(3) 教育効果の問題</p> <p>②大学院まで含めて、教育効果を高める観点から、大学において達成しにくい実務能力について、学習目標とレベルが設定されることを条件に長期間(半年~1年間)のインターンシップを実施し、単位認定することが考えられます。</p>
B 委員	<p>① 実社会で実際に適用されている情報処理業務について、いくつかの代表的な応用例をケーススタディとして取り上げ、次のような帰納法的に情報処理理論を理解するセクションを冒頭に置くことで、実務に興味をもって学習することができると思われます。</p> <p>実際の適用事例(例:インターネットを利用したチケット注文など)</p> <p style="text-align: center;">↓</p> <p>利用されている情報技術(例:DBMS(Database Management System)、ネットワークなど)</p> <p style="text-align: center;">↓</p> <p>その情報技術を実現している要素技術や基礎理論(例:アルゴリズム、排他制御など)</p>
E 委員	<p>(4) 学習方法</p> <p>③インターンシップは必須である。</p>
G 委員	<p>その他⑥教員不足が懸念されるので、企業での実習であるインターンシップ、大学と企業での教育を組み合わせたコーオペ教育も考慮し、カリキュラム例が示されるのが良い。</p>

委員	コメント原文 (ただし論点ごとに分解)
D 委員	<p>(3) 日本としての独自性の考慮</p> <p>例えば、CE 領域について、わが国の組み込みソフトの強みを一層強化していくとの観点から、データベースやネットワークなどが含まれている点に強く賛同いたします。電気・電子学科の領域にも及ぶ組み込み技術者育成の教育体系の整備が、東海大学大学院の組み込み専門職大学院の設立など、他国にも類を見ない取組みにより着実に進展している点については、産業界としても非常に心強く感じると共に、産学の連携をより一層深めていくべきと考えます。</p> <p>一方、日本独自の制約条件として、学習時間総数（教養1年、卒論1年とすると専門教育は2年のみ）が述べられておりました。設置認可に関係した制約があるのは承知の上ですが、海外と競争するには、専門教育の時間を増やす工夫をしていくことが必要です。例えば、卒業認定を研究論文に限定せず、グループによるソフトウェア開発や実証の成果で代替することにより、より多くの学習時間を確保することも推奨されうると考えます。</p>
D 委員	<p>(4) 大学教育としての観点</p> <p>実践的なスキルを教えることは企業が得意するところですが、大学教育の特徴として、例えばSEやCEの分野にも、工学的基本概念（統計、数学など）が含まれている点に強く賛同します。ただし、「原理・原則を教える」という点に関し、特に工学としての未成熟なソフトウェア開発などの部分においては、必要と思われる知識を座学で詰め込むというアプローチではなく、実践を通じて原理原則を体得するという観点で、PBLやインターンシップ、企業との共同プロジェクトなどの教育手法の検討も併せて検討していくことが不可欠と考えます（J07をベースにカリキュラムを編成する大学の裁量や差別化の範疇かもしれませんが）。</p> <p>繰り返しになりますが、BOKの策定から着手したことが、システム開発のリアリティを伴わない学生への知識中心の教育に繋がってしまわないことを切望します。加えて、こうした高い専門性を駆使し社会にインパクトを与えられる人材となるには、論理的思考やコミュニケーション力といった技術者としての基礎的な能力養成も不可欠です。BOKに表現するのは難しいですが、学習方法の工夫によって、相当のレベルアップが可能であることは経団連の支援コースでも実感している所ですので、J07を使った指導上のアドバイスや指針等を別途策定していくことが必要です。</p>

▪ ② 教育実施側の充実

委員	コメント原文 (ただし論点ごとに分解)
E 委員	<p>(5) J07カリキュラムを教えるために求められる要件</p> <p>①カリキュラム作成と並行して、インストラクタの育成、充実等、教える側の人材を質量ともに向上させる仕組みも検討する必要がある。</p>
E 委員	<p>(7) その他</p> <p>④短い時間内で効果を上げるためには、何のためにこれらの項目を勉強するのか、その必要性や重要性を明示したり、利用実態を認識させ、モチベーションを喚起する内容を盛り込む必要がある。</p>

委員	コメント原文（ただし論点ごとに分解）
D 委員	<p>(5) 実際の教育現場への反映</p> <p>・・・</p> <p>また、現在の情報系学科教育の中心である CS 領域のご説明では、“現在大学に在籍する教員の方々への配慮“を大変気にされている様子でした。使われるカリキュラムとしていくにあたり無視できない点かもしれませんが、わが国の将来を担う研究者に求められる知識・能力という観点からのトップダウン的な整理を切に要望します。</p>

