# 第 4 章　評価システム

## 1. 評価の手順

JASA の推奨する訓練期間は、第 3 章モデルカリキュラムに沿って 6 カ月とした。
その期間中、各ステップならびに節目に中間評価（マイルストーン）を設定した。

### 1）Off-JT 期間

9 科目 320 時間の Off-JT は、1 日 8 時間換算で 40 日に及ぶことから、訓練科目の理解度を講師が把握しながら進行するため、受講者のモチベーションを維持するため、各科目終了後（翌訓練日）に「確認テスト」を実施することとした。

また、Off-JT 終了時には各確認テスト結果と全体を通した講師所感をまとめ、Off-JT 評価レポートとして、受講者の企業担当者にフィードバックすることとした。

### 2）Off-JT 期間後（OJT 導入時）

組込み技術協会が主催する「ETEC(組込み技術者試験)」を受験させる。

この ETEC は、人材不足が叫ばれている中、組込み企業が人材育成を計画的に推進するために、組込み技術知識レベルを数値化できるツールである。

(詳細は ETEC ホームページ参照　http://www.jasa.or.jp/etec/)

この試験を OJT 導入時に行い、受講者のレベルを計ることで、客観的評価基準による新人訓練の育成到達度、OJT 運用のフォローアップ、訓練計画修正等の資料（たとえば、必要に応じて補習を設定）などに活用が可能になる。

[参考]　ETEC によるモデルカリキュラム(Off-JT)の効果測定結果

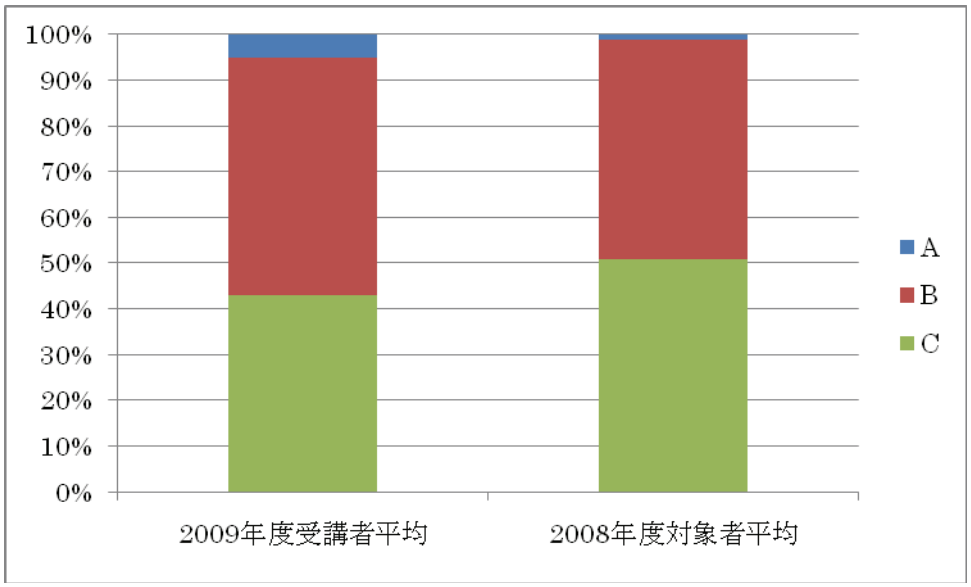
JASA では、2009 年にモデルカリキュラムを会員企業に導入するにあたり、2008 年度の新入社員を対象に ETEC のモニター受験をさせ、データを取得した。

その結果、モデルカリキュラム導入前(2008 年度新入社員)の平均点は 396 点であったのに対して、モデルカリキュラム導入後(2009 年度 OFF-JT 後)の平均点が 423 点となり 27 点アップを果たした。800 点満点に対して、3.8%の知識レベルの向上が見られた。

試験での評価基準として点数に連動したグレードも設けているが、モデルカリキュラム導入後 A グレード 5%に対して導入前は 1 %と高い水準となった。グレード A は組込み知

