

## 文書診断法の診断観点と診断水準における コーディング規約の分析

藤田 悠<sup>†1</sup> 山本 雅基<sup>†1,†4</sup> 中澤 達夫<sup>†1,†5</sup> 塩谷 敦子<sup>†2</sup>  
池田 貴一<sup>†3</sup> 小野 伸幸<sup>†1</sup> 楡井 雅巳<sup>†1</sup>

我々は、ソフトウェア開発プロセスの各作業工程で出力される開発文書を中間成果物として捉えており、その開発文書の品質を計測するための研究を行っている。開発文書を評価するために、記述の明晰さと、開発に十分な緻密さと、技術内容の適切さに分ける「診断観点」を定義した。次に、文書で指摘した問題点が表記上の問題であるか、意味上の問題であるかの違いで分ける「診断水準」を定義した。これらの定義に従って、文書を精読して問題のある記述を「診断種別」ごとに分けられた「診断項目」に分類する文書診断法を提案した。

本稿では、文書診断法の指標を用いてコーディング規約の基本概念として与えられたコーディングの「作法」を分析するために、「作法」を構成する「作法詳細」を「診断観点」と「診断水準」に対応づけた。結果として、コーディングの「作法」には、プログラムを作業工程における中間成果物の開発文書として評価するための性質があることを見出した。その結果を踏まえ、要求仕様書や設計書などの開発文書に加えて、プログラムを含むすべての中間成果物を評価する指標として「診断観点」と「診断水準」を提案する。

### Analysis of Coding Rules on Diagnosis Perspective and Diagnosis Level of Document Diagnostic Method

YUTAKA FUJITA<sup>†1</sup> MASAKI YAMAMOTO<sup>†1,†4</sup>  
TATSUO NAKAZAWA<sup>†1,†5</sup> ATSUKO SHIOYA<sup>†2</sup>  
KIICHI IKEDA<sup>†3</sup> NOBUYUKI ONO<sup>†1</sup> MASAMI NIREI<sup>†1</sup>

We think about any development documents including program as interim output written in each process, and are studying about the measurement of the quality of software development documents. We have proposed document diagnostic method inspecting problems on software development documents and classify these problems based on 'diagnosis perspective' and 'diagnosis level'. The 'diagnosis perspective' for software development documents, consists of three perspectives, 'technical writing', 'software development' and 'software technique'. The 'diagnosis level' consists of 'description' and 'explanation'.

In this paper, we relate coding manner to 'diagnosis perspective' and 'diagnosis level' for analyzing coding manner using index of document diagnosis method.

Consequently, coding manner has characteristics to measure program as development document of interim output in process. Hence, we propose to use 'diagnosis perspective' and 'diagnosis level' to evaluate development documents.

#### 1. はじめに

JIS X 0129-1<sup>1)</sup>では、ソフトウェア製品の品質に関わる6特性として、信頼性、保守性、移植性、効率性、機能性、使用性を規定している。しかし、ソフトウェア開発の作業工程が出力する中間成果物である開発文書の品質に関しては、決め手となる品質特性は定められていない。

開発文書の品質を測る方法として、開発文書の量を参考値と比較する方法と、開発文書の記述内容を精査する方法がある。

『組込みソフトウェア開発向け 品質作り込みガイド』<sup>2)</sup>は、まず、組込みシステムの品質という特質を計測し評価するための指標として、ソフトウェア開発の過程で実施される作業を評価するための「プロセス品質評価指標」と、開発の過程で作成される成果物の品質を評価する「プロダクト品質評価指標」を定義している。次に、これら2種類の品質評価指標を算出する際の基礎になる、(1)ソースコードボリューム基礎指標、(2)ドキュメントボリューム基礎指標、(3)工数ボリューム基礎指標、(4)テストボリューム基礎指標、の4種類の基礎指標を定義している。さらに、4種類の基礎指標にはそれぞれ、ソースコードの行数や設計書のボリュームなどの具体的な基礎指標が定義されている。これらの基礎指標を基にして算出される品質評価指標には参考値が与えられており、それを参考にして成果物の量としての側面を評価する。しかし、これらは限られた側面からみた開発文書の品質を表しているに過ぎない。

