

# 修士学位論文

題目

iコンピテンシディクショナリを利用した  
ソフトウェア開発者が持つ役割ごとのレベル評価の試み

指導教員

楠本 真二 教授

報告者

山田 悠斗

平成 30 年 2 月 7 日

大阪大学 大学院情報科学研究科

コンピュータサイエンス専攻

平成 29 年度 修士学位論文

i コンピテンシディクショナリを利用した  
ソフトウェア開発者が持つ役割ごとのレベル評価の試み

山田 悠斗

## 内容梗概

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ために、開発者の役割ごとのレベルを評価する方法が必要とされている。さらに、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、組織全体の開発者のレベルを測りたいという要望がある。この要望を満たすためには、開発者が持つ役割ごとのレベルを、統一された評価基準によってできる限り客観的に評価することが望ましい。

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) というものが存在する。これは、開発者が行うタスクを業務種別や開発分野ごとに列挙し、各タスク遂行力の評価方法を提供するものである。しかし、設計者や実装者といった役割ごとの評価には適していないことや、診断する項目数が多すぎることで、評価する開発者ごとに評価基準が変動する可能性があることといった課題が存在している。

本研究では、上記の課題を解決するために、iCD を用いて開発者の役割ごとのレベル評価を行う手法の提案を行う。具体的には、まず、役割ごとの評価を可能にするため、タスクディクショナリが持つタスクを業種ごとではなく役割ごとに分類し直す。さらに、現場への実用可能性を高めるために、タスク項目数の整理を行う。加えて、タスク遂行力を評価するための質問事項をタスクごとに新たに作成する。

さらに、対応を加えた iCD を用いて、ある企業に属する 6 名の開発者を対象とした役割ごとのレベル評価を試行する。試行の結果、各開発者の経歴や普段の活動を踏まえると、全員について妥当といえる測定結果が得られた。さらに、各開発者の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、対応を加えた iCD を用いて測定することができた。

## 主な用語

実証的ソフトウェア工学

i コンピテンシディクショナリ

ソフトウェア開発役割

ソフトウェア開発能力評価

## 目次

<b>1</b>	<b>まえがき</b>	<b>1</b>
<b>2</b>	<b>準備</b>	<b>3</b>
2.1	関連研究	3
2.2	ISO/IEC 15504	4
2.3	Automotive SPICE	4
2.4	i コンピテンシディクショナリ (iCD)	4
2.4.1	iCD の構成	4
2.4.2	タスクディクショナリを用いた開発能力の評価	5
2.4.3	iCD を用いた評価事例	7
2.4.4	iCD を用いた評価の課題	8
<b>3</b>	<b>役割レベル評価のための提案手法</b>	<b>9</b>
3.1	タスクの役割ごとの分類	9
3.2	タスク内容の粒度の統一	9
3.3	評価項目数の整理	11
3.4	質問事項の作成	11
3.4.1	質問パターンの作成	12
3.4.2	評価項目の質問パターンへの割り当て	14
3.4.3	タスクレベルと役割レベルの測定	15
<b>4</b>	<b>提案手法のケーススタディ</b>	<b>17</b>
4.1	タスクの追加	17
4.2	チェックシートと感想アンケートの作成	17
4.3	適用方法	17
4.4	適用結果	20
<b>5</b>	<b>考察</b>	<b>24</b>
5.1	役割レベルの測定結果の評価	24
5.2	測定方法の課題と対策	24
5.2.1	必須タスクと選択タスクの分類	25
5.2.2	質問事項の修正	26
5.2.3	役割レベルの詳細化	28

6 あとがき	29
謝辞	30
参考文献	31

## 表目次

1	タスクディクショナリの構成 . . . . .	6
2	タスク「プログラミング」が持つ評価項目 . . . . .	7
3	評価項目の診断基準 . . . . .	7
4	iCDを用いた評価事例における社員のスコアの例 . . . . .	8
5	本稿で対象とする役割 . . . . .	9
6	タスク「パッケージのカスタマイズ方針の決定」が持つ評価項目 . . . . .	12
7	評価項目「単体テスト技法の選択」に与える質問事項 . . . . .	12
8	用語パターンが持つ質問事項 . . . . .	13
9	作業パターンが持つ質問事項 . . . . .	13
10	業務パターンが持つ質問事項 . . . . .	14
11	評価パターンが持つ質問事項 . . . . .	14
12	指導パターンが持つ質問事項 . . . . .	14
13	各質問パターンに適した作業に関連する単語 . . . . .	15
14	感想アンケートの内容 . . . . .	19
15	A社の6名の開発者が持つ役割レベル . . . . .	20
16	感想アンケートの回答結果 . . . . .	23
17	A社の開発者の経験年数 . . . . .	24
18	必須タスクと選択タスクの選別の例 . . . . .	26

## 目次

1	iCD の構成 . . . . .	5
2	役割ごとのタスク分類の例 . . . . .	10
3	質問事項への回答結果に基づいた評価レベル測定の例 . . . . .	18
4	チェックシートの構成 . . . . .	18
5	役割レベル評価の適用方法 . . . . .	21
6	ある開発者の 6 種類の役割レベル . . . . .	21
7	ある開発者の役割「ソフトウェア要件分析者」に関するタスクレベル . . . . .	22

## 1 まえがき

ソフトウェア開発業界における開発プロジェクト失敗の要因には、プロジェクト全体の開発工程や開発方針だけでなく、各開発者が持つ能力や特性も関係するとされている [1, 2, 3]. そこで、開発者の持つ能力や特性を正確に評価することや、開発者に必要となる能力を調査することを目的とする研究が盛んに行われている [4, 5, 6, 7]. 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設計者や実装者といった役割を決定するために、開発者の役割ごとのレベルを評価する方法が必要とされている [8]. ある開発者に関して、実装者としての経験は十分であっても、設計者としては経験が浅く未熟である場合が存在するためである。さらに、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、組織全体の開発レベルを測りたいという要望がある。この要望を満たすためには、それぞれの役割における能力レベルごとの開発者数をまとめるといった方法が考えられる。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によってできる限り客観的に評価する方法が IT 企業全体で必要とされている。

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) [9] というものが存在する。独立行政法人情報処理推進機構 (IPA) [10] が提供しており、一般企業においても利用されている [11]. iCD の中には、開発者が行う業務をタスクとしてまとめ、業務種別や開発分野ごとに列挙しているタスクディクショナリというものが存在する。タスクディクショナリの中には各タスクの遂行力を評価するための 5 段階の診断基準が設けられており、これを用いることで開発者が自身のタスク遂行力を評価することが可能となる。しかし、iCD を用いた能力評価にはいくつかの課題が存在している。1 つは、タスクが業務種別や開発分野ごとに分類されているため、設計者や実装者といった役割ごとの評価には適していないことである。また、あらゆる業種を網羅しているために、評価項目数が多すぎるという課題も存在する。さらに、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るため、評価する開発者ごとに診断基準が変動する恐れがある。