識を十分有しているという評価であり、レベルの高い知識保有者がいると伺がえる。またグレードCが43%となり導入前の51%から8%改善されており、モデルカリキュラム実施が全体を押し上げている効果を上げていると考えられる。

図1




ETECでは、グレード以外に知識要素を「技術要素」「開発要素」「管理技術」の3つに分けて測定しており、技術要素が導入後に48%(導入前の+3%)、開発技術も導入後に50%(導入前の+4%)と成果を上げている。管理技術が導入後に46%（導入前の-6%）と下げているが、管理技術は通常新入社員レベルに要求されるものではなく、入社前のバックグラウンドが影響した結果と想定される。

表1

| 受験者数 | 2009年度受講者平均 | 2008年度対象者平均 | 全受験者平均 |
|---|---|---|---|
| 技術要素 | 48% | 45% | 57% |
| 開発技術 | 50% | 46% | 61% |
| 管理技術 | 46% | 52% | 66% |

組込み企業における組込み知識は開発経験で知識が高くなる結果がでている。まとまった教育の時間が確保できる新人訓練以外に知識習得をするチャンスはほとんどなく、業務外でスキルアップするのみになるので、新人訓練の重要性はますます増していくと考えられる。

### 3）OJT 中の評価

まずは OJT 導入時に OJT 終了時に行う「ジョブ・カード〔評価シート〕」の説明を OJT 指導者が受講者に行うことで、到達目標を持たせることを「評価シート利用マニュアル」で定めている。

3 か月以上に及ぶ OJT 期間、現場習熟度合いを 1 カ月～2 カ月の節目の時期に、中間評価を設定することを評価モデルとして組み込んでいる。

OJT 終了時に行う「ジョブ・カード〔評価シート〕」を用いて行うことで、受講者には到達度と先の目標、OJT 指導者には OJT 計画の進捗確認となる。

### 4）OJT 終了後の評価

「ジョブ・カード〔評価シート〕」にて正式な評価を実施し、ABC3 段階評価のうち、A または B が 8 割以上を合格とする。

## 2. 評価項目の設定意図(平成 20 年度策定版)

ジョブ・カード〔評価シート〕のうち「Ⅲ技能・技術に関する能力 (2)専門的事項」の評価項目の設定にあたり、委員会にでは、新人から管理職まで利用できる評価項目の策定を目指した。

また、その際に、既に組込みソフトウェア開発業界で標準になっている評価項目があるか調査を行い下記 2 つが候補となった。

1．『組込みスキル標準 (ETSS)』: 独立行政法人　情報処理推進機構　ソフトウェア・エンジニアリング・センター

2．『平成 19 年度「総合的かつ体系的な職務分析の推進に関する調査・研究」まとめ』: 独立行政法人　雇用・能力開発機構　職業能力開発総合大学校　能力開発研究センター

委員会にて、評価項目として標準的に使用してよいか内容を確認し、下記の結果となった。

1．組込みソフトウェア開発技術者のスキル項目を定義するフレームワークの提供にとどまり、それぞれの開発ドメイン（車載開発、携帯電話開発、家電製品開発など）毎に定義する必要があることが分かった。
但し、スキル項目については第 2 階層、第 3 階層と一部が例として提示されているが、第 2 階層では粒度が荒く、第 3 階層では具体的過ぎて一般的ではなかった。
また、新人から管理職までの対応については、スキル項目を一般的な項目で抽出すれば対応は可能であると判断できた。
スキル項目の抽出には、取り組んだ企業からのヒアリングで半年～1 年かけて実施しているほど難易度が高いため、断念した。

