

i コンピテンシディクショナリを用いたソフトウェア開発者の役割に対するレベル評価の試み

山田 悠斗¹ 土居 真之¹ 裕本 真佑¹ 肥後 芳樹¹ 楠本 真二^{1,a)} 塚本 貴弘²
折方 孝夫² 藤原 永年²

受付日 2020年9月16日, 再受付日 2020年12月15日,

採録日 2021年2月10日

概要: 本稿ではある企業でi コンピテンシディクショナリ (iCD) を開発者の自己研鑽に活用することを目指した取り組みを報告する。適用先企業の開発者が、自身が担当する役割ごとの能力の自己評価を、無理なく実施できるようにiCDが提供する評価項目をカスタマイズし(評価項目数の削減等)、回答しやすい評価を行えることを目指したチェックシートを作成した。6名の開発者を対象とした評価の結果、各開発者の経歴や普段の活動を踏まえ、全員に妥当といえる測定結果を得た。また、各開発者が自身で劣っている項目を確認でき、自己研鑽に有効に活用できる可能性を示した。

キーワード: i コンピテンシディクショナリ, ソフトウェア開発, 自己評価

Evaluation of the Level of Software Developer's Role Using the i-Competency Dictionary

YUTO YAMADA¹ MASAYUKI DOI¹ SHINSUKE MATSUMOTO¹ YOSHIKI HIGO¹ SHINJI KUSUMOTO^{1,a)}
TAKAHIRO TSUKAMOTO² TAKAO ORIKATA² NAGATOSHI FUJIWARA²

Received: September 16, 2020, Revised: December 15, 2020,

Accepted: February 10, 2021

Abstract: This paper presents a trial to utilize the iCD (i Competency Dictionary) for developers' self-improvement in a company. We customized the evaluation items provided by iCD (ex. reducing the number of evaluation items) so that the developers of the target company can conduct self-evaluation of their abilities for each role without difficulty. Based on the customized iCD, we created a checklist that is easy for them to answer. The results of the evaluation of six developers showed that all of the results were valid, based on each developer's background and usual activities. In addition, we showed the possibility that each developer can effectively use it for self-improvement by checking the inferior items by themselves.

Keywords: i-competency dictionary, software development, self-assessment

1. はじめに

ソフトウェア開発プロジェクト失敗の原因には、開発方法や技術だけでなく開発者の能力や性格も関係すると言われていることから [1], 開発者の能力や特性の評価や開発

者に必要となる能力を調査することを目的とした研究が盛んに行われている [2], [3]. また、一般にIT企業の開発現場では、開発者の適性を判断して開発チームにおける役割を決定することがしばしば行われ、要求分析者や設計者のような、開発者の役割ごとの開発能力を評価する方法が必要とされている。また、組織の経営層としても自社に適した規模のプロジェクトを請け負うために、組織全体が開発プロジェクトで実施する様々なタスク(要求分析, 設計等)を行ううえで、どの程度のレベルにあるかを把握したいという要望もある。このように、開発者が持つ役割ごとのレベルを、統一された評価基準を用いてできる限り客観

¹ 大阪大学大学院情報科学研究科
Osaka University, Suita, Osaka 565-0871, Japan

² 三菱電機コントロールソフトウェア株式会社
MITSUBISHI Electric Control Software Corporation, Kobe,
Hyogo 650-0027, Japan

^{a)} kusumoto@ist.osaka-u.ac.jp

的に評価する方法が必要とされている。

開発者の開発能力を評価するためのツールとして、“i コンピテンシディクショナリ (iCD)”[4]が存在する。iCD は、IT ビジネスにおける開発者が行うべきタスク（業務）と、タスクを遂行するために必要となるスキル（素養・技術）を列挙した項目群で構成される。更に、開発者の持つタスク遂行力の評価や必要となるスキル判定方法も提供しており、開発者の開発能力を分析する研究でも利用されている [7]。

本稿では、ある組織における開発者の自己研鑽を促進することを旨とした取り組みについて述べる。取り組みにおいては、iCD を積極的に活用することを前提とし、その最初のステップとしてタスクの評価に着目した。すなわち、iCD で提供されているタスクディクショナリと評価手法を利用して、効率的に自己研鑽の仕組みを実現することを目指した。その際に、以下の点を考慮する必要があった。

- (1) 開発者の役割ごとの評価：iCD ではタスクが業務種別や開発分野ごとに分類されており、分類ごとに設計者や実装者といった役割が定義されている。業務種別や開発分野ごとの評価を行ううえでは適しているが、本取り組みでは特定の役割、たとえば、設計者としてのどの業種・分野においても共通的に必要となるタスクを評価したいという要望があった。
- (2) 評価項目数の削減：iCD では評価対象を IT プロジェクト全体としているため、評価項目数が非常に充実していた。(1) で述べたように、本取り組みでは特定の役割で必要なタスクを評価することを目指したため、評価項目数をうまく削減する必要があった。
- (3) タスク遂行力の評価方法：iCD は開発者自身が 5 段階によるタスク遂行力の評価を行うことを想定し、評価基準も提供されている点で非常に有用である。一方で、自己研鑽を行ううえで、なるべく開発者の回答の手間を削減できるように工夫する必要があった。