本研究では、上記の課題を解決するために、iCD を用いて開発者の役割ごとのレベル評価を行う手法の提案を行う。まず、役割ごとの評価を可能にするため、タスクディクショナリが持つタスクを業種ごとではなく役割ごとに分類し直す。続いて、現場への実用可能性を高めるために、タスク項目数の整理を行う。また、タスク遂行力を評価するための質問事項をタスクごとに新たに作成する。ここで作成する質問事項は、開発者の知識や経験に基づいた 2 択の質問にする。これによって開発者が自身で判断しやすいものになり、診断基準の変動を抑制することが期待できる。

さらに、これらの対応を加えた iCD を用いて、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する。あるグループに属する 6 名の開発者に対応を加えた iCD を利用してもらい、A 社の対象とする 6 役割ごとのレベルの測定を行う。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う。

以降、2. では研究の背景となる関連研究や諸用語について述べる。3. では iCD に加える対応について述べる。4. では対応を加えた iCD の実環境への適用とその結果について述べる。5. では適用結果に基づいた考察を行う。6. では本研究のまとめと今後の課題を述べる。

## 2 準備

本章では研究の背景や諸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する研究はこれまでも盛んに行われており、注目されている分野の1つである [12]。ここでは、関連する研究の一部を紹介する。

文献 [4] では、ソフトウェアエンジニア (SE) に必要となるコンピテンシーに関する研究が行われている、コンピテンシーとは、特別な状況で優れた能力を発揮することを可能にする一連の知識・能力・行動のことを指す [13]。ソフトウェア開発プロジェクトを成功させるためには、SE が持つべきコンピテンシーを把握することが重要とされている [14]。そこで、SE が持つべきコンピテンシーを Technical, Social, Personal の3つの分類から列挙し、それぞれの重要度を SE の役割ごとに記入できるようなフレームワークを提供している。このフレームワークは SE が自身に不足しているコンピテンシーを発見することなどに利用できるが、各コンピテンシーの重要度などは利用する組織ごとに調整する必要がある。

文献 [5] では、プロジェクトマネージャ (PM) に必要となるコンピテンシーに関する研究が行われている。ソフトウェア開発プロジェクトの成功は、技術的な要因だけではなく PM の持つコンピテンシーによって左右されるとされている [15]。そこで、PM に必要な基本能力とそれを補うためのコンピテンシーを特定し、それらを高めるための普段の姿勢などを提示している。

文献 [6] では、石油業界における、プロジェクトマネジメントチームの持つコンピテンシーに関する研究が行われている。石油業界での建設プロジェクトはコストやスケジュール管理について期待を満たしていない場合が多く、その原因として計画の変更やエラー等による作業のし直しが多く起こっていることがあげられている [16]。そこで、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、チーム全体のコンピテンシーを数値として測定する手法を提案している。コンピテンシーを数値として得ることによって、見積もりを行う際の変数の1つとして用いることなどが可能となる。

文献 [7] では、SE の性格や特性に関する現在の知識体系を調査するために、系統的レビュー [17] を実施している。SE の性格や特性は業務に強く影響を与えるとされており、それに関する研究が現在も盛んに行われているためである [18]。レビューの結果に基づいて、用いられている心理モデルの紹介や今後の研究案の提供を行っている。

このように、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様々なものが存在する。しかし、開発者に必要とされる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 [19] 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。アセスメントモデルの枠組みと実施のための基本的な要求事項が定められており、ソフトウェア開発プロセスに関する様々な研究でも着目されている [20, 21]。しかし、完結した方法論は提供されていないため、実組織で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) [22] 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、プロセスアセスメントモデルを詳細に具体化しているという特徴があり、各プロセスにおける作業内容や作業成果物を規定している。そのため、プロセスの分析やプロセスごとの能力の評価などに利用することができる [23, 24, 25]。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) [9] 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。独立行政法人情報処理推進機構 (IPA) [10] が提供しており、企業が行う全てのタスクを明確化することによって現状の把握や課題の発見を促し、企業成長に貢献することを目的としている。iCD 内の項目を列挙するために、共通フレーム 2013 [26] や ITIL 2011 Edition [27]、ソフトウェアエンジニアリング知識体系 [28] といった様々なプロセス体系や知識体系が参照されている。また、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 [29]。

### 2.4.1 iCD の構成

iCD の構成を図 1 に示す。iCD はタスクを定義・列挙するタスクディクショナリと、スキルを定義・列挙するスキルディクショナリによって構成されている。タスクディクショナリでは、どのような企業や組織でも活用できるように広範囲で網羅的なタスク群が提供され