他方、開発文書の記述内容を精査するために、竹内らは自然言語処理技術を用いて問題点を抽出する手法を提案しており<sup>3)</sup>、Arthurらは文書に求められる品質特性に基づいて品質を測定するための指標を提案している<sup>4)</sup>。また、藤田らは、要求定義工程や設計工程などの作業結果が適切に開発文書に表されているか計測する文書診断法を提案している<sup>5)</sup>。文書診断法は、主に自然言語で記述された開発文書を評価する手法であり、「診断観点」として記述の明晰さと開発文書としての緻密さに分け、「診断水

<sup>†1</sup> 長野工業高等専門学校  
Nagano National College of Technology

<sup>†2</sup> 合同会社 イオタクラフト  
IOTACRAFT,LLC.

<sup>†3</sup> 株式会社 ミマキエンジニアリング  
MIMAKI ENGINEERING Co.,Ltd.

<sup>†4</sup> 名古屋大学  
Nagoya University

<sup>†5</sup> 信州大学  
Shinshu University

準」として文書の表記上の問題であるか意味上の問題であるかに分けて、文書の内容を精読して問題点を指摘している。藤田らは企業現場の要求仕様書や設計書を分析した結果、5種類の「診断種別」に類別された71項目の「診断項目」から文書を評価している。しかし、「診断観点」と「診断水準」がソフトウェア開発プロセスの全体にわたる様々な開発文書の品質を表し得るかについては十分に検証されていない。

我々は、ソフトウェア開発プロセスの各作業工程で出力される開発文書を中間成果物と捉えている。そのため、要求定義書や設計書に加えて、プログラミング言語で表されたプログラムも中間成果物と位置づけている。

プログラムに対して、「組込みソフトウェア開発向けコーディング作法ガイド Ver1.1」<sup>7)</sup> (Embedded System development Coding Reference) (以降、ESCR) は、コーディング規約を「品質を保つために守るべきコードの書き方」と捉えて、コーディング規約の基本概念を「作法」として定義し、「作法」に関連するコーディングの具体的な決めごとである「ルール」の参考例を提示している。ESCRでは、ソフトウェアの品質特性を基にしてプログラミング言語に依存しない「作法」と、「作法」ごとにC言語の命令文や表記法に言及した「ルール」を定めている。したがって、「作法」では、プログラムをソフトウェア製品としての側面から評価することを目的としているので、実装工程が出力する開発文書としての側面から評価することを想定していない。

本稿では、要求仕様書や設計書を評価するために与えた「診断観点」と「診断水準」を用いて、ESCRが定めた「作法」を分析する。分析の結果から「作法」が開発文書における品質のどの側面を評価するかを考察する。これにより、ソフトウェア開発の作業工程で作成される全ての成果物の品質を評価する手がかりを得る。

第2章では、作業工程が出力する種類の異なる成果物を評価するための2種類の指標を示す。第3章では、これら2種類の指標を対比させて、文書診断法で用いている開発文書を評価するための指標に「作法」を対応づける。第4章では、分類した結果に基づいて「作法」の位置づけを明らかにする。

## 2. ソフトウェア開発の作業工程の成果物を評価する指標

ソフトウェア開発プロセスの基本的なモデルであるソフトウェア開発V字プロセスモデル<sup>8)</sup>は、実装工程がV字の最下部に位置し、その左側に3種類のいわゆる上流工程が、右側に3種類の下流工程が配置される。上流工程は要求定義とアーキテクチャ設計と詳細設計からなり、下流工程は単体テストと結合テストと総合テストの工程からなる。各作業工程では、直前の工程の成果物を入力し、自工程の作業を行い、その成果を出力する。出力された成果物は次工程の入力となる。例えば、要求定義工程は、作業工程の成果物として要求定義書を出し、これはアーキテクチャ設計工程の入力となる。アーキテクチャ設計工程は、要求定義書を入力し、アーキテクチャの設計を