本稿では、上記の (1)~(3) に対応するために行った取り組みについてまとめた。まず、iCD を用いて開発者の役割ごとのレベル評価を行うために実施した内容を述べる。次に、その結果を利用して、ある企業（以降、A 社）を対象とした役割ごとのレベル評価を試行した。ある開発チームに属する 6 名の開発者に iCD に基づくチェックシートを利用してもらい、A 社で対象とした後述する 6 役割ごとのレベルの測定を行った。各開発者の経験年数との比較や、リーダーによる各開発者の経歴や普段の活動に基づいたフィードバックを交えることで、測定結果の評価を行った。

以降、2 章では本取り組みの背景となる関連研究や諸用語について述べる。3 章では iCD 適用にあたって工夫した内容について述べる。4 章では 3 章の結果として作成したチェックリストの実環境への適用とその結果について述べ

る。5 章では適用結果に基づいた考察を行う。6 章ではまとめと今後の課題を述べる。

2. 準備

本章では本取り組みの背景や用語について簡単に述べる。

2.1 関連研究

開発者の持つ能力や特性の評価に関連する研究はこれまでも盛んに行われている。たとえば、文献 [3] では、石油業界での建設プロジェクトに関して、プロジェクトマネジメントチームの問題解決力を測るために、チーム全体の能力を数値として測定する手法が提案されている。開発者に必要とされる能力の特定や評価に関する研究にはほかにも様々なものが存在するが、能力を評価するための汎用的かつ確実な手法というのは未だに提案されていない。そのため、実環境で評価を行うには、各企業や団体に合わせた独自の評価手法が必要となる。

2.2 ISO/IEC 15504

ISO/IEC 15504 [5] とは、国際標準化機構 (ISO) と国際電気標準会議 (IEC) が策定したソフトウェア開発を中心としたプロセスアセスメントモデルの共通の枠組みを提供したものである。SPICE (Software Process Improvement and Capability dEtermination) とも呼ばれている。プロセス診断モデルなどの枠組みのみを提供しており、完結した方法論は提供していないため、利用する際は各自の環境に合わせたモデルを作成する必要がある。

2.3 Automotive SPICE

Automotive SPICE (A-SPICE)[6] とは、SPICE を母体とした、車載ソフトウェア開発プロセスにおける共通フレームワークである。SPICE とは違い、エンジニアリングプロセスを詳細に具体化していることが特徴であり、各プロセスの作業成果物等を規定している。車載ソフトウェアだけでなく、他のソフトウェア開発にも共通するプロセスが多く存在するため、様々なソフトウェア開発分野において参考にすることができる。

2.4 i コンピテンシディクショナリ (iCD)

i コンピテンシディクショナリ (iCD)[4] とは、IT ビジネスにおける開発者が行うべきタスク（業務）と、タスクを遂行するために必要となるスキル（素養・技術）を列挙した項目群である。個人の持つタスク遂行力の評価や必要となるスキルを判定する方法も提供しており、開発者の開発能力を分析する研究でも利用されている [7]。

iCD はタスクを定義・列挙しているタスクディクショナリと、スキルを定義・列挙しているスキルディクショナリ

表1 タスクディクショナリの構成の概要
Table 1 Overview of the task dictionary structure.

タスク大分類 (49 項目)	タスク中分類 (194 項目)	タスク小分類 (622 項目)	評価項目 (2,438 項目)
アプリケーションシ ステム開発	ソフトウェアコード 作成・単体テスト	プログラミング	開発標準を遵守してコーディングを行う
			設計書に基づいたコーディングを行う
			API や代替プログラムを活用する
	

によって構成されている。各タスクの遂行に必要となるスキルが関係表で紐づけされている。本稿では開発者のタスク遂行力を用いて開発能力を評価することを目的としているため、タスクディクショナリに着目する。

- (1) タスクディクショナリの構成：概要を表1に示す。タスクディクショナリにはタスク大分類、タスク中分類、タスク小分類、評価項目という4種類の項目が存在する。タスク大分類では開発業務の種別や開発分野による分類が行われる。タスク大分類で示した各分野における作業内容がタスク中分類として列挙され、各中分類のより詳細な作業内容がタスク小分類として列挙されている。今回はタスク小分類をタスクと呼ぶこととする。評価項目はタスクごとに複数存在し、各タスクの遂行力を診断するための評価基準となっている。開発者は評価項目に書かれている内容の達成度を診断することで、自身の各タスクの遂行力を評価することができる。評価項目の診断方法として、L0~L4という5段階の診断基準が提供されている。たとえば、表1の場合、「開発標準を遵守してコーディングを行う」という業務を「独力で行うことができる」ならば、一つ目の評価項目はL3とする。
- (2) iCDを用いた評価事例：文献[8]では、iCDを用いた社員の職種ごとのレベル評価が報告されている。スキルではなくタスクに着目することで、一人ひとりの各業務の遂行力を把握することができ、現状の能力の認識や自身に必要な能力の発見につながることを可能としている。また、文献[9]では、三菱総研DCS(株)で実施されているiCDを用いた社員全体を対象とした認定制度と評価基盤について述べている。iCDの適用にあたって、管理部門向けの業務タスクの再定義、新たなタスクの追加を行っている。

3. 本取り組みの内容

3.1 iCD適用のための考慮点

本取り組みではiCDを活用したレベル評価を行う最初のステップとしてタスクの評価に着目した。iCDで提供されているタスクディクショナリと評価手法を利用して、効

率的に自己研鑽するための仕組みを実現することを目指した。一方で、1章でも述べたとおり、本取り組みで対象としている組織においては、iCDを適用するにあたって、以下の点で工夫が必要があった。