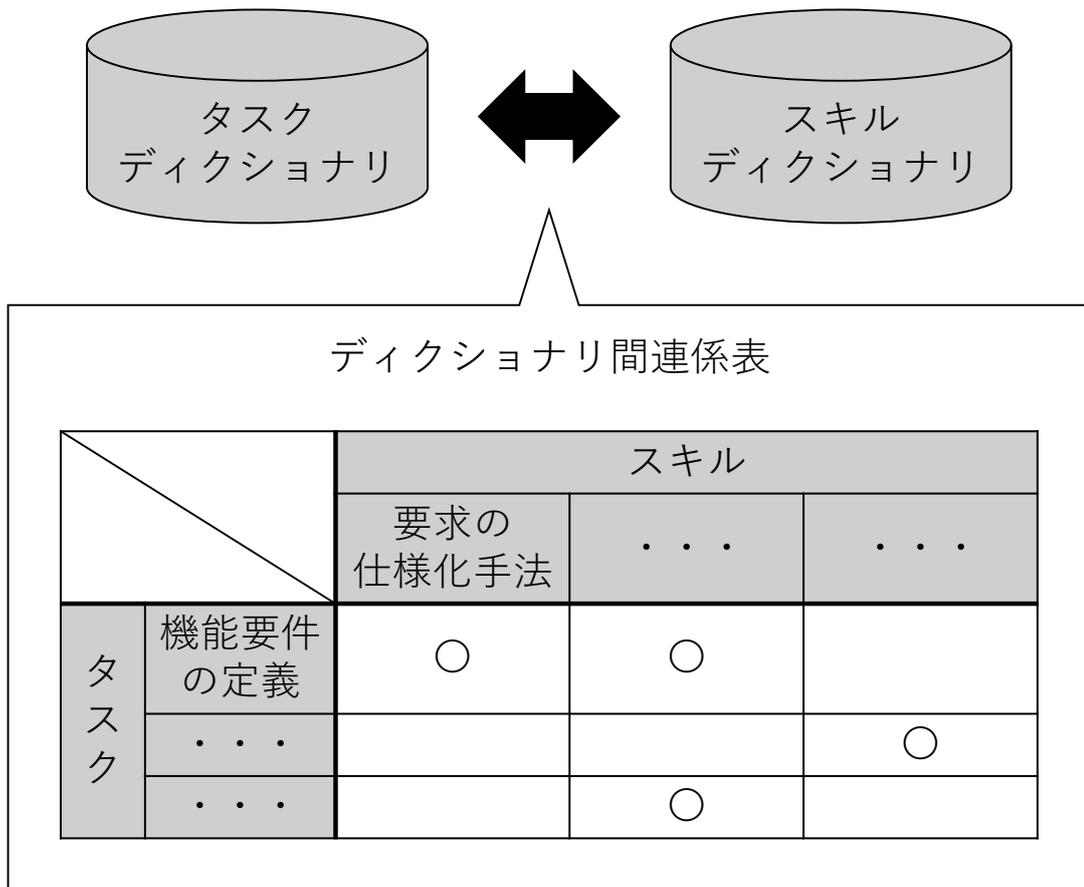


図 1: iCD の構成

ている。スキルディクショナリは、スキルの一覧を示すことで、開発者の現状把握や目標の設定など、スキルに着目した育成活動を支援することを目的としている。また、各タスクの遂行に必要となるスキルがディクショナリ間関係表によって紐づけされている。今回は開発者の業務に関する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。

#### 2.4.2 タスクディクショナリを用いた開発能力の評価

タスクディクショナリの構成を表 1 に示す。タスクディクショナリには以下の項目が存在する。

##### タスク大分類

タスクディクショナリの最も上の階層に位置し、開発業務の種別や開発分野による分

類が行われる。

#### タスク中分類

タスク大分類によって分類された各業務種別や開発分野における大まかな作業内容が  
列挙されている。

#### タスク小分類 (タスク)

各タスク中分類の作業のより詳細な内容が列挙されている。以降は、この項目を単に  
「タスク」と呼ぶ。

また、それぞれのタスクごとに評価項目というものが存在する。これは、各タスクの遂行  
力を診断するための評価基準である。例えば、タスク「プログラミング」が持つ評価項目を  
表2に示す。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。iCDでは、評価項目の診断方法として表3に示す5  
段階の診断基準が提供されている。例えば、表2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、1つ目の評価項目はL3と診断さ  
れる。

表 1: タスクディクショナリの構成

タスク大分類 (49 項目)	タスク中分類 (194 項目)	タスク小分類 (タスク) (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	ソフトウェア詳細設計	ソフトウェア コンポーネント設計 (機能分割・構造化)	...
		入出力詳細設計	...
		...	...
	ソフトウェアコード作成・ 単体テスト	プログラム構造設計 (モジュール分割・ モジュール設計)	...
		単体テスト仕様作成	...
		...	...

### 2.4.3 iCD を用いた評価事例

iCD を用いた評価の実例を紹介する。文献 [11] 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。対象とする職種はマーケティングやプロジェクトマネジメントといった IT スキル標準 [30] に基づいている。iCD に基づいて職種ごとのタスクを列挙し、

表 2: タスク「プログラミング」が持つ評価項目

タスク	評価項目
プログラミング	コーディング規約, 命名規則, レビュー記録, バージョン管理などの開発標準を遵守してコーディングを行う
	設計書 (詳細設計書, プログラム仕様書) に基づいたコーディングを行う
	作成するプログラムの機能を理解し, API や再利用, 代替プログラムを活用する
	データ構造を理解し, データアクセス技術 (SQL) を活用してプログラムを作成する
	処理速度を意識してプログラムを作成する
	セキュリティホールへの対策を理解して, セキュアなプログラムを作成する
	デバッグの手法やデバッガなどの有効な手段を活用してプログラミング上の欠陥や誤りを発見し, 修正を行う
	プログラムのデグレードが発生しないための管理方法を理解し, 実践する

表 3: 評価項目の診断基準

診断レベル	診断基準
L0	知識, 経験なし
L1	トレーニングを受けた程度の知識あり
L2	サポートがあれば実施できる, またはその経験あり
L3	独力で実施できる, またはその経験あり
L4	他者を指導できる, またはその経験あり

タスクごとのスコアを測定することで、社員のレベルを評価している。タスクごとのスコアを測定することによって、一人ひとりのより詳細な能力を把握することが可能となる。例えば、ある社員に関する表4のようなスコアが得られた場合は、プロジェクトマネージャに関するスコアが低いことが分かるため、関連するセミナーの受講の提案などを行うことができる。このように、iCDを用いることによって、現状の能力の認識や自身に必要な能力の発見に繋げることを可能としている。

#### 2.4.4 iCD を用いた評価の課題

iCDのタスクディクショナリを用いた開発能力の評価に関して、いくつかの課題があげられる。1つは、1人の開発者を設計者や実装者といった複数の役割ごとに評価する場合には適していないという点である。タスクディクショナリはタスク大分類によって業務種別や開発分野ごとに分類されており、役割ごとの分類が行われていないためである。また、タスクディクショナリはあらゆる業種を網羅しているため、タスクや評価項目の数が多すぎるという課題も存在する。さらに、評価項目を診断するために5段階の診断基準が提供されているが、この診断方法では自身による評価を行う際にL2やL3といった中間部分の線引きが難しくなる。これによって、回答する開発者によって診断基準が変動してしまう可能性がある。iCDを開発者の役割ごとの評価に利用するためには、これらの課題を解決するための対応を加える必要がある。

表 4: iCD を用いた評価事例における社員のスコアの例

職種	タスク	社員のスコア
コンサルティング	タスク C-1	L3
	タスク C-2	L4
プロジェクトマネージャ	タスク P-1	L2
	タスク P-2	L2

### 3 役割レベル評価のための提案手法

本章では、iCD を用いて開発者の役割ごとのレベル評価を行う手法について述べる。

#### 3.1 タスクの役割ごとの分類

本稿では、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5に示す6種類の役割を考える。タスクディクショナリ内のタスクは，タスク大分類やタスク中分類によって業種や分野ごとに分類されている。そのため，複数の役割に関する能力を評価するためには，タスクと評価項目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・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1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・中分類に依らないものとなり，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。