行い、アーキテクチャ設計書を出し、

本章では、ソフトウェア開発の作業工程の中間成果物を評価するための2種類の指標を示す。2.1節では自然言語で表された開発文書を対象とした指標、2.2節ではプログラミング言語で表されたプログラムを対象とした指標を示す。

### 2.1 自然言語で表された文書を評価する指標

我々は、主に自然言語で表されたソフトウェア開発文書の記述に問題がないかを精査する手法として文書診断法を提案した<sup>5)</sup>。文書診断法は、開発文書を精読し、その文書に含まれる問題点を指摘する手法であり、指摘した問題点を基に文書を評価する。

本節では、自然言語で表された開発文書の品質の構成を提案し、品質に強く関連する文書診断法の指標を与える。2.1.1項に、提案する開発文書の品質の構成を示す。2.1.2項に、先行研究<sup>5)6)</sup>で与えた2観点に1観点を加えた合計3観点からなる「診断観点」と、2水準からなる「診断水準」と、5種類からなる「診断種別」と、71項目からなる「診断項目」を示す。

#### 2.1.1 開発文書の品質

我々は、開発文書の品質を、記述が明晰かつ緻密であるかを評価する「文書化品質」と、文書に表される技術内容を評価する「技術要素品質」に分けることを提案する。これらに分けることで、該当の開発文書には記述すべき内容を理解したうえで間違いなく伝えるための記述が不十分であるのか、または専門的な技術内容が不十分であるのか、などの原因を分析し得る。

#### 2.1.2 文書診断法の指標

開発文書の品質に深く関連する、文書を診断するための3観点からなる「診断観点」と2水準からなる「診断水準」を与え、指摘して問題の種類を分類するために、5種類からなる「診断種別」と71項目からなる「診断項目」を与えた。

##### (1) 診断観点

技術文書全般に共通する、明晰に記述されているかを診断するために「技術文書観点」を与える。作業工程の成果物に資する緻密な記述がされているかを診断するために「開発文書観点」を与える。構築する文書の目的に適したソフトウェア技術の内容が適切に記述されているかを診断するために「ソフトウェア技術観点」を与える。これら3観点では、技術文書一般に共通する観点が下層に位置し、ソフトウェア開発に特化する観点が上層に位置する3層を構成し、「技術文書観点」、「開発文書観点」、「ソフトウェア技術観点」の順番に第1層、第2層、第3層に位置付ける。図1に3層の構造を示す。ソフトウェア開発文書を診断の対象としたとき、「文書化品質」には、「技術文書観点」と「開発文書観点」が関連する。さらに、「技術要素品質」には、「ソフトウェア技術観点」が関連する。ただし、本稿では「文書化品質」を計測の対象としているので、「技術要素品質」とそれに関連する「ソフトウェア技術要素」については立ち入らない。

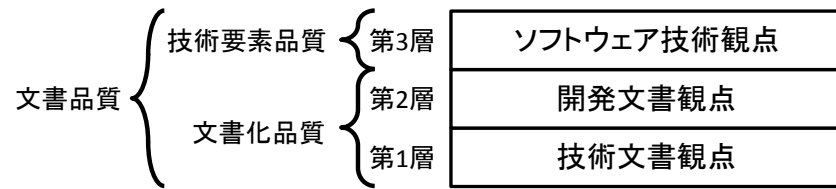


図1 ソフトウェア開発文書を対象としたときの文書品質と診断観点の構造

### (2)診断水準

文書を精査して指摘した問題点が、表記の問題であるか意味の問題であるかによって、「表記水準」と「意味水準」の2水準に分類する「診断水準」を与えた<sup>6)</sup>。

### (3)診断種別

「技術文書観点」と「開発文書観点」と、「表記水準」と「意味水準」から文書の診断をして指摘した問題点を種類ごとに分けるために、5種類の「診断種別」(a)表記ルール、(b)文・文章の表現、(c)図表の表記、(d)文書構造・構成・展開、(e)文書体系、を与えた。