▪ **③ 学習のためのモチベーション喚起対策**

委員	コメント原文（ただし論点ごとに分解）
G 委員	<p>その他②専門知識の項目自体は、概ね事前に学生に身につけておいてもらいたい知識が網羅されているように思える。一方で、受講する学生に対してそれぞれの科目を学習する目的・意義を理解してもらえないか不安に感じる。学生に対していかに動機付けをしてゆくかが今後の課題である。</p>

▪ **④ 教育現場への適用推進**

委員	コメント原文（ただし論点ごとに分解）
G 委員	<p>その他③上記と併せて、これらの標準に則ったカリキュラムを設定する大学をいかに増やしてゆくかがカギになると思う。</p>
D 委員	<p>(5) 実際の教育現場への反映</p> <p>大学から産業界に呈される課題の一つとして、大学での専門教育を考慮しない一律採用があげられます。日本の IT が世界に伍する実力をつけていくうえで、大学での情報教育を世界水準に上げていく必要があります、J07 の最終的な成果が実際の大学での現場での教育に着実に反映され、厳密な成績管理もあわせて、学生が社会にでる際に自らの能力を客観的に示す指標として提示することが出来れば、現在の一律採用の改善に大きく寄与するものと考えられます。</p> <p>換言すれば、J07 の教育現場への適用がすすまなければ、今回の皆様の労力は無駄なものとなってしまいます。10 年前に策定された J97 においても、その適用は十分でなかったと認識しておりますので、J97 の活用実態を踏まえた課題を分析しその方策を講じることが、J07 を真の有効なアセットとしていく上で不可欠な事であり、情報処理学会からの積極的な働きかけに期待します。</p> <p>・・・</p>

▪ **⑤ 修得時間の不足への対応**

委員	コメント原文（ただし論点ごとに分解）
A 委員	<p>(3) 教育効果の問題</p> <p>④ 幅広く学ばせるという意図は理解できますが、項目が膨大で、かつ、それぞれの項目に係る時間が短すぎるため、現在の大学教育の時間内でカバーしきれぬのか、学生がどこまで理解を深めることが可能かといった実施面での課題が残ると思われます。</p>
E 委員	<p>(4) 学習方法</p> <p>①大学の期間中だけでなく、大学院も含めたカリキュラムとしても検討するとよい。</p>

委員	コメント原文（ただし論点ごとに分解）
F 委員	<ul style="list-style-type: none"> 講義時間 現在の日本の講義時間に合わせてカリキュラム時間を割り当てるのではなく、柔軟に講義時間を決めてはどうか。どうしても制約があるのなら、医学部のように6年間で育成する、即ち大学院と一体となったカリキュラムも検討すべきではないか。

▪ **⑥ プログラミングスキルの充実**

委員	コメント原文（ただし論点ごとに分解）
G 委員	<p>その他⑧特に、理工系学生にとっては「プログラミング」は「文章作成」に近い教養的なスキルになりつつあり、全ての理工系卒業生が、簡単なプログラムなら書ける/実用的なプログラムが読めるレベルを身につけて卒業できることが望ましい。</p>

▪ **⑦ 産側からの支援**

委員	コメント原文（ただし論点ごとに分解）
E 委員	<p>(5) J O 7カリキュラムを教えるために求められる要件 ②本カリキュラムの講義を担当するには、教える内容が、どんな人材にどのように活用されるかを知っている必要がある。 そのために、産業界側は、インストラクタを受け入れるインターンシップや情報提供の仕組みを検討する必要がある。</p>
E 委員	<p>(1) 総合 ⑤産業界からも、中長期的な人材像とその育成についての情報を提示し、本カリキュラムがどこに位置付くかを明示できるようにしなければならない。</p>
E 委員	<p>(6) ユーザー企業/JUASとしての支援 ②インターンシップ、講師派遣、共同研究、実務実習(仕事として)の機会提供。・・・</p>
E 委員	<p>(6) ユーザー企業/JUASとしての支援 ②・・・修得内容による採用時の待遇の検討。 など</p>
E 委員	<p>(6) ユーザー企業/JUASとしての支援 ③インターンシップは、短期間ではなく、極力、全体感が分かるような立場で受け入れることを検討する。</p>
E 委員	<p>(6) ユーザー企業/JUASとしての支援 ①ビジネスモデルのケースを提供するので、事例研究を行い、新しいビジネスモデル創出能力を身に付けさせる方法も考えて欲しい。</p>

(4) その他

委員	コメント原文 (ただし論点ごとに分解)
H委員	<p>5. 今回のコメントの反映について</p> <p>日本のIT人材育成の方向性を明示する「情報専門学科カリキュラム標準J07」は、産業界としても、大変重要なものです。その上で、今回挙げさせていただきました各コメント内容は「情報専門学科カリキュラム標準J07」を実践的で有効性のある施策とするためには、ぜひ考慮していただきたいポイントであると考えます。</p> <p>作成スケジュール等の課題はあると思われませんが、可能な限り早期にこれらのコメントを吟味していただき、「情報専門学科カリキュラム標準J07」に反映させていただきたくよろしくお願い申し上げます。</p>