- (1) 開発者の役割ごとの評価：iCDではタスクが、タスクプロフィール(ビジネスタイプ別、開発対象別、開発手法別、役割別などからタスクを参照する切り口)として業務種別や開発分野ごとに分類されており、分類ごとに設計者や実装者といった役割が定義されている。タスクプロフィールの切り口にある「役割別」の「役割」は、本稿で扱っている「開発者の役割」とは異なる。たとえば、タスクプロフィールの役割として、「ITアーキテクト」や「ITマイスタ」が存在する。それぞれのタスクとして「大分類：アプリケーションシステム開発」、「中分類：ソフトウェア詳細設計」といった共通の分類が存在し、そこで定義されている小分類にも多くの似たものが存在する。本稿では「タスクプロフィールの役割」は異なっているが、共通して必要なタスクを担当する開発者の役割を表2のように定義している(たとえば「ソフトウェア詳細設計」を担当する開発者の役割が「詳細設計者」)。
- 過去のiCDの適用事例では、業務種別や開発分野ごとの評価が多く行われているが、本取り組みでは特定の役割、たとえば、設計者としてどの業種・分野においても共通的に必要となるタスクを評価したいという要望があった。
- (2) 評価項目数の削減：iCDでは評価対象をITプロジェクト全体を対象としているため、評価項目数が非常に充実していた。(1)で述べたように、本取り組みでは特定の役割で必要なタスクを評価することを目指したため、評価項目数をうまく削減する必要があった。
- (3) タスク遂行力の評価方法：iCDは開発者自身が5段階によるタスク遂行力の評価を行うことを想定しているものであり、評価基準も提供されている点で非常に有用であるが、一方で、自己研鑽を行ううえで、なるべく開発者の回答の手間を削減できるように工夫する必要があった。

表2 本稿で対象とする役割

Table 2 Target roles.

要件分析者	概要設計者
詳細設計者	プログラマ
単体テスト担当者	結合・統合テスト担当者

3.2 対応方法

3.2.1 タスクの役割ごとの分類

本取り組みでは、典型的なウォーターフォール型の開発、すなわち、要件分析、設計、実装、テストを実施する役割として、表2に示す6種類の役割を対象とした。タスクディクショナリ内のタスクは、タスク大分類によって業種や分野ごとに分類されている。したがって、複数の役割に共通して必要となる能力を評価するためには、タスクを役割ごとに分類し直す必要がある。異なるタスク大分類や中分類の間には、タスクや評価項目の内容が全く同じか、一部の単語以外同じというタスクがある程度存在する。そのようなタスクを一つのタスクとしてまとめることで、それぞれのタスクがタスク大分類によらないものとなり、それぞれの役割ごとにタスクを分類し直すことが可能となる。

3.2.2 評価項目数の整理

回答の手間を少なくし、開発者が効率よく評価を行えるように、タスクが持つ評価項目の整理と削減を検討した。まず、評価項目の内容をより直観的に理解しやすいものにするために、「○○の××」(例：単体テスト技法の選択)のような表記にまとめることで簡略化する。続いて、各タスクが持つ評価項目数に上限を設けた。上限以上の評価項目を持つタスクに関しては、項目の削除を行って上限へ収める。項目数を上限へ収める際は、適用組織にとって、より有用な評価項目を残すために以下の二つの選別基準を設けた。

選別基準1：評価項目の内容がタスク名の内容と合致している

選別基準2：評価項目の内容がタスクの成果物を作成するものである

選別基準1は、タスクにより近い内容の評価項目を選別するために設けている。選別基準2は、2.3節で述べたA-SPICE等でも各プロセスから生じる成果物が列挙されていることから、成果物を作成する作業が重要なものとして設けている。

3.2.3 質問事項の作成

回答者が効率よく診断をできるように、評価項目に対する質問事項を作成した。作成する質問事項は回答する開発者の知識や経験に基づいた2択の質問とした。たとえば、「単体テスト技法の選択」という評価項目に対しては以下に示す三つの質問事項を与える。質問事項は、いくつかの

レベルに分かれており、その評価項目に対する実力がより高いと診断されるものをレベルの高い質問事項として設定する。

レベル1：単体テスト技法の種類を知っている

レベル2：単体テスト技法の選択を実施した経験がある

レベル3：単体テスト技法の選択を実施した際に、他者からの評価を受けている

このように、自身のタスクレベルを評価するための判断材料を2択の質問事項として提供することで、開発者が直観的に回答しやすくなることを目指した。以下の(1)~(3)で、その詳細を述べる。

(1) 質問パターンの作成

評価項目には様々な種類があるため、それらに対して効率良く質問事項を作成する必要がある。そこで、iCDでの自己診断の基準を参考にして、いくつかの質問事項のパターンを作成し、各評価項目をより適当なパターンへ割り当てていく。今回は、以下の5種類の質問パターンを作成した。各質問パターンが与える質問事項に関しては、括弧書きの部分が評価項目の内容ごとに変更される部分である。