### (4)診断項目

我々がこれまでに実施した文書診断の結果を基にして、「診断種別」ごとに問題点を内容別に分けて「診断項目」として与えた。例えば、「技術文書観点」の「診断種別」である「表記ルール」に含まれる「診断項目」は(a)誤字・脱字、(b)長熟語、(c)用字・用語の不統一、(d)記号の使用が不適切、(e)意味的な誤用、(f)一語一義に解釈できない、(g)使い分け不備、(h)用語定義なし、の8項目である。同様に、「文・文章の表現」には、14項目、「図表の表現」には11項目、「文書構造・構成・展開」には13項目、「文書体系」には2項目が与えられた。さらに、「開発文書観点」には、「表記ルール」には3項目、「文・文章の表現」には7項目、「図表の表現」には1項目、「文書構造・構成・展開」には10項目、「文書体系」には2項目が与えられた。「診断観点」と「診断種別」ごとの「診断項目」の件数を表1に示す。

診断種別	技術文書観点	開発文書観点
表記ルール	8	3
文・文章の表現	14	7
図表の表現	11	1
文書構造・構成・展開	13	10
文書体系	2	2

## 2.2 プログラミング言語で表されたプログラムを評価する指標

プログラムの品質を保つためにコードの書き方を定めたコーディング規約<sup>9)10)</sup>がある。また、「組込みソフトウェア開発向けコーディング作法ガイド Ver1.1」<sup>7)</sup> (ESCR)では、ソフトウェア製品に関わる品質特性を基に、コーディング規約の基本概念が「作法」としてまとめられており、具体的な決めごとである「ルール」が「作法」に関連づけられている。本節では、ESCRが定めている、ソフトウェア製品に関わる品質特性に基づく「作法概要」と「作法詳細」と「ルール」を説明する。

### 2.2.1 プログラムの品質特性

ESCRでは、JIS X 0129-1<sup>1)</sup>が定める6特性からなるソフトウェア製品の品質特性、信頼性、保守性、移植性、効率性、機能性、使用性、のうち、機能性と使用性については実装より前の工程で作り込む性質であるので、実装の工程の成果物であるプログラムには、信頼性、保守性、移植性、効率性の4特性が深く関わると述べられている。これら4特性に関連づけて「作法」を定義しているため、それぞれの「作法」がどのようなソフトウェア製品の品質に強く影響しているかが明らかにされている。

### 2.2.2 ESCRの指標

ESCRでは、「作法」として11件の「作法概要」と各作法概要を詳細化した51件の「作法詳細」が階層化して定められている。「作法概要」はプログラミング言語に依存せず、「作法詳細」は一部の作法ではC言語特有の命令文や表記法について言及するが基本的にはプログラミング言語に依存しない。さらに、51件の各作法詳細には、C言語の命令文や表記法などに言及した具体的な「ルール」が合計129件定められている。

#### (1)「作法概要」

「作法概要」には「領域は初期化し、大きさに気を付けて使用する。」「データは範囲、大きさ、内部表現に気を付けて使用する。」「プログラムはシンプルに書く。」などの概念的な「作法」が、品質特性に関連付けられて全11件が定められている。11件の「作法概要」を表2の第1列に示す。

#### (2)「作法詳細」

概念的に定められている「作法概要」を詳細化して、より具体的に表した「作法詳細」が定められている。例えば、作法概要「プログラムはシンプルに書く。」に対しては、「構造化プログラミングを行う。」「1つの文で1つの副作用とする。」「目的の違う式は、分離して記述する。」「複雑なポインタ演算は使用しない。」の全4件の「作法詳細」が定められている。同様に、他10件の「作法概要」それぞれに「作法詳細」が定められており、全11件の「作法概要」に定められた「作法詳細」は51件である。11件の「作法概要」それぞれに属する「作法詳細」の件数を表2の第2列に示す。