2. 各領域について

(1) CS (コンピュータ科学) について

① 賛同する点

委員名	コメント原文 (ただし論点ごとに分解)
C 委員	CS として既存の知識は十分に網羅している。
H 委員	情報を学ぶ上での基礎となる分野として、项目的には十分に網羅されている。

② 重視すべき (不足している) 内容

委員名	コメント原文 (ただし論点ごとに分解)
C 委員	サイエンスであるからには、新しい手法考案を感じさせる知識としての展望・方向性を示すべき。

③ 実施方法について (時間)

委員名	コメント原文 (ただし論点ごとに分解)
B 委員	「情報管理」の章について、一般的な事例をベースに配分時間を増やしてはいかがでしょうか。この章で取り上げられている技術が、実社会で適用されている情報処理業務との接点であると考えられるためです。
F 委員	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。
H 委員	「IM: 情報管理」と「SE: ソフトウェア工学」のユニットのコア時間をもう少し増やすと、より実践的な内容になるのではないかと。
H 委員	問題解決の手順を自ら組み立てられる能力を身に付けるため、「PF: プログラミングの基礎」や「AL: アルゴリズムの基礎」の科目では、演習時間を多く取り入れてほしい。

④ カリキュラム時間

委員名	コメント原文 (ただし論点ごとに分解)
H 委員	コア時間の合計が 240 時間というのは短いのではないかと。

(2) IS (情報システム) について

① 賛同する点

委員名	コメント原文 (ただし論点ごとに分解)
F 委員	産業界として期待している領域。
F 委員	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する。
H 委員	目標とするレベルが設定されていたので、分かりやすいものでした。

② カリキュラムの基本方針

委員名	コメント原文（ただし論点ごとに分解）
A 委員	主にユーザ企業の IT 部門要員であり、そのトップは CIO であることをイメージしている。中長期的なビジョンをもち、業務を含めた情報システムの全体像を描く立場で活躍できるよう、ビジネスに精通し、経営層とも会話できる情報専門家としての位置づけを明確にした上で、カリキュラムを整備する必要がある。
D 委員	対象となる組織や情報システムの課題と密接であることから、一段階高度な教育（たとえば、大学院教育や社会人教育）として再整理すべき。（一部の意見）
D 委員	現在、IS や IT の特徴的な部分は入社後 3~5 年の実務経験の後に担当しており、学部教育では、この入社後 3~5 年の実務経験に相当する部分を、如何に効率よく体系的に学ぶかという点に着目していただきたい。
D 委員	学習時間が限られているので、細分化して即戦力としての能力を高めるよりも、多様な実務に対応しうるに最低限必要な“共通的な知識・能力”を有していることの方が、企業・学生の双方にとって有用ではないか。

③ 重視すべき（不足している）内容

委員名	コメント原文（ただし論点ごとに分解）
A 委員	フーリエ変換や遺伝的アルゴリズムなどの数学的手法、画像処理や図形処理、言語処理の基本的アルゴリズムなども科目に加えることが、育成上有益。
B 委員	情報システムを利用する側の観点からの項目がやや不十分なように見受けられます。具体的には、要件定義、機能を実現する方式や技術の決定、システム利用・運用、などの面や、内部統制や監査の観点などが必要となってきます。
C 委員	情報システムには幅広い分野の知識が必要であることは理解できるが、広く浅いという印象。重点化するポイントを提示すべき。
E 委員	「IS の法的、倫理的側面」については、今後、コンプライアンスが重視されるのでより強化すべき。
E 委員	「基本的な組織の機能」については、企業活動の内容が理解できないとシステムの設計はできないのでより強化すべき。

④ 構成（フォーマット等）

委員名	コメント原文（ただし論点ごとに分解）
E 委員	大項目の数値が対応する中項目の数値より高いものがある。（例：IS3.8“情報とビジネス”は 5 だが 3.8.1~3.8.3 は 1）
H 委員	他の領域に合わせた記述を希望。
H 委員	他の領域と比較すると、中項目、小項目のレベルが揃っていない。

⑤ 構成（時間）

委員名	コメント原文（ただし論点ごとに分解）
H 委員	シラバス作成の指標として学習時間は必要。
H 委員	学習時間が不明確な場合、他領域との組み合わせが難しくなる。