類似したタスクを1つのタスクとしてまとめ，役割ごとに分類する例を図2に示す。「アプリケーションシステム開発」と「ソフトウェア製品開発」という2つの大分類の間には，ともに「共通基盤の設計」というタスクが存在する。これらのタスクが持つ評価項目についても，「アプリケーション」と「ソフトウェア製品」という単語以外は全て同じ内容である。このような場合は，異なる単語を1つに括ることによって「共通基盤の設計」というタスクを1つのタスクとしてまとめる。このタスクはソフトウェア方式設計者が行うタスクであるため，改めて「ソフトウェア方式設計者」の役割へと分類し直す。

#### 3.2 タスク内容の粒度の統一

タスクディクショナリに存在するタスクは内容の粒度にばらつきがある。例えば，「機能要件の定義」というタスクの場合は，「定義」という1つの作業内容が含まれている。一方で，「開発環境の構築と維持」というタスクの場合は，「構築」と「維持」という2つ以上の作業内容が含まれている。このとき，異なる2つの作業内容の視点から1つの評価を行う必要が

表 5: 本稿で対象とする役割

ソフトウェア要件分析者	ソフトウェア方式設計者
詳細設計者	コーダー(実装者)
単体テスト担当者	ソフトウェアテスト担当者

タスク大分類	タスク	評価項目
アプリケーションシステム開発	共通基盤の設計	・・・
		アプリケーションの要件を考慮して設計する
ソフトウェア製品開発	共通基盤の設計	・・・
		ソフトウェア製品の要件を考慮して設計する



タスク	評価項目
共通基盤の設計	・・・
	{アプリケーションorソフトウェア製品}の要件を考慮して設計する



役割	タスク	評価項目
ソフトウェア設計者	共通基盤の設計	・・・
		{アプリケーションorソフトウェア製品}の要件を考慮して設計する

図 2: 役割ごとのタスク分類の例

あり、タスクの評価が困難となる。このような2つ以上の作業内容を持つタスクを作業ごとに分割することで、1つのタスクに1つの作業内容が含まれるようにする。そうすることでタスク内容の粒度を揃え、タスクごとの評価を行いやすくする。例えば、「開発環境の構築と維持」というタスクの場合は、「開発環境の構築」と「開発環境の維持」という2つのタスクに分割される。

### 3.3 評価項目数の整理

回答の手間を少なくし、現場への実用可能性を高めるために、タスクが持つ評価項目の整理と削減を行う。まず、評価項目の内容をより理解しやすいものにするために、「○○のxx」(例：単体テスト技法の選択)のような文章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る。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、より有用な評価項目を残すために以下の2つの選別基準を設ける。

**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、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。これらの選別基準に基づいてタスクが持つ評価項目を選別する場合の例を示す。「パッケージのカスタマイズ方針の決定」というタスクは、表 6 に示す 5 つの評価項目を持っている。このとき、選別基準 1 によって、「カスタマイズ」と「パッケージ」という単語が含まれている「カスタマイズ内容の決定」と「パッケージとの連携の検討」という評価項目が優先的に選別される。さらに、選別基準 2 によって、設計書という成果物が作成される内容である「アドオン機能に関する方式設計」という評価項目が優先的に選別される。

### 3.4 質問事項の作成

診断基準が変動する可能性があるという課題を解決するために、開発者が評価項目を診断するための質問事項を新たに作成する。このとき、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 択の質問にする。これによって開発者自身による判断を行いやすくし、開発者ごとの診断基準の変動を抑制する効果が期待できる。例えば、「単体テスト技法の選択」という評価項目に対しては表 7 に示す 3 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。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。例えば、表7に示した質問事項に対してレベル1とレベル2にのみ当てはまる場合は、評価項目「単体テスト技法の選択」のスコアが3点満点中の2点と診断される。このような質問事項を、評価項目ごとに作成する。

### 3.4.1 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、それらに対してより効率良く質問事項を作成する必要がある。そこで、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。

#### 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。

(例) 単体テスト技法の選択

#### 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、作業を遂行するための具体的な手順を理解する必要があるもの。

(例) プロセスの設計

表6: タスク「パッケージのカスタマイズ方針の決定」が持つ評価項目

タスク	評価項目
パッケージのカスタマイズ方針の決定	カスタマイズ内容の決定
	アドオンの機能の決定
	パッケージとの連携の検討
	アドオンの機能との連携の検討
	アドオン機能に関する方式設計

表7: 評価項目「単体テスト技法の選択」に与える質問事項

レベル	質問事項
1	単体テスト技法の種類を知っている
2	単体テスト技法の選択を実施した経験がある
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

### 業務パターン

作業を遂行するために専門的な知識や手順を必要としないもの。

(例) 結合テストの実施結果の報告

### 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。

(例) ソフトウェア方式設計の実現可能性の評価

### 指導パターン

ツールや技術の利用方法の指導に関係するもの。

(例) 開発環境の利用方法の指導

質問パターンごとに、評価項目を診断するための複数の質問事項を作成する。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。質問事項は共通事項と個別事項に分かれており、共通事項は各パターンで同一の内容となり、個別事項は評価項目の内容ごとに変更される。それぞれの質問パターンが持つ質問事項を表8～12に示す。

表 8: 用語パターンが持つ質問事項

レベル	質問事項	
	個別事項	共通事項
1	作業対象	の知識を持っている
2	作業	の実施経験がある
3	作業	について他者からの評価を受けている

表 9: 作業パターンが持つ質問事項

レベル	質問事項	
	個別事項	共通事項
1	作業手順	の知識を持っている
2	作業	の実施経験がある
3	作業	について他者からの評価を受けている
4	作業	について他者への評価を行っている

### 3.4.2 評価項目の質問パターンへの割り当て

各評価項目を、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13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。例えば「単体テスト技法の選択」という評価項目の場合は、表13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。ただし、この方法によって用語パターンに割り当てられた評価項目の中には、作業対象がソフトウェア業界では一般に知られているものであるため、用語パターンへ割り当てるのは不適切と判断される項目が存在した。このような評価項目は例外と

表 10: 業務パターンが持つ質問事項

レベル	質問事項	
	個別事項	共通事項
1	作業	の実施経験がある
2	作業	について他者からの評価を受けている

表 11: 評価パターンが持つ質問事項

レベル	質問事項	
	個別事項	共通事項
1	確認事項	の自己評価を行っている
2	確認事項	について他者からの評価を受けている
3	確認事項	について他者への評価を行っている