#### (3)「ルール」

51件の各作法詳細には、C言語の命令文や表記法などに言及した具体的な「ルール」

が合計 129 件与えられている。

例えば、作法概要「プログラムはシンプルに書く。」に定められている 4 件の「作法詳細」の内、「1つの文で1つの副作用とする。」に対しては「コンマ式は使用しない。コンマ式は for 文の初期化式や更新式以外では使用しない。」「1つの文に、代入を複数記述しない。ただし、同じ値を複数の変数に代入する場合を除く。」の 2 件のルールが定められている。その他、「構造化プログラミングを行う。」に対しては 5 件、「目的の違う式は、分離して記述する。」に対しては 3 件、「複雑なポインタ演算は使用しない。」に対しては 1 件のルールが作法詳細に対して定められている。このようにして、作法概要「プログラムはシンプルに書く。」には合計 11 件のルールが定められている。同様に、作法詳細に対して定められたルールを全て合わせると合計 129 件になる。各作法詳細に対するルールの件数を「作法概要」ごとに表 2 の第 3 列に示す。

表 2 ESCR V1.1 「作法概要」ごとの「作法詳細」と「ルール」の件数一覧

作法概要	作法詳細件数	ルール件数
領域は初期化し、大きさに気を付けて使用する。	3	7
データは範囲、大きさ、内部表現に気を付けて使用する。	8	21
動作が保証された書き方にする。	6	12
他人が読むことを意識する。	12	29
修正し間違えないような書き方にする。	2	6
プログラムはシンプルに書く。	4	11
統一した書き方にする。	7	20
試験しやすい書き方にする。	2	4
コンパイラに依存しない書き方にする。	5	12
移植性に問題のあるコードは局所化する。	1	3
資源や時間の効率を考慮した書き方にする。	1	4

### 3. 文書診断法の指標への ESCR の対応づけ

プログラムをソフトウェア開発の作業工程が出力する開発文書とみなしたとき、自然言語で表された開発文書の評価をするために我々が提案している文書診断法から見ると、ESCR の指標がどのように位置づけられるか検討する。

#### 3.1 文書診断法の指標と ESCR の指標の関係

ESCR では、ソフトウェア製品の品質特性を基に「作法概要」と「作法詳細」を階層化し、それぞれの作法詳細に「ルール」が提示されている。「作法概要」はプログラミング言語に依存せず、「作法詳細」は C 言語特有の表記法について一部言及してい

るが基本的にプログラミング言語に依存せず、ルールはプログラミング言語に依存するものとして定められている。

文書診断法では、「診断観点」と「診断水準」は特定の言語に依存しない。また、「診断項目」は自然言語特有の記述に関して言及しているが、ESCR の「ルール」程の具体的な言及はされていない。また、「診断種別」は、「診断項目」を種類ごとにまとめるために設けられている。

これらの指標のつながりと抽象度を基にして、文書診断法と ESCR の指標について指標内部の関係と、互いに対比させた関係を図 2 に示す。抽象度の高い指標が上部に位置し、低い指標が下部に位置する。四角形はそれぞれの指標の集合を表し、角丸四角形は指標の集合に含まれる要素を表す。実線は異なる集合の要素のつながりを表し、点線は言語依存の境界を表す。境界線より上には言語に依存しない指標が位置しており、下には言語に依存する指標が位置している。

文書診断法と ESCR の指標の位置づけと関連性を観察すると、ESCR では抽象度が高い領域から低い領域まで一貫して関連付けられている。他方、文書診断法では「診断項目」からそれぞれの指標につながるがある。

我々は、言語依存の境界線上にある文書診断法の「診断項目」と ESCR の「作法詳細」が同程度に具体化されている指標であると捉え、「診断項目」が「診断観点」と「診断水準」に対応づけているように、「作法詳細」を「診断観点」と「診断水準」に対応づけた。