2．様々なドメイン毎に、評価項目が定義されていたが、情報通信機械器具製造業（組込みシステム）　技術　ソフトウェア開発が今回対象とする「組込みソフトウェア開発」に一致していると確認できた。
また、既に公開されている評価項目が新人から管理職までをカバーしていることも確認できた。
今回の策定では、既に評価項目が公開されていることも考慮し、評価項目の文言については修正を加えず、分類の仕方についてのみ、委員会にて決定した。
分類は「能力ユニット」を「ソフトウェア詳細設計、ソフトウェアコード作成、プログラムテスト」の 3 つとし、「職業能力要素の細目」単位で対応させた。
＜専門的事項への対応結果＞参照

今回の評価項目については、トライアル企業に実施後にコメントをいただき、また、ヒアリングも実施し評価時の課題などを洗い出すこととした。

評価維持の課題については、再度委員会にて検討し、反映することとした。

＜専門的事項への対応結果＞

**専門的事項**

| **能力ユニット** | **職務遂行のための基準** | *職業別能力要素の細目の内容* |
|---|---|---|
| **ソフトウェア詳細設計** | *能力要素の細目* | *1.ソフトウェア詳細設計* |
| | | *2.NTCR 設計* |
| | | *3.ソフトウェア詳細設計レビュー* |
| **ソフトウェアコード作成** | *能力要素の細目* | *1.プログラミング* |
| | | |
| | | |
| **プログラムテスト** | *能力要素の細目* | *1.プログラムテスト設計* |
| | | *2.コードレビュー* |
| | | *3.プログラムテスト項目レビュー* |
| | | *4.プログラムテストの実施* |
| | | *5.不具合情報の解析* |

## 3. 評価項目の改定点

ジョブ・カード〔評価シート〕のうち「Ⅲ技能・技術に関する能力 (2)専門的事項」に関してもカリキュラム同様､平成 21 年度本訓練実施企業に協力いただき､見直しを図った。

平成 21 年度の評価シートは、既存の３つの能力ユニットの項目に対して、各項目の必要性、適切なユニット配置、作業の流れに沿った項目の順番、意図が分かるような表現等を改めて確認した。