用語パターン

作業対象への知識があれば、その作業を遂行できるもの。以下の三つの質問事項を与える。

レベル1：(作業対象)の知識を持っている

レベル2：(作業)の実施経験がある

レベル3：(作業)について他者からの評価を受けている

作業パターン

作業を遂行するために、作業対象への知識だけでなく、具体的な手順を理解する必要があるもの。以下の四つの質問事項を与える。

レベル1：(作業手順)の知識を持っている

レベル2：(作業)の実施経験がある

レベル3：(作業)について他者からの評価を受けている

レベル4：(作業)について他者への評価を行っている

業務パターン

作業を遂行する際に専門的な知識や手順を必要としないもの。以下の二つの質問事項を与える。

レベル1：(作業)の実施経験がある

レベル2：(作業)について他者からの評価を受けている

評価パターン

成果物への確認事項に対する自己評価や他人への評価に関係するもの。以下の三つの質問事項を与える。

レベル1：(確認事項)の自己評価を行っている

レベル2：(確認事項)に関して他者からの評価を受けている

レベル3：(確認事項) に関して他者への評価を行っている

指導パターン

ツールや技術の利用方法の指導に関するもの。以下の三つの質問事項を与える。

レベル1：(ツールの利用方法) の知識を持っている

レベル2：(ツール) の利用経験がある

レベル3：(ツールの利用方法) について他者への指導を行っている

(2) 評価項目の質問パターンへの割り当て

各評価項目を上述した5種類の質問パターンのいずれかに割り当てる。そのために、それぞれの質問パターンに適した作業に関連する単語を設定する。設定した単語の一覧を表3に示す。簡略化した評価項目の後半部分と設定した単語を照らし合わせることで、各評価項目をいずれかの質問パターンへ割り当てていく。たとえば「単体テスト技

法の選択」という評価項目の場合は、表3より、用語パターンの中に「選択」という単語が含まれるので、用語パターンへと割り当てられる。

(3) タスクレベルと役割レベルの測定

作成した質問事項への回答結果に基づいて、タスクごと、ならびに役割ごとの評価レベルを測定する。測定の流れを示した例を図1に示す。

まず、質問事項への回答結果から、各評価項目のスコアを測定する。自身に当てはまると回答された質問事項のうち、最も高いレベルの数値が評価項目のスコアとなる。いずれの質問事項にも当てはまらない場合や、普段携わらない作業に関する質問事項である場合は、スコアは0となる。図1の例では、評価項目1-1、1-2、1-3のスコアが2点/4点、1点/3点、4点/4点と測定されている。

続いて、タスクごとの評価レベルを測定する。各タスクが持つ評価項目のスコア合計値と、満点の場合との得点率

表3 各質問パターンに適した作業に関連する単語

Table 3 Terms related to the task appropriate for each question pattern.

用語パターン	優先付け 分界 分割 手配 編成	定義 選択 詳細化 取り組み 準備	作成 列挙 統一 遵守	分類 決定 割り当て 活用	抽出 検出 導入 判断
作業パターン	分析 確保 設定 検証 結合	把握 共有化 テスト 管理 統合	対応付け 構築 チューニング コーディング 計画	設計 対処 開発 向上	見積もり 設置 実装 コードレビュー
業務パターン	責任分担	記録	報告	整理	
評価パターン	評価	レビュー	確認	見直し	
指導パターン	指導				

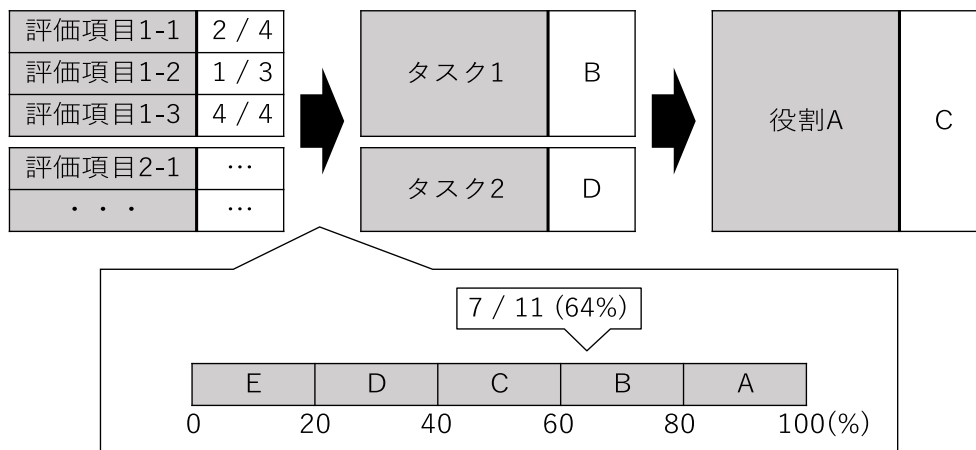


図1 質問事項への回答結果に基づいた評価レベル測定例

Fig. 1 Example of measuring evaluation level based on the results of answering the questionnaires.