表 12: 指導パターンが持つ質問事項

レベル	質問事項	
	個別事項	共通事項
1	ツールの利用方法	の知識を持っている
2	ツール	の利用経験がある
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

し、作業パターンへ割り当てる。例外処置をとる作業対象は、カスタマイズ、パッケージ、モジュール、ネットワーク機器、テストツールの5種類である。

### 3.4.3 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3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かったり、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3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3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

表 13: 各質問パターンに適した作業に関連する単語

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

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 3 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。この処理は、3.1 節で説明した対応によって役割ごとにタスクが分類されているために可能となっている。

## 4 提案手法のケーススタディ

本章では、A社の開発者を対象とした、対応を加えたiCDを用いた役割ごとのレベル評価の適用方法と適用結果について述べる。今回の適用では、表5に示す6種類の役割を対象とする。

### 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を参考にしているが、役割ごとに分類したタスクをA-SPICEと照らし合わせた結果、A-SPICEに記載されているいくつかの作業が不足していることが判明した。そこで、不足しているA-SPICE中の作業内容を新たなタスクとして追加することで、A社への適用に合わせたタスク群を作成する。例えば、ソフトウェア方式設計に関して、A-SPICEでは設計作業から検証作業までのプロセスが記載されているが、タスクディクショナリでは検証作業に関するタスクは存在していない。そこで、対応を加えたiCDに「ソフトウェア設計の検証」というタスクを新たに追加し、タスクに対する評価項目と質問事項をA-SPICEに基づいて作成する。

### 4.2 チェックシートと感想アンケートの作成

対応を加えたiCDをA社の開発者が利用できるように、チェックシートを作成する。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4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する。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る。アンケート内容の一覧を表14に示す。

### 4.3 適用方法

A社の開発者を対象とした役割レベル評価の適用方法を図5に示す。A社のある開発グループに属する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らう。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

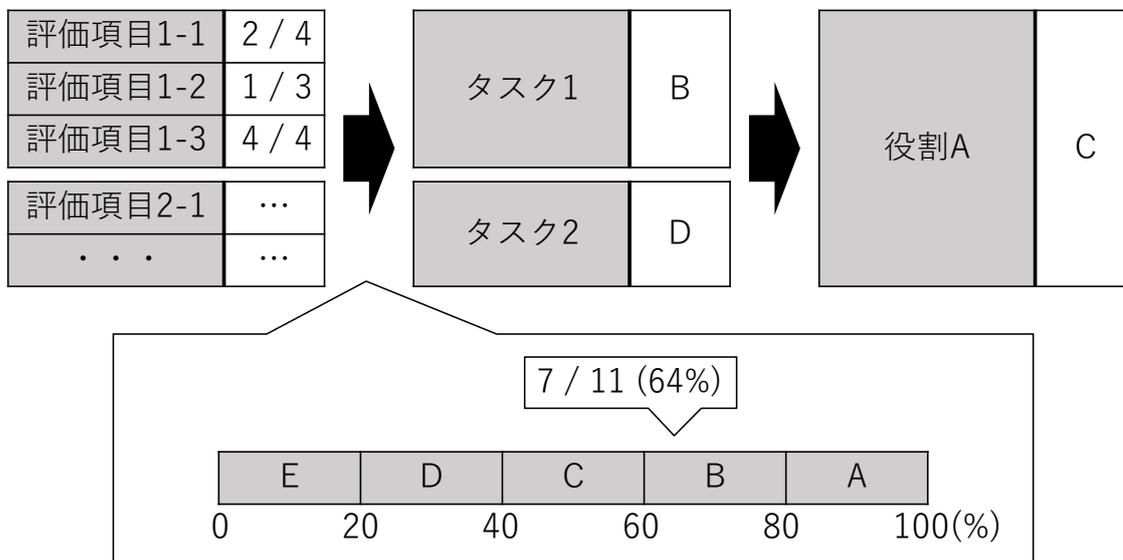


図 3: 質問事項への回答結果に基づいた評価レベル測定の例

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
<b>ソフトウェア要件分析者</b>				
<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
...				
<b>ソフトウェア方式設計者</b>				
<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 4: チェックシートの構成

表 14: 感想アンケートの内容

番号	質問内容	回答方式
1	質問にかかった時間は約何分ですか？	回答時間
2	質問量は適切でしたか？	「多い」から「少ない」の5段階
3	質問分類(最上段の5種類の項目)は回答しやすいものでしたか？	「回答しづらい」から「回答しやすい」の5段階
4	(3で「回答しづらい」「どちらかといえば回答しづらい」場合)どの項目が回答しづらいものでしたか？	自由回答
5	質問内容(各行の質問)の順序について、業務での作業順序と合っていますか？	「合っていない」から「合っている」の5段階
6	質問内容(各行の質問)は回答しやすいものでしたか？	「回答しづらい」から「回答しやすい」の5段階
7	(6で「回答しづらい」「どちらかといえば回答しづらい」場合)どのような点で回答しづらいと感じましたか？	自由回答
8	チェックシートの形式は利用しやすいものでしたか？	「利用しづらい」から「利用しやすい」の5段階
9	自身の能力を適切に評価してくれるようなチェックシートであると感じましたか？	「そう感じなかった」から「そう感じた」の5段階
10	9の回答について、どのような点で9の回答のよさを感じましたか？	自由回答
11	自身の役割ごとの開発スキルに興味がありますか？	「興味がない」から「興味がある」の5段階
12	この評価結果を通じてどのようなフィードバックを受けることが望ましいですか？(例：推奨セミナーの提示)	自由回答
13	その他気になった点がありましたら、ご自由にお書きください	自由回答

もらうことで、妥当性の確認を行う。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する。

#### 4.4 適用結果

A社の6名の開発者を対象とした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6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不足していることが分かる。また、図6に示した開発者に関する、「ソフトウェア要件分析者」の役割が持つタスクごとのレベルを図7に示す。この結果から、この開発者がソフトウェア要件分析者としての能力をより向上させるためには、アプリケーション共通基盤の設計に関する能力を高めればよいことが分かる。

次に、6名の開発者が持つ6種類の役割レベルの一覧を表15に示す。先程の図6,7に示した測定結果は開発者5のものである。表15を見ると、分析者や設計者といった役割よりも実装者やテスト担当者といった役割の方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を確認する。表14に示す質問のうち、自由回答以外の回答結果を表16に示す。質問1はチェックシートの回答に要した時間が回答されている。他の質問は5段階の回答が行われるものであり、肯定的な回答を5、否定的な回答を1としている。3の場合は「どちらともいえない」という回答となる。表16を確認すると、まず、質問1によって各開発者の回答時間が最短で30分、最長では120分を要したということが分かる。1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実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