## 1）「ソフトウェア詳細設計」の変更点

ジョブ・カード(評価シート)　Ⅲ技能・技術に関する能力　(2)専門的事項　能力ユニット「ソフトウェア詳細設計」の変更点一覧

| 能力ユニット | 職務遂行のための基準 | | |
|---|---|---|---|
| | 項番 | 項目 | 変更点（●）と理由（■） |
| ソフトウェア詳細設計 | (1) | ソフトウェア設計の基本技法を知っている | 変更なし |
| | (2) | ソフトウェア構造記述方法を知っている | 変更なし |
| | (3) | オブジェクト指向設計手法を知っている | ●「構造化設計、オブジェクト指向設計などの」⇒「オブジェクト指向」に変更<br>■項目の内容により項番を整理した。(2)のソフトウェア構造記述方法と内容が重複すため削除、対象項目を限定した。 |
| | (4) | ソフトウェア設計ツールを知っている | 変更なし |
| | (5) | ソフトウェア品質特性を考慮した設計技法を知っている | 変更なし |
| | (6) | 各種OSの仕様を考慮した設計技法を知っている | ●「仕様を考慮した」を追加<br>■項目の内容により項番を整理した。求められる範囲が広すぎるため対象を「OSを利用できる」という意図を明示した。 |
| | (7) | 対象となるデバイスに関するデバイス仕様の情報収集ができる | ●「各種デバイス」⇒「対象となるデバイス」、「知っている」⇒「情報収集ができる」と「適正が判断できる」の2項目に分けることで項番を追加<br>■項目の内容により項番を整理した。求められる範囲が広すぎるため対象を業務に関連する範囲の限定。「知っている」の意図を具体化した。 |
| | (8) | 対象となるデバイスに関するデバイス仕様の適性を判断できる | |
| | (9) | ソフトウェア詳細設計の作業内容を知っている | 変更なし |
| | (10) | ソフトウェア方式設計の非機能要件(信頼性・使用性・効率性・保守性・移植性)を実現する設計ができる | ●「非機能要件」⇒(信頼性・使用性・効率性・保守性・移植性)を追加<br>■項目の内容により項番を整理した。非機能要件の内容を具体的に明示。 |
| | (11) | モジュール独立性の観点(モジュール強度、モジュール間結合度)からモジュール構造を設計できる | ●新項目として追加<br>■設計する上での必要要素として新規に追加した。 |
| | (12) | モジュール独立性の観点からモジュール間インタフェースを設計できる | ●新項目として追加<br>■設計する上での必要要素として新規に追加した。 |
| | (13) | 構造化されたソフトウェア・モジュール内の処理内容を詳細化できる | ●「ソフトウェア方式設計書で定義された」⇒「構造化された」に変更、「データ構造」⇒削除<br>■項目の内容により項番を整理した。求められる内容を明確化した。 |
| | (14) | モジュール内の処理内容をプログラムユニット構成図、状態遷移図などで表現できる | 変更なし |
| | (15) | 設計手法に対応するツールを使うことができる | ●「CASE」⇒削除<br>■項目の内容により項番を整理した。「CASE」に限定する必要がないため限定を解除。 |
| | (16) | ソフトウェアの構造・処理内容からデータ構造を詳細化できる | ●「構造・処理内容を」⇒「構造・処理内容からデータ構造を」に変更<br>■項目の内容により項番を整理した。詳細化する対象目的を明示した。 |