⑥ レベル設定

委員名	コメント原文（ただし論点ごとに分解）
H 委員	学部学生に対して 5(実社会の問題に適應する)の技能を求めることは妥当か。

(3) SE (ソフトウェアエンジニアリング) について

① 賛同する点

委員名	コメント原文 (ただし論点ごとに分解)
F 委員	産業界として期待している領域。
F 委員	「開発技術を手厚く教える」という姿勢は賛成。
F 委員	「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。
G 委員	基礎工学の習熟に重点を置いたカリキュラム方針に対しても共感する。
D 委員	大学教育の特徴として、「ソフトウェア開発にとどまらない一般工学原則を知らなくてはならない」という点に強く賛同。
D 委員	「ボトムアップ的にライフサイクルを学んでいく年次進行案」、「学部教育として、分析や設計の開発技術を手厚く教えるバランスになっている」、「データ構造とアルゴリズム・プログラミング言語基礎は、アルゴリズム論やプログラミング言語論ではなく、ソフトウェア開発に比重を置いた配分となっている」についても、賛同。
H 委員	SE 教育の中心をソフトウェア開発一般に置くのであれば、必要なテーマは網羅している。

② 基本方針

委員名	コメント原文 (ただし論点ごとに分解)
H 委員	複数の領域の学科を有する大学でカリキュラムを作成する場合に、SE 学科で講義すべき内容として、BOK が自己完結していることは問題ではないか。項目とコアという体系にすれば、共通科目と選択科目というように編成しやすくなる。

③ 他領域との重複について

委員名	コメント原文 (ただし論点ごとに分解)
B 委員	各カテゴリーの学習時間数が示されていますが、それらの適切性までは確認しきれませんでした。この「SE ソフトウェアエンジニアリング (software engineering)」については、前の「IS 情報システム (information systems)」との重複感がある項目が最も多い部分です。
C 委員	精力的な BOK という印象だが、他領域との重複が散見されることから、領域共通／固有を整理すべき。

④ 重視すべき（不足している）内容

委員名	コメント原文（ただし論点ごとに分解）
A 委員	「ソフトウェア V&V」や「ソフトウェア開発マネジメント」を CC の半分以下とすることで日本版の特徴を出そうとしているのかもしれないが、特にこの2つのエリアは、できるだけ深く理解することが望ましいと考える。
A 委員	ある程度焦点を絞った方が実践的と考えます。例えば、項目一覧表において「確率・統計」、「離散数学」、「論理と推論・計算理論」は CS 領域に任せ、SE 領域では「コンピュータとソフトウェア工学基礎」以降の項目に焦点を絞るといった整理が考えられる。
A 委員	「ソフトウェアモデリングと要求開発」において、クラス図やコンポーネント図がキーワードとして挙げられているが、UML に関してはそれ以外の図もソフトウェア開発の現場に置いて非常に重要な役割を果たすことから、UML とそれを活用するためのモデリング技法を併せて体系的に教育する必要がある。
A 委員	ISO15408 に代表されるように、品質の高い仕様を策定する上で形式手法の重要性が増していることから、仕様策定に係わる項目の充実を図るべき。
E 委員	実社会の大規模システムでは IS や IT が要求分析や概要設計までを担当し、SE が詳細設計以降を担当する場合もあることを考えると V&V(設計時の障害対応や性能の検討など)やプロジェクト管理を重視すべき。
H 委員	産業界における業務系のアプリケーション開発の立場からすると、データベースに関する基礎教育をもう少し充実させないと、即戦力とすることは難しい。
H 委員	覚えたものを応用し、実践していくためには、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。離散数学など、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機会を増やしていけば、間違いや漏れの少ないプログラミングの考え方を身に付けることができるようになるので、プログラミング能力と品質が向上すると考える。

⑤ 教育順序

委員名	コメント原文（ただし論点ごとに分解）
A 委員	主に SIer に席を置くエンジニア（大工ではなく建築士、業界では IT コンサルタント等）を育てる課程であるとする、トップダウン型が良い。
G 委員	ボトムアップ的（実装→設計→分析）な順で教育カリキュラムを進める方針は良い。設計は実装経験なしでは理解できないものであり、分析は設計・実装経験なしでは理解できないものなので、上記順でのカリキュラム体系は妥当。

⑥ 構成（時間）

委員名	コメント原文（ただし論点ごとに分解）
H 委員	360 時間というコア時間は、少し長い。他の領域のコア時間は 300 時間以内となっているので、学生の学科間の移動が難しくなる。

⑦ 用語

委員名	コメント原文（ただし論点ごとに分解）
A 委員	用語：要求開発という用語は不適切。CC の翻訳でも要求獲得、要求分析、要求定義です。IS の領域でなら開発もあり得ますが、SE では獲得と分析と定義を使い分ければ済む。

⑧ レベル設定

委員名	コメント原文（ただし論点ごとに分解）
A 委員	教員の得手／不得手、興味のある／なしに依存することを避けるため、IS のように標準的な重み付け（学習の最低レベル）を設けることを検討する必要がある。