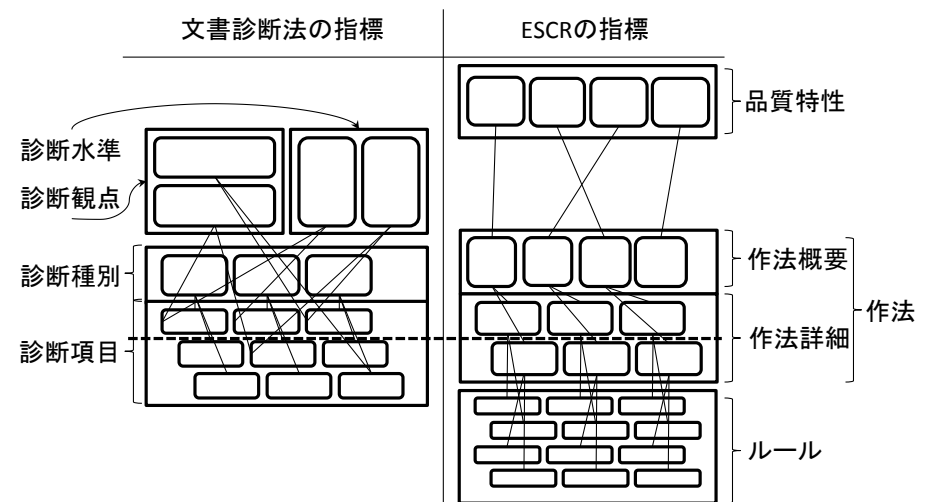


図 2 文書診断法と ESCR の指標の比較

### 3.2 診断観点と診断水準への ESCR の対応づけ

コーディングの「作法」は、プログラムの記述に関する基本概念であるので、開発文書の品質のうち「文書化品質」に該当する。そのため、ESCR の「作法詳細」は「技術文書観点」または「開発文書観点」どちらかに対応づけられる。そこで、「作法詳細」に用いられている C 言語特有の用語を自然言語で用いられている用語に置きかえたときに、技術文書一般にあてはまるとき「技術文書観点」を、開発文書が必要とする緻密な記述にあてはまるとき「開発文書観点」を対応づけた。

次に、「文書診断観点」の「診断水準」に従って、ESCR の「作法詳細」を「表記水準」または「意味水準」に対応づけた。「作法詳細」が、プログラムの表記に関する場合には「表記水準」を、プログラムの意味内容に関する場合には「意味水準」を対応づけた。

例として、作法概要「プログラムはシンプルに書く」に定められている 4 件の「作法詳細」の対応づけを示す。「構造化プログラミングを行う」では、論旨に合わせて文書の構造を決めて記述することに関連するので「技術文書観点」とした。また、構造を決めるには、記述の内容に関わるため「意味水準」とした。「1つの文で1つの副作用とする」では、「技術文書観点」の診断項目「一文一義でない」に関連して、ひとつの文にはひとつの事を記述するべきであることを示唆しているので「技術文書観点」の「意味水準」とした。「目的の違う式は、分離して記述する」では、その章や節や段落に、論題に関連のない事柄を分けて記述しなければ論旨が分かりにくくなるという、技術文書にも当てはまる内容であるので「技術文書観点」とした。また、目的の違いを理解しなければ分離する部分を区別できないので「意味水準」とした。「複雑なポインタ演算は使用しない」では、指示語が続くことで指し示す部分が分かりにくくなることを防ぐためであると考えられるので「技術文書観点」とした。また、ポインタという表記が安易に使われていることについては内容に入り込まずに指摘し得るので「表記水準」とした。

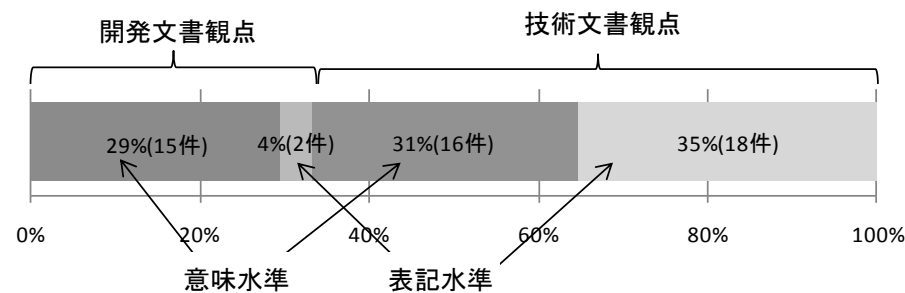


図3 文書診断法の「診断観点」と「診断水準」ごとの ESCR の「作法詳細」の分布

51 件の「作法詳細」の全てを文書診断法の「診断観点」と「診断水準」に対応づけた結果、「技術文書観点」の「表記水準」には 18 件、「開発文書観点」の「表記水準」には 2 件、「技術文書観点」の「意味水準」には 16 件、「開発文書観点」の「意味水準」には 15 件が対応づけられた。「開発文書観点」の「表記水準」の指標が 2%であるが、ほかはいずれも 30%前後であった。「診断観点」と「診断水準」に対する「作法詳細」の分布の件数と割合を図 3 に示す。