| | (17) | プログラムテスト方針を策定できる | ●「(単体テスト)」⇒削除、「検討ができる」⇒「策定できる」に変更<br>■項目の内容により項番を整理した。「プログラムテスト」に用語統一した。「策定」に用語統一した。 |
| | (18) | ソフトウェア詳細設計書を記述できる(OSを使用しない場合) | ●「(OSを使用しない場合)」⇒追加、「作成」⇒「記述」に変更<br>■項目の内容により項番を整理した。(19)との対比として、設計条件を限定した。「記述」に用語統一した。 |
| | (19) | ソフトウェア詳細設計書を記述できる(OSを使用する場合) | ●「各種OSの仕様に基づいた」⇒「(OSを使用する場合)」に変更、「作成」⇒「記述」に変更<br>■項目の内容により項番を整理した。(18)との対比として、設計条件を限定した。「記述」に用語統一した。 |
| | (20) | ソフトウェア・インスペクション、ピアレビューなどのレビュー手法を知っている | 変更なし |
| | (21) | ハードウェア最適制御設計、実時間設計などの観点からモジュール処理内容・データ構造の妥当性を検証できる | 変更なし |
| | (22) | モジュール処理内容・データ構造について妥当性を検証できる | 変更なし |
| | (23) | ソフトウェア詳細設計書の内容の正確性・妥当性を確認できる | 変更なし |
| | (24) | ソフトウェア方式設計で設計した内容が漏れなくソフトウェア詳細設計に反映されていることを確認できる | ●新項目として追加<br>■設計する上での必要要素として新規に追加した。 |
| | (25) | リアルタイム処理に関わる複雑な問題点を指摘できる | ●新項目として追加<br>■設計する上での必要要素として新規に追加した。 |
| | (26) | 非機能要件(信頼性・使用性・効率性・保守性・移植性)の観点から問題点を指摘できる | 変更なし |
| | (27) | 検証の結果、不適当な部分があればモジュール処理内容・データ構造の見直しができる | 変更なし |
| | (28) | ソフトウェア詳細設計のレビュー結果をレビュー記録にまとめられる | ●「ソフトウェア方式設計担当者をレビューに参加させることができる」⇒全文変更<br>■「レビューに参加させることができる」というスキルの意味が不明、改めて求められる内容を具体的に明示した。 |
| | (29) | 詳細設計に対するレビューの有効性アセスメント(レビューの充分性、網羅性、修正状況等)ができる | ●「アセスメント」⇒「アセスメント」(レビューの充分性、網羅性、修正状況等)を追加<br>■「アセスメント」の指す意味がわからないため内容を例示した。 |
| | (30) | ソフトウェア詳細設計の終了判断ができる | 変更なし |
| | 削除 | (デザインパターンを知っている) | ■「デザインパターン」を示す意味が不明確なため、削除した。 |
| | 削除 | (ソフトウェア方式設計書からソフトウェアの構造・処理内容を詳細化できる) | ■(13)に集約したため、削除した。 |
| | 削除 | (新たな機能追加や原価低減などのための変更容易性を考慮した設計ができる) | ■変更容易性の示す範囲が不明瞭なため、削除した。 |
| | 削除 | (各種デバイスの仕様に基づいたソフトウェア詳細設計書を作成できる) | ■(7)(8)で求める内容を具体化したため、削除した。 |
| | 削除 | (イベント競合やデッドロック発生といった複雑な問題点を指摘できる) | ■(25)を新規に追加したため、削除した。 |