を用いてレベルを測定する。得点率が0~20%の場合はレベルE, 21~40%の場合はレベルD, 41~60%の場合はレベルC, 61~80%の場合はレベルB, 81~100%の場合はレベルAと測定される。図1の例では、評価項目1-1, 1-2, 1-3が割り当てられているタスク1に関して、得点率が64%となるので、タスクレベルはBと測定される。

最後に、役割ごとの評価レベルを測定する。各役割が持つそれぞれのタスクについて、レベルE~Aを0~4点として点数化する。そして、先程と同様にして、各タスクが持つ点数の合計値に対する得点率に基づいた5段階のレベル付けを行う。図1の例では、タスク1, 2が割り当てられている役割Aに関して、タスクが持つ点数の得点率が50%となるので、役割レベルはCと測定される。

4. ケーススタディ

本章では、ある組織（以降、A社）の開発者を対象として、iCDを参考に3章で述べた方法で作成したチェックシート（以降、チェックシートと呼ぶ）を用いた役割ごとのレベル評価のケーススタディと適用結果について述べる。今回の適用では、表2に示した6種類の役割を対象とした。

4.1 タスクの追加

A社では開発プロセスを把握する際にA-SPICEを参考に行っているため、役割ごとに分類したタスクに含まれていないA-SPICEの作業を新たなタスクとして追加した。新たなタスクを追加したiCDを以降では調整iCDと呼ぶ。

4.2 チェックシートと感想アンケートの作成

調整iCDをA社の開発者が利用できるように、3章で述べた方法に基づいてチェックシートを作成した。

作成にあたっては、回答者の負担をなるべく減らすように、各作業に対する評価項目は3個以下になるように設定した。なお、チェックシートの元となる3章で述べた方法で作成した調整iCDのタスクディクショナリの構成としては、役割が6種類（表2に示す6役割）、各役割で実施する作業内容が合計67個（たとえば、「要件分析者」の作

業内容としては、「ソフトウェア要件の分析」、「運用環境における影響の特定」、「要件の優先順位付け」等の9項目、評価項目数が合計137個（たとえば、「ソフトウェア要件の分析」の評価項目としては、「要件の実現可能性評価」等の3項目）となった。構成をまとめたものを表4に示す。なお、元々のタスクディクショナリで対応する部分の評価項目数は約950個である（作成したチェックシートでは137個で、約1/8となっている）。

チェックシートにはA社への適用に合わせた評価項目に対する質問事項を列挙する。チェックシートの一部を図2に示す。最上段には、各質問の共通事項がまとめられている。各行には質問事項の固有事項が評価項目ごとに並べられており、回答者は当てはまる質問事項のチェックボックスにチェックを記入する。右の質問ほどレベルの高い内容となっているため、各行の中で複数の項目にチェックが入る場合は、最も右の項目にのみチェックを記入すればよい。

また、チェックシートとは別に、チェックシートに関する感想アンケートを作成した。チェックシートへ回答した後に感想アンケートへの回答を行ってもらい、回答時間や質問内容への回答のしやすさなどを尋ねた。

4.3 適用方法

A社のある開発チームに属する6名の開発者を対象とし、チェックシートと感想アンケートへの回答を行っても

表4 チェックシートの構成要素数

Table 4 Number of items of the checklist.

役割	作業内容数	評価項目数
要件分析者	9	22
概要設計者	15	33
詳細設計者	7	17
プログラマ	17	31
単体テスト担当者	5	10
結合・統合テスト担当者	14	24
合計	67	137

知識を持っている	自己評価を行っている	実施経験がある	他者からの評価を受けている	他者への評価・指導を行っている
要件分析者				
<input type="checkbox"/> 既存Webサイトの分析手順		<input type="checkbox"/> 既存Webサイトの分析	<input type="checkbox"/> 既存Webサイトの分析	<input type="checkbox"/> 既存Webサイトの分析
<input type="checkbox"/> 既存Webサイトの利用傾向の把握手順		<input type="checkbox"/> 既存Webサイトの利用傾向の把握	<input type="checkbox"/> 既存Webサイトの利用傾向の把握	<input type="checkbox"/> 既存Webサイトの利用傾向の把握
...				
概要設計者				
<input type="checkbox"/> 開発手法の選択基準		<input type="checkbox"/> 開発手法の選択	<input type="checkbox"/> 開発手法の選択	
...				

図2 チェックシートの構成

Fig. 2 Structure of the checklist.

らった。それぞれの回答に制限時間は設けない。開発者のチェックシートへの回答結果に基づいて、各開発者の役割レベルと、各役割が持つタスクのレベルが測定される。各開発者のリーダーにそれぞれの経歴や普段の活動に基づいた測定結果のフィードバックを実施してもらうことで、妥当性の確認を行った。また、リーダーによるフィードバックや感想アンケートへの回答結果から測定方法に対する課題点を抽出し、それらの対策を検討した。

4.4 適用結果

適用結果の一部を紹介する。ある開発者について実際に得られた6種類の役割レベルを図3に示す。この結果から、この開発者は「詳細設計者」としての能力が他の役割と比べて低いことが分かる。また、図3に示した開発者に関する、「単体テスト担当者」の役割が持つタスクごとのレベルを図4に示す。この結果から、この開発者が単体テスト担当者としての能力をより向上させるためには、モジュールと単体テストの一貫性を確認する習慣を身に付ければよいことが分かる。

次に、個々の開発者が持つ6種類の役割レベルの一覧を表5に示す。先程の図3、図4に示した測定結果は開発者5のものである。表4を見ると、分析者や設計者といった役割よりもプログラマやテスト担当者といった役割のほう

が各開発者の平均レベルが高い傾向にある。このことから、A社では基本的に、まず実装やテストの作業経験を培った後に、分析や設計という上流工程の作業へ移行するというキャリアパスを採用していることが読み取れる。