⑨ 実施方法について

委員名	コメント原文（ただし論点ごとに分解）
F 委員	ビジネスモデルを用いた講義時間も確保すべき。
G 委員	グループによる模擬開発をカリキュラムに追加するのが良いのではないかと。学部生の段階ではプロジェクト・マネジメントの実践までは難しいかもしれないが、例えばメトリクスの計測とその分析等、実践的なカリキュラムの中から習得できるものも多いと思う。
D 委員	「コアとして分析・設計を重視し、コーディング、品質、マネジメントの配分を軽くした」という表現には賛同しかねるが、知識項目に対して時間配分をするアプローチによる弊害ではないかと考える。現在セットで検討されている演習や PBL の中で、ものづくりに不可分なものとして品質を学ぶような工夫をされていくことを要望。
H 委員	PBL は全体工程を把握する意味でも非常に効果的であり、開発の面白さを実感できる。PBL を含まない場合、現在の SE の学習内容だけでは興味を感じられる要素が少なく、学生が情報嫌いになってしまうのではないかと。

（４）CE（コンピュータエンジニアリング）について

① 賛同する点

委員名	コメント原文（ただし論点ごとに分解）
B 委員	組込みシステム設計を1つの重点項目にしたことはよい点であると思います。現在は量販製品への搭載システムが日常的に多く使われているため、この分野も情報処理技術応用の重要な要素であると考えられるからです。
C 委員	知識の範囲としては問題ない。
H 委員	組込み系の学習要素が体系的に整理されており、大学以外でも参考になる内容。

② 他領域との重複について

委員名	コメント原文（ただし論点ごとに分解）
A 委員	知識体系に、本来は電気・電子学科の領域であるハードウェア製造に携わるもの（電気回路、信号等）が含まれており、J07 以外の工学分野との関係についても、関係整理が必要な場合がある。

③ 重視すべき（不足している）内容

委員名	コメント原文（ただし論点ごとに分解）
C 委員	組込みソフトならではの設計、実装、テストについてももう少し深くしては。
F 委員	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。

④ 実施方法について

委員名	コメント原文（ただし論点ごとに分解）
H 委員	講義だけではなく、演習を含む内容であれば、学生は、さらに興味深く学習を進められるのではないか。

⑤ メンテナンス

委員名	コメント原文（ただし論点ごとに分解）
H 委員	ハードウェアの技術動向など、時代に沿った見直しが必要な領域であり、今後とも時代を意識したカリキュラムを期待する。

(5) IT（インフォメーションテクノロジー）について

① 賛同する点

委員名	コメント原文（ただし論点ごとに分解）
F 委員	産業界として期待している領域である。
F 委員	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。
H 委員	目標が明確になっており、構成、内容ともに非常にわかりやすくよいカリキュラム標準になっている。

② 基本方針

委員名	コメント原文（ただし論点ごとに分解）
A 委員	最新の技術を応用した新しいビジネスの実装方法を試行錯誤し提案する情報専門家として、IS や SE との違いをはっきりさせるために、先端技術者としての能力に重点を置くように area 別の配分を見直す必要がある。
A 委員	SaaS や SOA など、その時点の最新技術動向なども一部取り入れることが可能なカリキュラムの見直しの仕組みが必要。
C 委員	知識の範囲が広いため、IT システムの運用管理に必要な知識と、IT システム構築のための知識のバランスについて方針を明示すべき。
D 委員	対象となる組織や情報システムの課題と密接であることから、一段階高度な教育（たとえば、大学院教育や社会人教育）として再整理すべき。（一部の意見）
D 委員	現在、IS や IT の特徴的な部分は入社後 3~5 年の実務経験の後に担当しており、学部教育では、この入社後 3~5 年の実務経験に相当する部分を、如何に効率よく体系的に学ぶかという点に着目していただきたい。
D 委員	学習時間が限られているので、細分化して即戦力としての能力を高めるよりも、多様な実務に対応しうるに最低限必要な“共通的な知識・能力”を有していることの方が、企業・学生の双方にとって有用ではないか。

③ 他領域との重複について

委員名	コメント原文（ただし論点ごとに分解）
B 委員	章立て（ユニットの内容）は適切だと思われます。他セクションと重複する項目については他セクションに任せ、最新動向や、情報システムが持つ社会性、システムのライフサイクルの考え方などに時間をかけるようにしてはいかがでしょうか。

④ 重視すべき（不足している）内容

委員名	コメント原文（ただし論点ごとに分解）
A 委員	領域に係わる基本的な知識に関する項目を充実させるよう見直しが必要。
A 委員	情報システムの信頼性を高める観点からの見直しも必要。
F 委員	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。
F 委員	ビジネスモデルを用いた講義時間も確保すべき。
H 委員	個々のソフトウェア、プログラム作成に必要な知識も重要だが、情報システム全体の設計や構築、保守、運用に関する部分により重点をおくことが望ましい。

⑤ 構成（時間）

委員名	コメント原文（ただし論点ごとに分解）
H 委員	IT 分野が育てる人材のイメージ、ゴールとなる職種・資格のイメージとの関係では、履修時間数のバランスがとれていない。