## 2）「ソフトウェアコード作成」の変更点

ジョブ・カード(評価シート) Ⅲ技能・技術に関する能力 (2)専門的事項 能力ユニット「ソフトウェアコード作成」の変更点一覧

| 能力ユニット | 職務遂行のための基準 | | |
|---|---|---|---|
| | 項番 | 項目 | 変更点(●)と理由(■) |
| ソフトウェアコード作成 | (1) | プログラミング手法を知っている | 変更なし |
| | (2) | プログラミング言語仕様を知っている | ●「言語仕様を知っている」⇒「プログラミング言語仕様」に変更<br>■用語を統一した。 |
| | (3) | ソフトウェア開発環境を選定できる | ●旧項目「各種ソフトウェア構築ツールを使うことができる」⇒削除<br>■改定版(3)～(5)として分割し、より詳細な項目として再定義した。 |
| | (4) | ソフトウェア開発環境を構築できる | ●旧項目「各種ソフトウェア構築ツールを使うことができる」⇒削除<br>■改定版(3)～(5)として分割し、より詳細な項目として再定義した。 |
| | (5) | ソフトウエア開発環境を使うことができる | ●旧項目「各種ソフトウェア構築ツールを使うことができる」⇒削除<br>■改定版(3)～(5)として分割し、より詳細な項目として再定義した。 |
| | (6) | 作成するソフトウェアコードの特性に対応したコーディング規約を策定できる | ●「プログラムコード」⇒「ソフトウェアコード」に変更。項番を変更<br>■用語を統一、項目の内容により項番を整理した。 |
| | (7) | コーディング規約の個別項目の内容を知っている | 変更なし |
| | (8) | コーディング規約を遵守したソフトウェアコードを作成できる | 変更なし |
| | (9) | 作成したソフトウェアコードについてコーディング規約から逸脱している箇所を検出できる | ●「レビュー」⇒「検出」に変更<br>■「不具合の発見」という意味で「検出」とした。 |
| | (10) | ソフトウェア詳細設計書からソフトウェアコードを作成できる | 変更なし |
| | (11) | ソフトウェア詳細設計で設計した内容が漏れなくソフトウェアコードに反映されていることを確認できる | ●項目として新規追加<br>■上記(10)の内容を細分化し、作成～確認という段階ができるように追加した。 |
| | (12) | 各種OSの仕様に基づいたソフトウェアコードを作成できる | 変更なし |
| | (13) | 各種デバイスの仕様に基づいたソフトウェアコードを作成できる | 変更なし |
| | (14) | 非機能要件(信頼性・使用性・効率性・保守性・移植性)を考慮したソフトウェアコードを作成できる | ●項目として新規追加<br>■ISO9126(ソフトウェア品質の評価に関する国際規格)よりソフトウェアの品質特性モデルの構造を引用し定義した。 |
| | (15) | プログラムテストを考慮したソフトウェアコードを作成できる | ●項目として新規追加<br>■テスト工数圧縮による製品コスト圧縮が要求されており、本項目のスキルが要求されているため。 |
| | (16) | ソフトウェアコードレビューでの指摘事項を以降のプログラミング作業に反映できる | ●【プログラムテスト】より項目移動<br>■ソフトウェアコードレビューを、テストスキルではなくコード作成スキルに分類した。 |
| | (17) | ソフトウェア構成管理ツールを使ってソフトウェアコードを管理できる | ●項目として新規追加<br>■ソフトウェア構成管理ツールを使用して管理を行っている企業も有るため。 |
| | (18) | コード静的解析ツールを使うことができる | 変更なし |
| | (19) | ソフトウェアコードの複雑度を分析して構造上の問題点を指摘できる | 変更なし |
| | 削除 | (各種OSの仕様を知っている) | ■【ソフトウェア詳細設計】にも同じ項目があり重複するため、削除した。 |
| | 削除 | (各種デバイスの仕様を知っている) | ■【ソフトウェア詳細設計】にも同じ項目があり重複するため、削除した。 |
| | 削除 | (各種ソフトウェア構築ツールを使うことができる) | ■改定版(3)～(5)として分割し、より詳細な項目として再定義したため、削除。 |