続いて、各開発者の感想アンケートへの回答のうち、特徴的な回答を紹介する。チェックシートへの回答時間を尋ねた項目では、各開発者の回答時間が最短で30分、最長では120分を要したという回答が得られた。120分を要した理由としては、チェックシートに記載されているタスク内容や用語の確認を行った、「実施経験がある」、「他者からの評価を受けている」等の回答の基準の解釈等に時間を要してしまったという報告があった。一つの調査シートの回答としては、回答時間が長くかかりすぎている。回答者の負担を減らして適用可能性を上げるために、回答時間をより短くする工夫が必要となる。また、質問量の適切さや質問内容の回答しやすさを尋ねた項目に関しては、否定的な回答のみが得られた。一つの原因としては、回答の際の基準や用語の説明が不十分であったことも考えられる。質問数や質問内容に関しても、より回答者が回答しやすいものにするための工夫が必要であるといえる。以上、得られた課題に対する対応については、5.2節で考察する。

5. 考察

本章では、A社を対象とした役割レベルの測定結果に対する考察と、測定方法の課題点に対する対策について述べる。

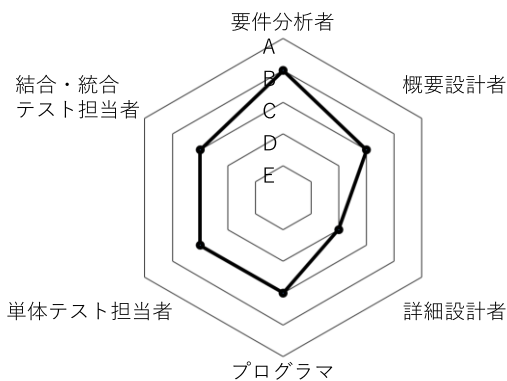


図3 ある開発者の6役割の役割レベル
Fig. 3 Level of six roles of a developer.

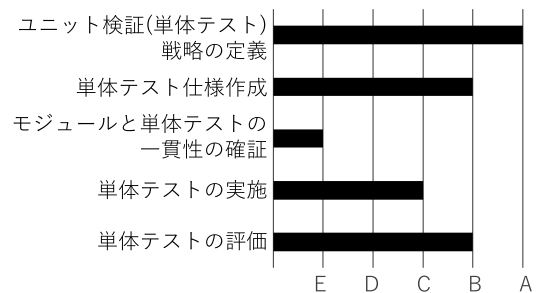


図4 ある開発者の「単体テスト担当者」に関するタスクレベル
Fig. 4 A developer's task level for "unit tester".

表5 A社の6名の開発者が持つ役割レベル

Table 5 Role level of the six developers in company A.

	開発者 1	開発者 2	開発者 3	開発者 4	開発者 5	開発者 6
要件分析者	D	D	D	D	B	B
概要設計者	D	D	D	D	C	B
詳細設計者	D	E	D	D	D	B
プログラマ	D	C	C	C	C	B
単体テスト担当者	D	E	C	D	C	B
結合・統合テスト担当者	C	D	D	C	C	B

5.1 役割レベルの測定結果の評価

A社に所属する6名の開発者が持つ役割レベルを測定し、表5のような結果が得られた。これらの測定結果に対する評価を行った。

まず、開発者1から開発者6の持つ役割レベルについて、経験年数との比較を行った。各開発者の経験年数を表6に示す。「若手」はA社に入社して10年未満の社員、「中堅」は入社10年以上、20年未満の社員、「熟練」は入社20年以上でチームリーダー等を担当している社員である。表4と表5を比較すると、経験年数の長い開発者ほど役割レベルの平均が高い傾向にある。しかし、開発者5に関しては、熟練者としては役割レベルの平均がやや低くなっており、経験年数の長さで役割レベルの高さが一致していない。

そこで、リーダーによる測定結果に関するフィードバックを確認した。各開発者の持つ役割レベルに関しては、各々の経歴から考えると妥当な結果であるというフィードバックが得られた。熟練者としては役割レベルの平均が低い開発者5に関しては、これまでプログラム設計や製作を他所へ依頼するプロジェクトに多く携わってきた開発者であるため、詳細設計者やプログラマに関するランクが低くなっているということであった。よって、調整iCDを用いて得られた測定結果は妥当であるという意見が得られた。

これらのことから、単に経験年数の確認だけでは測定することのできない実力に関しても、調整iCDを用いて測定することができ、開発者自身が自己評価を行い、自分のレベルを確認し、不足している部分を改善していくという自己研鑽に有効であると考えている。一方で、今回のケーススタディでは6名の開発者にすべての役割の評価を行ってもらったが、実際の適用においては、適用時点で主として担当している役割のみを評価するといった対応も必要であることが確認できた。自己研鑽においては、現在の仕事で担当している役割や今後担当する役割についての改善を目指すのが効果的であると考えられる。

5.2 測定方法の課題と対策

適用した測定方法に関して、リーダーによるフィードバックや各開発者の感想アンケートの結果を踏まえて、以下の四つの課題を抽出した。

課題1：回答時間が長くなりすぎている。

課題2：対象チームが普段必要としない分野や作業の質問

が点在していたために、全対象者に関して一部の点数が低くなっている。

課題3：与えられた質問事項が不適切である。質問内容の意図が読み取りづらく、回答者にとって正しく回答できているかが不明な質問が存在している。

課題4：レベルB~Dの役割レベルに関しては、同じ役割レベルであっても開発者間で優劣のある部分が存在する。たとえば、2人の開発者が同じ役割でレベルCと評価されていたとしても、D寄りのレベルC、B寄りのレベルCといった差が存在している。