表 15: A社の6名の開発者が持つ役割レベル

	開発者					
	1	2	3	4	5	6
ソフトウェア要件分析者	D	D	D	D	B	B
ソフトウェア方式設計者	D	D	D	D	C	B
詳細設計者	D	E	D	D	D	B
コーダー(実装者)	D	C	C	C	C	B
単体テスト担当者	D	E	C	D	C	B
ソフトウェアテスト担当者	C	D	D	C	C	B

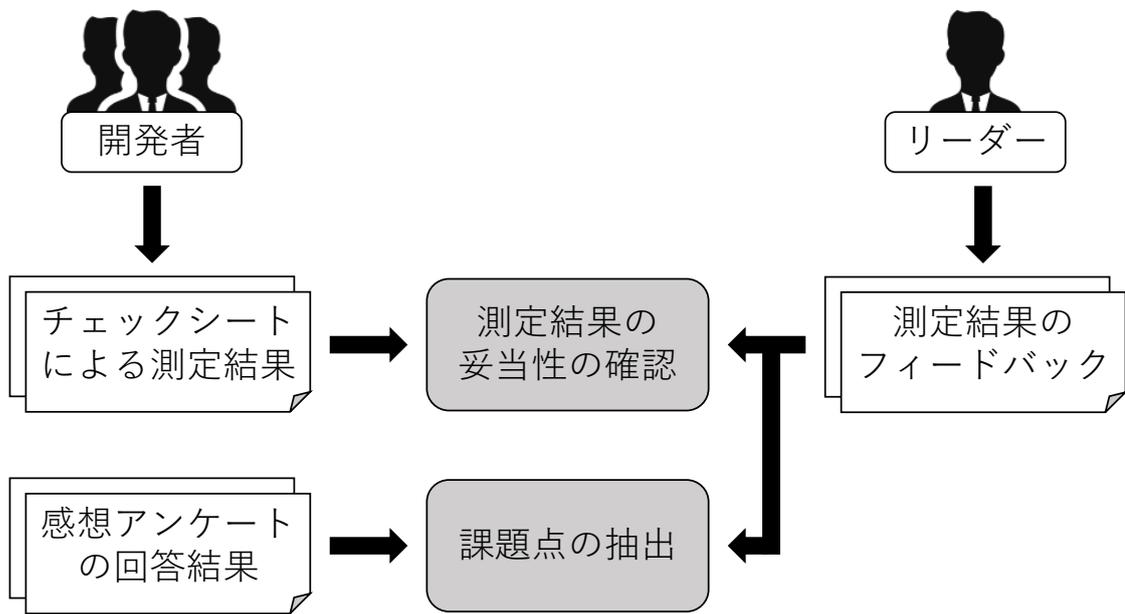


図 5: 役割レベル評価の適用方法

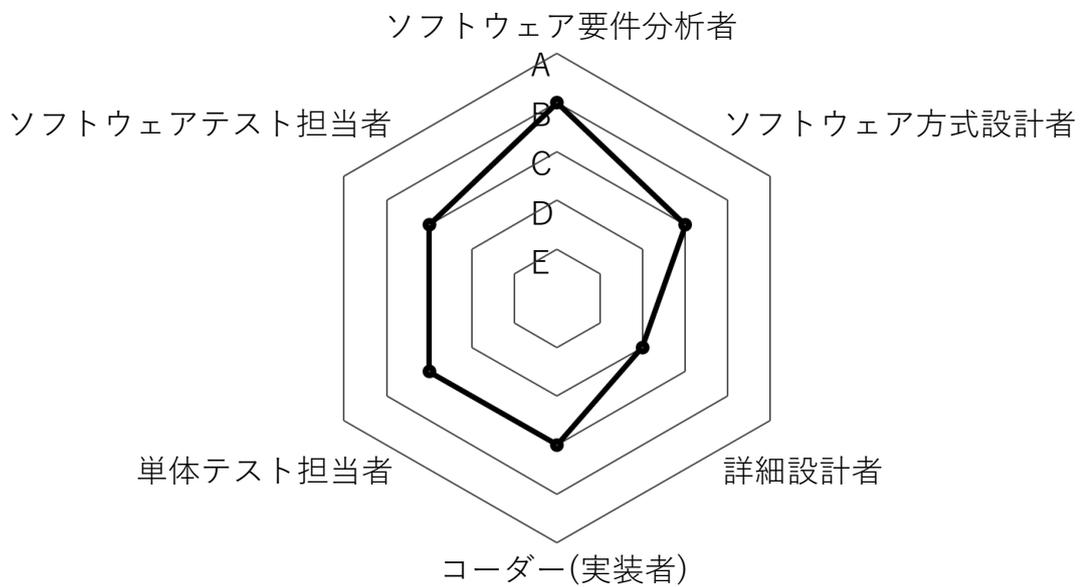


図 6: ある開発者の 6 種類の役割レベル

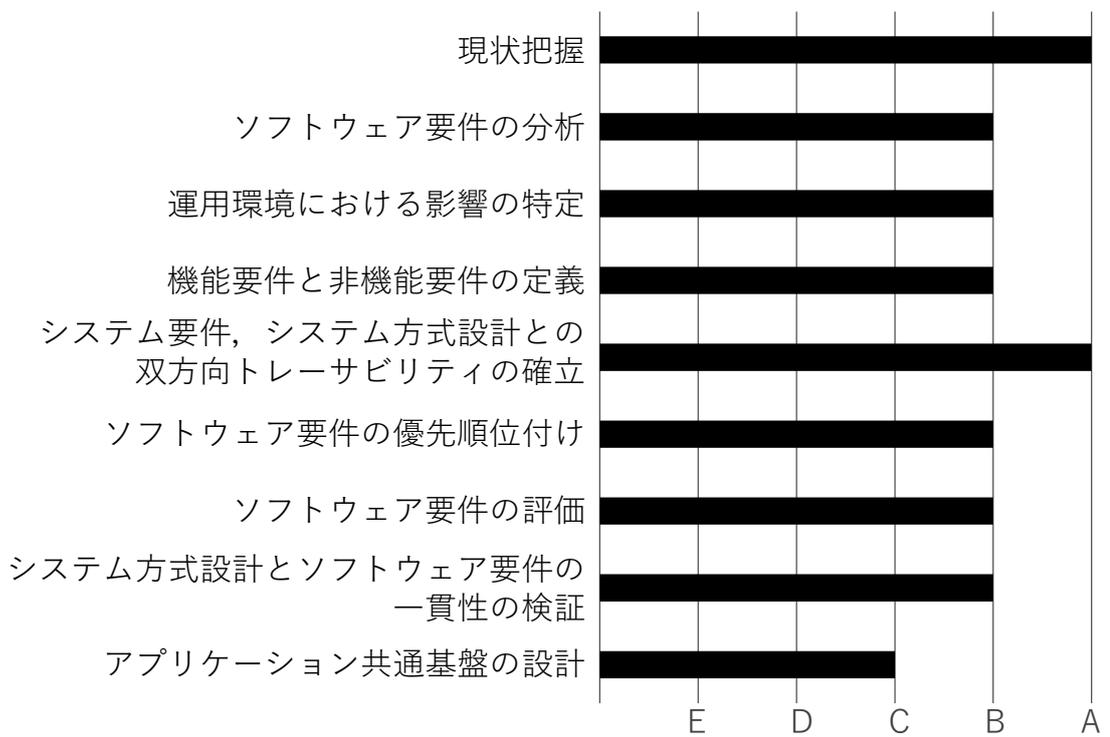


図 7: ある開発者の役割「ソフトウェア要件分析者」に関するタスクレベル

なる。他の質問に関しては、質問2と質問3が否定的な回答のみであり、最優先で改善すべき内容となっている。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。

表 16: 感想アンケートの回答結果

	開発者					
	1	2	3	4	5	6
質問 1	30分	60分	120分	70分	40分	30分
質問 2	1	2	2	2	1	2
質問 3	1	1	2	2	2	2
質問 5	4	4	2	4	4	3
質問 6	2	3	2	2	3	3
質問 8	5	4	2	2	2	1
質問 9	3	3	1	2	3	2
質問 11	4	3	4	5	3	4

## 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。

### 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15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う。

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う。各開発者の経験年数を表17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年以上でグループリーダー等を担当している社員である。表15と表17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。

次に、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する。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、詳細設計者やコーダー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、対応を加えたiCDを用いて得られた測定結果は正しいという意見が得られた。これらのことから、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、対応を加えたiCDを用いて測定することができているといえる。

### 5.2 測定方法の課題と対策

提案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作成したチェックシートに対する感想アンケートの結果を踏まえて、以下の4つの課題を抽出した。

表 17: A社の開発者の経験年数

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

**課題 1** 回答時間が長くかかりすぎている。最短でも 30 分、長ければ 120 分の回答時間を要している。

**課題 2** 対象グループが普段携わらない分野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。例えば、Web 開発分野には携わっていないため、Web 開発に関する設計などの質問に回答できない、というような問題が生じた。

**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。例えば、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 と課題 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。

### 5.2.1 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。いずれの部署やグループ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、全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。必須タスクと選択タスクへの分類と、部署ごとのタスク選別の例を表 18 に示す。この場合、部署 A に所属する開発者はタスク 1, 2, 5 に関する質問事項にのみ回答すればよく、部署 B に所属する開発者はタスク 1, 2, 3 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繋げられる。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。

### 5.2.2 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。表8～表12に示す質問事項は、以下の5種類にまとめられる。

- (個別事項)の知識を持っている
- (個別事項)の自己評価を行っている
- (個別事項)の実施経験または利用経験がある