J07に対するコメント

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						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、トレースだと思う。トレースを中心に+10単位ほど必要なのではないか。
	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						
MR		マルチメディア表現	3		情報理論として習得させたい。				
	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	-						

JO 7に対するコメント

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る。				
	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							
	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きているため。(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	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							

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. 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			
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。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。			
	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	経営的な視点からの内容で			
	IS2. 3. 1	計測とモデル化	3		ユーザ組織(経営層から末	あるならば、その視点を明			

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. 3. 2	確実性、不確実性およびリスクの下での意思決定	3		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。	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. 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+						
	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						
	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、				

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. 1	現行の定期的、専門的、学術的刊行物	1			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. 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. 1	コミュニケーション能力	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、日々の学習の中で繰り返し反芻するよう指導するとよいと考えます。	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. 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						
	IS2. 11	基本的な組織の機能	4	++					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。
	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	+					

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. 15	会計監査と管理	2	+					
IS3		システムの理論と開発							
	IS3. 1	システムと情報の概念	2		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。	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。			
	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	情報システムの特長	2						
	IS3. 2	システム開発のアプローチ	5		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。作り上げる構造物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。	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。		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。	
	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	+					
	IS3. 3	システム開発の概念と方法論	5		ビジネスアプリケーション全体に対してオブジェクト指向技術がどう位置づけられるものであったとしても、目指すものや良さ、限界などを理解しておくことは必須と考えます。アプローチに関しても、何が違うのか、メリットとデメリットは何かなどは重要な知識と考えます。	CMMI (Capability Maturity Model Integration) や SLCP (Software Life Cycle Process) の考え方を紹介するとしたら、この章でしょうか。			
	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		システム開発システム、システム管理システムとしての「リポジトリシステム」に支援されたシステム設計・開発・維持環境とその運用に関する知識は必須と考えます。「IS3. 4. 1: CASE」を拡張しても良いでしょう。	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。		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。 →3. 4. 3. 2 Boehm, 1984	
	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	+					

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. 9. 4	創造的な設計プロセスを促進する技術	1		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. 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						
	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ま			
	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の多いので、それがわかるようにする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、先の方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						

JO7に対するコメント

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。			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。
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。			
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。			
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			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	SE以外の領域でも共通として必要ではないか。			
一般工学基礎				22.5				
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						
		CMP.cf.6 システムの基礎						
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						
		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)						
		FND.ef.7 ソフトウェア工学以外の分野における工学原理(材料強度、デジタル回路の原理、論理設計、熱力学の基礎)						
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						

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
		PRF.pr.2	倫理綱領とプロフェッショナルとしての行動					
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				<p>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。</p> <p>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する。</p>
	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.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	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					
		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		<p>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。</p>		
	CMP		コンピュータ基礎					
		CMP.cf	コンピュータ科学基礎					
		CMP.cf.9	プログラミング言語の基礎					
		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	ミドルウェア(コンポーネントとコンテナ)					
		CMP.ct.12	分散ソフトウェアのための構築技術					
		CMP.ct.13	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.14	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.15	プラットフォーム標準(POSIXなど)					
		CMP.ct.16	テストファーストプログラミング					
		CMP.tl	構築のためのツール					
		CMP.tl.1	開発環境					

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	
		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						
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.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.dd		詳細設計						
		DES.dd.3	コンポーネント設計						
		DES.dd.4	コンポーネントとシステムのインタフェース設計						
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。			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.md.1	モデリングの原則(分解、抽象化、汎化、投影/ビュー、明快性、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(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)						
		MAA.tm.3	構造のモデリング(アーキテクチャなど)						
		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	機能のモデリング(コンポーネント図など)						
		MAA.tm.6	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、ゴールなど)						
		MAA.tm.7	組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、外部インタフェース分析など)						
		MAA.tm.8	要求競合のための分析(機能競合、品質の家、ビューポイント分析など)						
		MAA.tm.9	アナリシスパターン(プロブレムフレーム、仕様の再利用など)						
	MAA.rv		要求の評価						
		MAA.rv.1	レビューとインスペクション						
		MAA.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)						
		MAA.rv.3	受け入れテストの設計						
		MAA.rv.4	品質特性の評価						
		MAA.rv.5	形式言語による分析/モデル検査						

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.9には含まれている)
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.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	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				
形式手法				22.5				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					
	CMP.fm		形式手法					
		CMP.fm.1	抽象機械の適用(SDL、Paisleyなど)					
		CMP.fm.2	仕様記述言語および技法の適用(ADM、B、CPS、VDMなど)					
		CMP.fm.3	仕様からのソースコードの自動生成					
		CMP.fm.4	プログラム導出					
		CMP.fm.5	候補となる実装の解析					
		CMP.fm.6	異なる実装と仕様とのマッピング					