なお、ケーススタディの結果に関しては、リーダーが各開発者の回答を確認することによって課題2や課題3による誤回答を修正した結果となっている。以降では、これらの課題を解決するための対策を検討していく。

(1) 必須タスクと選択タスクの分類：課題1と課題2を解決するための対策として、役割ごとにまとめられたタスクを「必須タスク」と「選択タスク」というものに分類する。いずれの部署やチームに所属していても必ず行うタスクは「必須タスク」へ分類する。専門性や独自性が存在し、すべての部署が行うとは限らないタスクは「選択タスク」へ分類する。必須タスクに関する質問事項には、全部署の開発者が回答しなければならない。一方で、選択タスクに関しては、各部署のリーダーが自身の部署に必要となるタスクを選択し、開発者は選択されたタスクに関する質問事項にのみ回答すればよい。

このような対策を加えることによって、回答すべき質問の数を部署ごとに削減することができるため、回答時間の短縮が期待できる。さらに、各部署のリーダーがタスクを選別することで、開発者が普段携わらない分野の質問に回答することを抑えられるため、測定結果の精度の向上にもつなげられる。異なる部署同士の開発者の役割レベルを比較したい場合は、必須タスクのみを用いて得られる測定結果を比較すればよい。

(2) 質問事項の修正：課題3を解決するための対策として、各質問パターンによって作成される質問事項の修正を行う。5.4節に示した質問事項は5種類にまとめられる。回答者がより正確にこちらの意図に沿って質問事項に回答できるようにするために、5種類の質問事項に対する意図の明記と、一部の質問内容の変更を行う。

表6 A社の開発者の経験年数

Table 6 Number of years of experience of developers at company A.

	開発者1	開発者2	開発者3	開発者4	開発者5	開発者6
経験年数	若手	若手	中堅	中堅	熟練	熟練

(個別事項)の知識を持っている

この質問事項は、作業対象などの個別事項に対する知識を正しく有しているかどうかを確認したいという意図から作成されたものである。しかし、「知識を持っている」と判断するための基準が不明である、という意見を多く受けたため、より判断基準を明確にした以下の質問事項に変更する。

(個別事項)の説明を参照できる

詳細な説明として、「研修を受けており、資料のどの箇所を確認すれば参考にできるかを把握している場合を指す」という注釈を明記する。

(個別事項)の自己評価を行っている

この質問事項に対する意図を明確にするため、「作業の確認事項に関するチェックや見直しを自身で能動的に行う習慣がある場合を指す」という注釈を明記する。

(個別事項)の実施経験または利用経験がある

この質問事項に対する意図を明確にするため、「指示やアドバイスを受けながらでよいので、始めから最後までその作業の実施やツールの利用を行った経験がある場合を指す」という注釈を明記する。

(個別事項)について他者からの評価を受けている

この質問事項は、他者からのレビューを受けていればその個別事項に対する正しい知識や実力が得られるとみなして作成されたものである。しかし、A社では全開発者が全作業に関して必ず他者からのレビューを受けているため、実力を把握するにはこの質問は不適切であった。そこで、実力の有無を回答者の経験から確認できるようにするため、以下の質問事項に変更する。

(個別事項)について自身で計画を調整して作業を完了している

詳細な説明として、「作業に変更等が生じた際に、自分自身で計画を調整して作業を完了させた経験がある場合を指す」という注釈を明記する。ただし、評価パターンの場合には、実力の有無ではなく他者からレビューを受けているという経験の有無が重要となるので、元の質問事項を用いる。

(個別事項)について他者への評価または指導を行っている

この質問事項は、作業などの個別事項に関して他者を率いる立場にあるかどうかを確認したいという意図から作成されたものである。しかし、A社では新人以上の開発者であればレビューアとして指定される可能性があるため、他者を率いる立場にあるかを確認するにはこの質問は不適切であった。そこで、以下の質問事項に変更する。

(個別事項)についての講義や指導を行っている

詳細な説明として、「対象に関する講義や、

レビューだけでなく部下に対する指導を行っている場合を指す」という注釈を明記する。

- (3) 役割レベルの詳細化：課題4を解決するための対策として、役割レベルの詳細化を行う。役割レベルがレベルB~Dとなり、その役割が持つタスクレベルに関して役割レベルと2段階離れたものが存在する場合は、役割レベルにプラスまたはマイナスの記号を追加する。たとえば、役割レベルがレベルCとなった際に、レベルEのタスクが一つ以上存在する場合は役割レベルをレベルC-とし、レベルAのタスクが一つ以上存在する場合は役割レベルをレベルC+とする。レベルEとレベルAがともに存在する場合はレベルC+-と記述する。

この対策によって、各開発者は自身が持つ役割レベルをより詳細に把握でき、自己改善に役立つと考えられる。たとえば、レベルに-の記号がある場合、その役割で低いタスクレベルの部分を中心に対応することが可能となる。