- (個別事項)について他者からの評価を受けている
- (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

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、質問事項の意図の明記と、一部の質問内容の変更を行う。

#### (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。

表 18: 必須タスクと選択タスクの選別の例

タスク	分類	部署 A	部署 B
タスク 1	必須	○	○
タスク 2	必須	○	○
タスク 3	選択	-	○
タスク 4	選択	-	-
タスク 5	選択	○	-

#### (個別事項)の自己評価を行っている

この質問事項は、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。この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

この質問事項は、上司等からの指示やアドバイスを受けているかどうかは関係無く、作業などの個別事項に対する実施・利用をひと通り経験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。この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、作業などの個別事項に対する実力の有無を確認したいという意図から、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る。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、評価パターンに属する評価項目の場合は、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る。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か

詳細な説明として、「対象に関する講義を行ったり、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。

### 5.2.3 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レベル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。例えば、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1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1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が共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、各開発者が持つ役割レベルに関するより詳細な比較が可能となる。また、従来の評価では把握しづらかった、各開発者の持つ突出した能力の有無を確認することも容易となる。

## 6 あとがき

本研究では、ソフトウェア開発者が持つ役割ごとのレベル評価を実施するために、iCDによる開発能力評価に対して対応を加えた。対応の内容としては、iCDが列挙するタスクの役割ごとの整理と、タスク項目数の削減を実施した。また、iCDを用いた際の診断基準の変動を抑制するために、開発者の知識や経験の有無といった自身で判断しやすい質問事項を作成した。さらに、これらの対応を加えたiCDに基づいたチェックシートを用いて、A社を対象とした役割ごとのレベル評価を実施した。その結果、各開発者の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、対応を加えたiCDを用いれば測定することができた。

今後の課題としては、チェックシートの改善とシステム化があげられる。作成したチェックシートには、質問内容やレベル評価方法に関するいくつかの課題点が存在している。それらに対して対策を加え、再び開発者への適用とフィードバックを実施することで、チェックシートの改善を行っていく必要がある。さらに、チェックシートに基づいた評価支援ツールを作成する。これによって、回答の効率化やデータ蓄積による利便性の向上を可能にし、実企業への適用可能性を上げることが期待できる。

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今後の課題としてあげられる。そのために、今回提供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が考えられる。また、チェックシートと他の性格診断テストを併用し、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といった方法も考えられる。

## 謝辞

本研究の全過程を通し，理解ある親身なご指導を賜り，的確なご助言を頂きました 楠本 真二 教授に心より感謝申し上げます。

本研究を行うにあたり，的確なご指導およびご助言を頂きました 肥後 芳樹 准教授に深く感謝申し上げます。

本研究を行うにあたり，丁寧かつ適切なお指導を頂きました 松本 真佑 助教に深く感謝申し上げます。

本研究を行うにあたり，多大なご助言を頂きました 折方 孝夫 様，塚本 貴弘 様，藤原 永年 様に心より感謝申し上げます。

研究生活の中で様々な相談に乗って頂き，また励まして頂きました 大阪大学大学院情報科学研究科コンピュータサイエンス専攻博士前期課程2年 下仲 健斗 氏，同 中島 弘貴 氏，同 山本 将弘 氏に深く感謝申し上げます。