## 3)「プログラムテスト」の変更点

ジョブ・カード(評価シート) Ⅲ技能・技術に関する能力 (2)専門的事項 能力ユニット「プログラムテスト」の変更点一覧

| 能力ユニット | 職務遂行のための基準 | | |
|---|---|---|---|
| | 改定版 | 項目 | 変更点(●)と理由(■) |
| ソフトウェアコード作成 | (1) | プログラムテスト計画を策定できる | ●「検討できる」⇒「策定できる」に用語変更<br>■検討は上流工程で行うため、下流工程のここでは「策定」とした。 |
| | (2) | テスト計画に基づいてプログラムテスト項目を設計できる | 変更なし |
| | (3) | 詳細設計書及びソフトウェアコードをもとにプログラムテスト項目を抽出できる | 変更なし |
| | (4) | プログラムテスト項目の正確性、妥当性を確認できる | 変更なし |
| | (5) | ソフトウェアコードレビュー計画を策定できる | ●「レビュー計画」⇒「ソフトウェアコードレビュー計画」に用語変更<br>■「ソフトウェアコードレビュー」に用語統一した。 |
| | (6) | ソフトウェアコードレビューの有効性アセスメントができる | ●「レビュー計画」⇒「ソフトウェアコードレビュー計画」に用語変更<br>■「ソフトウェアコードレビュー」に用語統一した。 |
| | (7) | ソフトウェアコードの正確性、妥当性を確認できる | 変更なし |
| | (8) | コードカバレッジ基準の種類とそれぞれの違い等を知っている | ●旧「コードカバレッジを知っている」より変更<br>■より具体的な内容に変更した。 |
| | (9) | ホワイトボックステスト・ブラックボックステスト等のテスト手法を知っている | ●旧「ホワイトボックステスト・ユニットテストなどの設計手法を知っている」より変更<br>■設計は上流工程で行うため、下流工程のここでは「手法」とした。 |
| | (10) | 使用するテストツール、デバッグツールの選定ができる | ●旧「テストツール、デバックツールの選定ができる」より変更<br>■企業によりテスト環境は異なるため、「使用する」を追記し、自社環境を意識させる事とした。 |
| | (11) | テストツール、デバッグツール等のテスト環境を構築できる | ●項目追加<br>■選定(項目10)と使用(項目12)の間に、環境構築が必要となるため。 |
| | (12) | テストツール、デバッグツール等のテスト環境を使うことができる | ●旧「ユニットテストツールなどのテストツールを用いてテスト項目を実施できる」より変更<br>■デバッグツールのみしか使用しない企業もあるため、「テストツール」と「デバッグツール」に分割した。 |
| | (13) | テストスクリプト、テストコード（ドライバやスタブ）等を作成できる | ●項目を追加し、「(ドライバやスタブ)」を追加<br>■項目14のレビューの前に、作成が必要とされるため項目を追加し、テストコードの定義を明確にするため「(ドライバやスタブ)」を追加した。 |
| | (14) | 作成されたテストスクリプト、テストコード（ドライバやスタブ）等をレビューできる | ●「(ドライバやスタブ)」を追加<br>■テストコードの定義を明確にするため「(ドライバやスタブ)」を追加した。 |
| | (15) | プログラムテスト項目に従い、必要なテストデータを作成できる | ●項目追加<br>■テストスクリプト、テストコードの作成に加え、テストいコードの作成も重要なスキル項目であるため。 |
| | (16) | プログラムテスト項目に従い、テストを実施できる | ●旧「プログラムテスト計画に従いテスト項目を実施できる」の「テスト項目」より「項目」を削除<br>■実施するのは「テスト項目」ではなく、「テスト」であるため。 |
| | (17) | ソフトウェアテストツールやＩＣＥ（イン・サーキット・エミュレータ）等を用いてテストを実施できる | ●旧「シミュレータ・エミュレータなどのテスト環境を構築・利用できる」を、新項目11.12、17と細分化<br>■環境の構築～テスト実施までを作業手順毎に細分化した。 |
| | (18) | テスト実施時の不具合情報を収集できる | 変更なし |
| | (19) | 不具合発生時の再現環境や再現手順を明確に示すことができる | ●項目追加<br>■不具合の再現は原因究明において重要な要素であるため。 |
| | (20) | 発生している不具合事象から不具合箇所を切り分けることができる | 変更なし |
| | (21) | 不具合の是正策を検討できる | 変更なし |
| | (22) | 不具合是正を目的としたソフトウェアコードの修正ができる | ●旧「ソフトウェアコードを修正できる」より変更<br>■この工程においての「ソフトウェアコードの修正」は、不具合是正を目的とするため。項目の内容により項番を整理した。 |
| | (23) | 不具合に関する情報を障害票として起票できる。 | ●項目追加<br>■各社呼称は違うが、「障害票」を使用している企業が多数あるため。 |
| | (24) | 発生した不具合を、障害管理一覧表にまとめ、品質状況を見える化できる | ●項目追加<br>■各社呼称は違うが、「障害管理一覧表」を使用している企業が多数あるため。 |
| | (25) | 不具合修正時に回帰テストにより修正確認ができる | 変更なし |
| | (26) | テスト実施結果及び不具合情報からテスト終了を判断できる | 変更なし |
| | **移動** | (レビューでの指摘事項を以降のプログラミング作業に反映できる) | ●「ソフトウェアコード作成」項目へ移動<br>■ソフトウェアコード作成でこそ、必要であるスキルのため、当該能力ユニットに移動させた。 |
| | **重複** | (レビューの有効性アセスメントができる) | ■プロジェクトマネージャ等、上級職責者に求められるスキルであるため、削除した。 |
| | **削除** | (各種レビュー手法やインスペクション手法を知っている) | ■この工程では求められていないため、削除した。 |
| | **削除** | (ソフトウェアコードのデザインレビューができる) | ■この工程では求められていないため、削除した。 |
| | **削除** | (ソフトウェア詳細設計担当者をレビューに参加させることができる) | ■プロジェクトマネージャ等、上級職責者に求められるスキルであるため、削除した。 |
| | **削除** | (ソフトウェアコードが正しく作成されているか確認できる) | ■ソフトウェアコードレビューのカテゴリであるため、このカテゴリからは削除した。 |
| | **削除** | (ソフトウェア単体の論理やデータが設計どおりに実装されていることを確認できる) | ■ソフトウェアコードレビューのカテゴリであるため、このカテゴリからは削除した。 |
| | **削除** | (デバックツール・メモリリーク検出ツールを用いることができる) | ■本項目が示す範囲が狭すぎ、かつ目的が不明確であるため、削除した。 |
| | **削除** | (不具合原因を特定できる) | ■本項目(18)～(21)でカバーされているため、削除した。 |