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
		CMP.fm.7	詳細化					
		CMP.fm.8	正当性の検証					
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、明快性、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	数理モデルと仕様記述言語(ZやVDM)の紹介					
		MAA.md.4	モデリング言語の性質					
		MAA.md.5	モデルの表現と意味(モデルの表現の理解)					
		MAA.md.6	明快性(前提が全くない場合、全ての前提を記述する場合)					
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					
		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	システム目的の策定(ユーザ参加型設計、ステークホルダー間の Win-Win関係、品質機能展開、プロトタイプングなど)					
		FND.ec.3	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)					
		FND.ec.4	システム価値の実現(優先順位付け、リスクの解決、コストのコントロールなど)					
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.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					
EVO			ソフトウェアの進化や保守					
	EVO.pro		進化や保守のプロセス					
		EVO.pro.1	進化や保守の基礎					
		EVO.pro.2	進化や保守の対象間の関係(前提、要求、アーキテクチャ、設計、ソースコードなど)					
		EVO.pro.3	進化や保守のモデル(理論面、法制面など)					
		EVO.pro.4	進化や保守のコストモデル					
		EVO.pro.5	進化や保守の計画(アウトソーシング、内製など)					
	EVO.ac		進化や保守のアクティビティ					
		EVO.ac.1	レガシーシステムの扱い(ラッパーの利用など)					
		EVO.ac.2	プログラムの理解とリバースエンジニアリング					
		EVO.ac.3	(技術面とビジネス面での)システムおよび業務プロセスのリエンジニアリング					
		EVO.ac.4	影響解析					
		EVO.ac.5	(技術面とビジネス面での)マイグレーション(移行)					
		EVO.ac.6	リファクタリング					
		EVO.ac.7	プログラム変換					
		EVO.ac.8	データのリバースエンジニアリング					
PRO			ソフトウェア開発プロセス					
	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)					

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
	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(個人の改善、チームの改善)					
	PRO.con.6	品質の分析とコントロール(欠陥予防、レビュー、品質特性、根本原因分析など)					
	PRO.con.7	プロセスモデルのモデリングと分析					
	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(アジャイル、ヘビーウェイト、ウォーターフォール、スパイラルなど)					
	PRO.imp.3	ライフサイクルモデルの標準(IEEE標準、ISO標準など)					
	PRO.imp.4	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.5	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.6	プロセスのテーラリング					
	PRO.imp.7	ISO/IEEE 12207: プロセスの要求					
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。		
	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、ユーザビリティなど)					
	QUA.cc.6	品質向上に対する取り組みの次元					
	QUA.cc.7	人、プロセス、技法、ツール、技術の役割					
	QUA.std	ソフトウェア品質に関する標準					
	QUA.std.1	ISO 9000シリーズ					
	QUA.std.2	ISO/IEEE 12207: 「包括的」な標準					
	QUA.std.3	組織ごとの標準の適用					
	QUA.std.4	IEEEにおける品質関連の標準					
	QUA.pro	ソフトウェア開発プロセスの改善					
	QUA.pro.1	ソフトウェアプロセス改善のモデルとメトリクス					
	QUA.pro.2	プロセス改善モデルの持つ品質に関する側面					
	QUA.pro.3	ISO15504の概要					
	QUA.pro.4	ISO 15504における品質関連の領域					
	QUA.pro.5	ソフトウェア CMMおよび CMMIIにおける品質関連の領域					
	QUA.pro.6	ソフトウェア産業に対するマルコムボルドリッジ賞の基					
	QUA.pro.7	その他のプロセス(改善)モデル					
	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	プロセス保証の整理と報告					
	QUA.pca.4	プロセス保証の技法					
	QUA.pda	製品の保証					
	QUA.pda.1	製品保証の起源					
	QUA.pda.2	保証と V&Vの違い					
	QUA.pda.3	製品品質モデル					
	QUA.pda.4	根本原因分析と欠陥予防					
	QUA.pda.5	製品品質のメトリクスと測定					

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
		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				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					
		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.2	システム目的の策定(ユーザ参加型設計、ステークホルダー間の Win-Win関係、品質機能展開、プロトタイピングなど)					
	PRF		プロフェッショナルプラクティス					
		PRF.psy	グループダイナミクス / 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	ステークホルダーとの対話					
		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					
		PRF.com	(SEに特化した)コミュニケーションスキル					
		PRF.com.1	読み込み、理解、要約(ソースコードやドキュメントなど)					
		PRF.com.2	記述(職務記述書、報告書、評価報告書、理由書など)					
		PRF.com.3	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	プロフェッショナルリズム					
		PRF.pr.1	ア kredite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					
	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	品質特性(非機能特性)の分析(安全性、セキュリティ、性能、根本原因分析など)					
		MAA.af.4	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					
		MAA.af.5	トレーサビリティ					
		MAA.af.6	フォーマルメソッドによる分析					
		MAA.rfd	要求分析の基礎					
		MAA.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)					
		MAA.rfd.2	要求分析のプロセス					
		MAA.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)					
		MAA.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)					