## 4. 考察

### 4.1 文書診断法と ESCR の指標の対比

文書診断法と ESCR の指標について、抽象度の高い指標を上部に、具体度の高い指標を下部に位置させて対比させた結果、ESCR の指標では抽象度の高い品質特性から、抽象度の低い「ルール」まで段階的に体系付けられていることが確かめられた。他方、文書診断法の指標では指摘した「診断項目」を中心にして、それを「診断観点」と「診断水準」と「診断種別」で類別するように位置づけられているため、指標の目的や抽象度に違いがあることが確かめられた。

最も抽象度が高い指標は、ESCR の指標では品質特性であり、他方、文書診断法では開発文書の品質に関わる「診断観点」と「診断水準」である。開発文書には品質特性が与えられていないため、「診断観点」と「診断水準」をソフトウェア製品の品質特性に関連づけるべきか、「診断観点」と「診断水準」から開発文書の品質特性を定義すべきであるかについては、今後検討の余地がある。

最も抽象度が低い指標は、ESCR の指標では「ルール」であり、他方、文書診断法の指標では問題を指摘するための「診断項目」である。「ルール」には C 言語の命令文や表記法について具体的に「このように書くべき」といった指示として与えられているが、「診断項目」では個別具体的な指示は与えられていない。そのため、文書診断法では、指摘された問題点への対応は書き手に委ねられている。したがって、文書診断法にはより具体的な指標を与える余地がある。例えば、具体的な指標として自然言語での書き方の指示を与えることで記述の統一が図られる。ただし、自然言語の表現を制限することによる影響は十分に検討する必要がある。

### 4.2 「文書診断法」から見た「コーディング作法」の特徴

各診断観点と各診断水準に対応づけられた「作法詳細」の特徴を考察する。

「技術文書観点」に対応づけられた 34 件の「作法詳細」には「領域は初期化してから使用する」「宣言の書き方を統一する」などがある。これらは、プログラミング特有の語彙が用いられているが、技術文書一般に適用することが可能な作法である。

「開発文書観点」に対応づけられた 17 件の「作法詳細」には「問題発生時の原因を調査しやすい書き方にする」「情報損失の危険のある演算は使用しない」などがある。

これらは、次工程であるテスト工程を考慮した記述やソフトウェア開発のために緻密な記述など、開発文書として必要な作法である。

「表記水準」に対応づけられた 20 件の「作法詳細」には「構造化されたデータやブロックは、まとまりを明確化する」「コーディングスタイルを統一する」などがある。表記に統一性を持たせることや構造を把握しやすく記述したりすることは、可読性を高めるためにプログラムに必要な作法である。

「意味水準」に対応づけられた 31 件の「作法詳細」には「ソースファイル取り込みについて、処理系依存部分を確認し、依存しない書き方にする」「領域の属性は明示する」などがある。記述する内容や処理の内容を把握して、それが分かるように文書に表すために必要な作法である。

これらの結果、プログラミング言語で表された実装工程の成果物としてのプログラムが備えるべき性質は、自然言語で表された作業工程の中間成果物としての開発文書が備えるべき性質と共通していることが確かめられた。

#### 4.3 ソフトウェア開発工程の品質計測

本稿では、主に自然言語で書かれた要求仕様書や設計書などに適用してきた文書診断法をプログラムのコーディング規約の基本概念である「作法」に対応づけた。このことはプログラムに対して、「診断観点」と「診断水準」を用いて開発文書として評価する可能性を示唆する。さらに、「診断観点」と「診断水準」は、自然言語やプログラミング言語以外の記述形式で表された開発文書の評価する指標になる可能性がある。文書診断法の「診断観点」と「診断水準」がソフトウェア製品の品質特性とどう関わっているのかを今後検討していく。