6. おわりに

本稿では、ソフトウェア開発者が自身が持つ役割ごとのレベル評価を実施することで、自分に足りないところを明確にし、自己研鑽を効率よく行うための取り組みについて述べた。取り組みにあたっては、iCDが提供するタスクディクショナリを用いたレベル評価を活用した。まず、タスクディクショナリには、開発者が所属する組織の特性に応じて、タスクを追加するなどの対応を行った。次に、対応を加えた調整iCDに基づいたチェックシートを作成し、A社の6名の開発者が役割ごとのレベルを自己評価した。その結果、各開発者の経験年数の確認だけでは把握することのできない実力に関しても評価することができ、開発者自身のレベル評価と改善に役立つ可能性が確認できた。

今後の課題としては、以下の項目が考えられる。

- (1) 質問パターン：3.2.3項で述べた5種類の質問パターンは、表2に示した六つの役割を前提としている。したがって、プロジェクトマネージャ等の他の役割に関して、5種類の質問パターンで対応できるわけではない。他の役割に対して5種類の質問パターンで対応可能か、あるいは、別の質問パターンを用意する必要があるかどうかを確認する必要がある。
- (2) 適用事例の充実：今回の評価結果は、4章で述べたケーススタディの元でのものである。今後、提案した手法をA社の他の開発チームも含め、より多くの開発者について適用し、得られた評価結果の分析を行うことで、調整iCD作成にあたって設定した条件(評価項目数の上限(3個)等)、3.2.3項で述べたタスクレベルと役割レベルの測定方法等について、継続的に改善を進める必要がある。

(3) チェックシートの改善とレベルチェックのシステム化: 5.2 節で述べたチェックシートの改善に加えて, タスクだけでなく, タスクとスキルの両方の面から開発者のレベル評価を行うことも必要である. 今回開発したチェックシートに対して iCD 内のスキルディクショナリが持つ要素を組み込むといった方法を検討している. また, ケーススタディではエクセルのシートへの記入としていたが, 多くの開発者が簡便にレベルの自己評価, 自己研鑽を行うためには, チェックのシステム化も必要である. その際には, たとえば, 性格診断テスト等も併用し, それぞれの結果を比較することで, 開発者の特性をより詳細に評価することも考えられる.

参考文献

- [1] Rivera-Ibarra, J.G., Rodríguez-Jacobo, J. and Serrano-Vargas, M.A.: Competency framework for software engineers, In Software Engineering Education and Training (CSEE&T), 2010 23rd IEEE Conference on, pp.33-40. IEEE (2010).
- [2] Peters, L.: A knowledge & competencies checklist for software project management success, In SEKE, pp.241-243 (2014).
- [3] Zadeh, M.T., Dehghan, R., Ruwanpura, J.Y., and Jergeas, G.: An index to assess project management competencies in managing design changes, International Journal of Construction Engineering and Management, Vol.5, No.1, pp.11-24 (2016).
- [4] iCD オフィシャルサイト: <https://icd.ipa.go.jp/icd/>.
- [5] ISO/IEC 15504-5: 2012: Information technology — Process assessment — Part 5: An exemplar software life cycle process assessment model (<https://www.iso.org/standard/60555.html>). (C11)
- [6] S.I.G.Automotive: Automotive SPICE-Process Assessment Model (2007).
- [7] 大月美佳, 掛下哲郎ほか: i コンピテンシ・ディクショナリを活用した J07 および情報処理技術者試験と職種のマッチング分析ツール, 研究報告コンピュータと教育 (CE), Vol.2015, No.19, pp.1-8 (2015).
- [8] 石川拓夫: 2 万人のレベル診断~日立の IT 人材強化の取り組み~. HRDI セミナー「デジタル革命への挑戦~iCD 活用, 更なるその先へ~」, 独立行政法人情報処理推進機構 (IPA) (2017).
- [9] 但吉英山: 社員と組織の持続的成長を目指した iCD 活用事例, 情報処理学会デジタルプラクティス, Vol.11. No.1 (2020).



山田 悠斗 (非会員)

2018 年大阪大学大学院情報科学研究科博士前期課程修了.



土居 真之 (非会員)

2018 年大阪大学基礎工学部卒業, 2018 年同大学大学院情報科学研究科博士前期課程入学. 2020 年同修了.



松本 真佑 (正会員)

2010 年神戸大学大学院システム情報学研究科特命助教, 2016 年大阪大学大学院情報科学研究科助教. 博士 (工学). エンピリカルソフトウェア工学に関する研究に従事.

事.



肥後 芳樹 (正会員)

2007 年大阪大学大学院情報科学研究科助教, 2015 年から同准教授. 博士 (情報科学). ソースコード分析, 特にコードクローン分析, リファクタリング支援, ソフトウェアリポジトリマイニングおよび自動プログラム修正に関する研究に従事.



楠本 真二 (正会員)

1991 年大阪大学基礎工学部助手, その後, 講師, 助教授を経て, 2005 年同大学大学院情報科学研究科教授. 博士 (工学). ソフトウェアの生産性や品質の定量的評価に関する研究に従事.



塚本 貴弘 (非会員)

1997 年三菱電機コントロールソフトウェア株式会社に入社. 公共システム等のソフトウェア開発業務を経て, 技術者の育成業務に従事.



折方 孝夫 (非会員)

1986 年三菱電機コントロールソフトウェア株式会社に入社. 電力システム等のソフトウェア開発, プロジェクトリーダーなどを経て, プロセス改善業務および技術者育成業務に従事.



藤原 永年 (非会員)

1998年三菱電機コントロールソフトウェア株式会社入社。鉄鋼システムのソフトウェア開発を経て、産業分野向けのパッケージ製品開発に従事。