研究室生活を大変豊かにして頂きました 大阪大学大学院情報科学研究科コンピュータサイエンス専攻博士前期課程1年 有馬 涼 氏，同 佐々木 美和 氏，同 谷門 照斗 氏，同 内藤 圭吾 氏，同 松尾 裕幸 氏，同 山田 涼太 氏に深く感謝申し上げます。

研究室の環境維持に多くのご助力を頂きました 大阪大学基礎工学部情報科学科4年 田中 紘都 氏，同 土居 真之 氏，同 松本 惇之介 氏，同 林 純一 氏に深く感謝申し上げます。

最後に，本研究に至るまでに，講義，セミナー等でお世話になりました大阪大学大学院情報科学研究科の諸先生方に，この場を借りて心より御礼申し上げます。

## 参考文献

- [1] David Klappholz, Lawrence Bernstein, and Daniel Port. Assessing attitude towards, knowledge of, and ability to apply, software development process. In *Software Engineering Education and Training, 2003.(CSEE&T 2003). Proceedings. 16th Conference on*, pp. 268–278. IEEE, 2003.
- [2] R. Jensen. Don't forget about good management. *CrossTalk Magazine*, Vol. 30, , 2000.
- [3] Deborah H. Stevenson and Jo Ann Starkweather. PM critical competency index: IT execs prefer soft skills. *International Journal of Project Management*, Vol. 28, No. 7, pp. 663–671, 2010.
- [4] José Gamaliel Rivera-Ibarra, Josefina Rodríguez-Jacobo, and Miguel Angel Serrano-Vargas. Competency framework for software engineers. In *Software Engineering Education and Training (CSEE&T), 2010 23rd IEEE Conference on*, pp. 33–40. IEEE, 2010.
- [5] Lawrence Peters. A knowledge & competencies checklist for software project management success. In *SEKE*, pp. 241–243, 2014.
- [6] Mahsa Taghi Zadeh, Reza Dehghan, Janaka Y. Ruwanpura, and George Jergeas. 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.
- [7] Manuel Wiesche and Helmut Krcmar. The relationship of personality models and development tasks in software engineering. In *Proceedings of the 52nd ACM conference on Computers and people research*, pp. 149–161. ACM, 2014.
- [8] 白井久美子, 越島一郎, 梅田富雄. 改革プログラム遂行時の人的資源に関する基本的考察. 国際 P2M 学会誌, Vol. 7, No. 1, pp. 147–162, 2012.
- [9] iCD オフィシャルサイト. <https://icd.ipa.go.jp/icd/>.
- [10] IPA 独立行政法人情報処理推進機構. <https://www.ipa.go.jp/index.html>.
- [11] 石川拓夫. 2万人のレベル診断～日立のIT人財強化の取り組み～. 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.
- [12] Zulkefli Mansor and Norsaedah HazimahRedzuan. Agile project manager competencies in software development projects : A systematic literature review. In *WSEAS*, 2015.

- [13] James D. Klein, J. Michael Spector, Barbara Grabowski, and Ileana de la Teja. *Instructor competencies: Standards for facetoface, online, and blended settings*. IAP, 2004.
- [14] A.A. Sánchez, J.G.T. Ronda, and R.S. Sánchez. *Fundamentos de dirección y gestión de recursos humanos*. Thomson, 2004.
- [15] Parisa Ghazi, Ana M. Moreno, and Lawrence Peters. Looking for the Holy Grail of software development. *IEEE Software*, Vol. 31, No. 1, pp. 96–96, 2014.
- [16] Peter E.D. Love, David J. Edwards, and Zahir Irani. Forensic project management: An exploratory examination of the causal behavior of design-induced rework. *IEEE Transactions on Engineering Management*, Vol. 55, No. 2, pp. 234–247, 2008.
- [17] Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, Vol. 33, No. 2004, pp. 1–26, 2004.
- [18] Daniel Varona, Luiz Fernando Capretz, Yadenis Piñero, and Arif Raza. Evolution of software engineers' personality profile. *ACM SIGSOFT Software Engineering Notes*, Vol. 37, No. 1, pp. 1–5, 2012.
- [19] Khaled El-Emam and Inigo Garro. ISO/IEC 15504. *International Organization for Standardization*, 1999.
- [20] Khaled El-Emam and Andreas Birk. Validating the ISO/IEC 15504 measure of software requirements analysis process capability. *IEEE transactions on Software Engineering*, Vol. 26, No. 6, pp. 541–566, 2000.
- [21] Béatrix Barafort, Bernard Di Renzo, and Olivier Merlan. Benefits resulting from the combined use of ISO/IEC 15504 with the Information Technology Infrastructure Library (ITIL). In *International Conference on Product Focused Software Process Improvement*, pp. 314–325. Springer, 2002.
- [22] S.I.G. Automotive. Automotive SPICE-Process Assessment Model (2007). *Website: [http://www.automotivespice.com/fileadmin/softwaredownload/automotiveSIG\\_PAM\\_v25.pdf](http://www.automotivespice.com/fileadmin/softwaredownload/automotiveSIG_PAM_v25.pdf)* (Stand 01.09. 2014).
- [23] Christian Kreiner, Richard Messnarz, Andreas Riel, Damjan Ekert, Michael Langgner, Dick Theisens, and Michael Reiner. Automotive knowledge alliance AQUA–integrating automotive SPICE, six sigma, and functional safety. In *European Conference on Software Process Improvement*, pp. 333–344. Springer, 2013.

- [24] Richard Messnarz, Hans-Leo Ross, Stephan Habel, Frank König, Abdelhadi Koundoussi, Jürgen Unterreitmayer, and Damjan Ekert. Integrated Automotive SPICE and safety assessments. *Software Process: Improvement and Practice*, Vol. 14, No. 5, pp. 279–288, 2009.
- [25] Fabrizio Fabbrini, Mario Fusani, Giuseppe Lami, and Edoardo Sivera. Integrating joint reviews with automotive SPICE assessments results. In *International Conference on Software Process*, pp. 357–368. Springer, 2008.
- [26] 共通フレーム 2013 ～経営者、業務部門とともに取組む「使える」システムの実現～. 情報処理推進機構, 2013.
- [27] Pierre Bernard. *Foundations of ITIL® 2011 Edition*. Van Haren, 1970.
- [28] 松本吉弘. ソフトウェアエンジニアリング基礎知識体系-SWEBOK V3.0. 株式会社 オーム社, 2014.
- [29] 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.
- [30] 経済産業省. IT スキル標準, 2002.