#### 4.4 プログラム教育への適用

藤田らは文書診断を行い、その結果に応じた教育を実施することで、技術者に不足している技術力を育成できる可能性がある<sup>5)</sup>。

本稿で行った対応づけの結果に、藤田らが提案する文書診断法による文書の改善活動を適用することで、実装工程の成果物の開発文書であるプログラムを基にした改善活動が可能である。ESCR の「作法詳細」を指標とした文書診断をプログラムに対して行い、その結果を分析して技術者に不足している技術力を明らかにし、それに応じた教育を行うことで、品質の高いプログラムを出力するために必要な技術力を育成し得る。文書診断の結果から技術者に不足している技術力を分析することと、それに対応する教育の検討は今後の課題として残されている。

### 5. まとめ

我々は、プログラムを含む、要求定義書や設計書などを作業工程が出力する中間成果物であると捉えた。プログラムをソフトウェア開発の作業工程が出力する開発文書

とみなしたとき、我々が自然言語で表された開発文書の評価をするために提案している文書診断法から見ると ESCR の指標がどのように位置づけられているかを検討した。

文書診断法と ESCR の指標を対比することで、抽象度に沿ってそれぞれの指標の位置づけが相対的に明らかになった。

「診断観点」と「診断水準」に ESCR の「作法詳細」を対応づけた結果から、C 言語で表されたプログラムが備えるべき性質は、自然言語で表された開発文書が備えるべき性質と共通していることが確かめられた。

本稿の結果から、「診断観点」と「診断水準」を用いることで、作業工程の中間成果物である開発文書としての側面を評価し得ることが確かめられた。本稿で行ったコーディングの「作法」に対する試みは、「診断観点」と「診断水準」を様々な記述形式に適用可能であるかを検討する端緒となった。

「診断観点」と「診断水準」を用いてプログラムを評価することで、文書診断法によって技術者に必要な技術力や教育を分析する手法がプログラムに対しても応用し得ると考える。

### 参考文献

- 1) 日本規格協会: ソフトウェア製品の品質第 1 部: 品質モデル, 財団法人日本規格協会(2003).
- 2) 経済産業省, (独) 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: 組込みソフトウェア開発向け 品質作り込みガイド(2008).
- 3) 竹内広宜, 荻野紫穂, 中田武男, 坂本佳史, 福岡直明: テキスト分析技術を用いた開発関連文書の品質分析, 組込みシステムシンポジウム 2009, pp 93-100 (2009).
- 4) J.D. Arthur and K.T. Stevens: Assessing the adequacy of documentation through document quality indicators, Proceedings of the Conference on Software Maintenance, pp 40-49, (1989).
- 5) 藤田 悠, 山本 雅基, 中澤 達夫, 塩谷 敦子, 池田 貴一, 楡井雅巳: 組込みソフトウェア開発文書診断法, 情報処理学会研究報告, Vol.2010-EMB-16, No.37 (2010).
- 6) 藤田 悠, 山本 雅基, 中澤 達夫, 塩谷 敦子, 池田 貴一, 楡井 雅巳, 小野 伸幸: 文書診断図を用いたソフトウェア開発文書の診断表現, 情報処理学会 組込みシステムシンポジウム 2010 論文集, Vol.2010, No.10, pp49-54 (2010)
- 7) 経済産業省, (独) 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: 改訂版 組込みソフトウェア開発向け コーディング作法ガイド [C 言語版] (2007)
- 8) 経済産業省, (独) 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: 改訂版 組込みソフトウェア開発向け 開発プロセスガイド(2007).
- 9) 日本規格協会, MISRA - C 研究会: 組込み開発者におくる MISRA-C:2004-C 言語利用の高信頼化ガイド(2006).
- 10) L. W. Cannon and R. A. Elliott and L. W. Kirchhoff and J. H. Miller and J. M. Milner and R. W. Mitze and E. P. Schan and N. O. Whittington and Henry Spencer and L. W. Cannon and R. A. Elliott and L. W. Kirchhoff and J. H. Miller and J. M. Milner and R. W. Mitze and E. P. Schan and N. O. Whittington and Henry Spencer: Indian hill c style and coding standard (1997)