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
	MAA.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)					
	MAA.rfd.6	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					
	MAA.rfd.7	要求とアーキテクチャの競合					
	MAA.rfd.8	システムエンジニアリングや人間中心設計と要求との					
	MAA.rfd.9	厄介型問題(構造の歪みに起因する問題、解がたくさんある問題など)					
	MAA.rfd.10	制約としての COTS					
	MAA.er	要求の獲得					
	MAA.er.1	要求を獲得する対象(ステークホルダー、ドメインエキスパート、操作環境や組織環境など)					
	MAA.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)					
	MAA.er.3	進んだ要求獲得の技法(民族的技法、知識獲得な					
	MAA.rsd	要求の仕様化と文書化					
	MAA.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)					
	MAA.rsd.2	ソフトウェア要求の仕様化					
	MAA.rsd.3	仕様記述言語(構造化自然言語、UML、Z・VDM・SCR・RSMLなどの形式言語)					
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.con.1	一般的なプロジェクトマネジメント					
	MGT.con.2	古典的なマネジメントモデル					
	MGT.con.3	プロジェクトマネジメントの役割					
	MGT.con.4	全社的/組織的なマネジメントの構造					
	MGT.con.5	ソフトウェア開発マネジメントの種類(調達マネジメント、プロジェクトマネジメント、開発マネジメント、保守マネジメント、リスクマネジメントなど)					
	MGT.pp	プロジェクトの計画					
	MGT.pp.1	計画と評価					
	MGT.pp.2	WBS(Work Breakdown Structure)の作成					
	MGT.pp.3	タスクのスケジューリング					
	MGT.pp.4	工数の見積もり					
	MGT.pp.5	リソースの割り当て					
	MGT.pp.6	リスクマネジメント					
	MGT.per	プロジェクトのメンバと組織					
	MGT.per.1	組織の構造、位置づけ、責務、権威付け					
	MGT.per.2	フォーマルおよびインフォーマルなコミュニケーション					
	MGT.per.3	メンバの割り当て					
	MGT.per.4	メンバの教育、キャリア開発、評価					
	MGT.per.5	会議のマネジメント					
	MGT.per.6	チームビルディングとモチベーション向上					
	MGT.per.7	対立の解消					
	MGT.ctl	プロジェクトのコントロール					
	MGT.ctl.1	変更のコントロール					
	MGT.ctl.2	モニタリングと報告					
	MGT.ctl.3	結果の測定と分析					
	MGT.ctl.4	修正と挽回					
	MGT.ctl.5	インセンティブと規律					

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
		MGT.ctl.6	実績の基準					
		MGT.cm	ソフトウェア構成管理					
		MGT.cm.1	リビジョン管理					
		MGT.cm.2	リリース管理					
		MGT.cm.3	支援ツール					
		MGT.cm.4	ビルド					
		MGT.cm.5	構成管理のプロセス					
		MGT.cm.6	保守に関する考慮事項					
		MGT.cm.7	分散環境とバックアップ					

JO7に対するコメント

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。			
	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。			
	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					
	CE-OPS0	歴史と概要	1		RTOSを作る人をイメージすると、時間が短いのでは。また、RTOSを活用する人とすると、設計の原則はコアとしたい。いずれを目指すのか？			
	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い。			
	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。
	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	-					
CE-SWE		ソフトウェア工学	16					・システム開発スキル向上の

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-SWE0	歴史と概要	1					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。	
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						
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						
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	-						

J07に対するコメント

インフォメーションテクノロジー (IT: information technology)			コア	コア	変更	A委員	変更	C委員	変更	F委員	H委員		
ITF	IT基礎	33	28	-5	ITは、先端的な情報技術を応用した即実現可能なビジネスモデルの提案を行うことを目指す。卒業後も毎年開発される新技術を吸収し提案し続けることを目的として、在学中は、過去のITを広く基礎から理解する力を養うべきと考え					・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。			
ITF1	ITの一般的なテーマ	17	17										
ITF2	組織の問題	6	6										
ITF3	ITの歴史	3	3										
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	2	2										
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									
IAS	情報保証と情報セキュリティ	23	23		※エリア毎のバランスのみ考慮し、増やすべきエリア、減らすべきエリアに関する意見を数値で表記した。								
IM	情報管理	34	28	-6									
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									
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									
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										
SIA	システムインテグレーションとアーキテクチャ	21	20	-1									
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										
SIA5	テストと品質保証(QA)	3	3										

インフォメーションテクノロジー (IT: information technology)				A委員	変更	C委員	変更	F委員	H委員
	SIA6	組織の特性	1	1					
	SIA7	アーキテクチャ	1	1					
							+		
SP		社会的な観点とプロフェッショナルとしての課題	23	22	-1				
WS		Webシステムとその技術	21	21					