

## 組込みソフトウェア開発文書診断法

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

ソフトウェアの製品品質向上の要求に応えるために、企業ではプロセス品質の向上が求められている。

我々は、ソフトウェア開発プロセスの改善の手段として、ソフトウェア開発文書品質の改善に着目した。そこで、企業の技術者が作成するソフトウェア開発文書における問題点を抽出する手法として、文書診断法を提案する。この文書診断法は、文法などの形式的な誤りだけでなく、ソフトウェア開発プロセスに関する問題点を検出するために、人手により行う。この手法を適用し、企業の開発文書を診断することにより、開発文書の品質を向上させるための課題が明らかになると考える。

文書診断の結果、計算機処理による診断では十分に分析することが難しい、図表や意味的な判断を必要とする問題点を抽出した。今後、文書診断法の改善と、文書診断によりプロセス品質および製品品質が向上することを評価・計測する方法の検討が課題である。また、文書診断および診断の修正作業により、技術者の基礎能力とソフトウェア開発能力が向上することを期待する。

### Diagnostic Method for Embedded Software Development Documents

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

In order to meet the demand of the improvement in product quality of software, improvement in process quality is required in the company. Improvement in quality of software development documents is thought to be effective for the improvement of a software development process.

In this report, we propose a document diagnostic method as the technique of extracting problems in the software development documents which an engineer draws up. In this method, it is expectable to be able to detect the problem not only about formal errors, such as grammar, but problems in software develop-

ment process because not automatic extraction by a computer but the human being judge.

By using this method, problems in actual development documents, which are in charts and which requires semantic judgment, have been extracted. These problems are thought to be difficult to extract by automatic diagnosis.

#### 1. はじめに

ISO 9126-1(JIS X 0129-1)によれば、「プロセスの評価及び向上は製品品質を向上させる手段」である<sup>1)</sup>。2009年度組込みソフトウェア産業実態調査報告書-経営者及び事業責任者向け調査<sup>2)</sup>では、システムにおける不具合の最大の原因は、ソフトウェア(31.5%)とされ、事業責任者は、その再発防止対策として、ソフトウェア開発プロセスの見直しを1番目に挙げている。したがって、企業の開発現場において、ソフトウェア開発プロセスの改善が重要であり、Software Engineering Process Group(SEPG)の活動が広がっている。しかし、プロセスの改善が具体的な製品品質の向上に結びついているかの明確な検証は行われておらず、様々な取組みは手探りの状態である。

我々は、品質改善の基盤はソフトウェア開発文書の改善にあると位置づけ、この文書の改善が、プロセス改善、ひいては製品品質の改善に結びつくと考えた。また、開発文書の役割として、塩谷・菊池<sup>3)</sup>は、「開発の手段」の他に「標準化の推進」、「プロジェクト管理の手段」、「共有財産化と技術の伝承」の計4項目をあげており、開発文書の品質の向上が効果を及ぼす側面は多い。

文書の改善方法として、従来、レビューという手法が採られている。しかし、現実には不具合が後工程に流出する割合はおおよそ45%にも上るとの報告<sup>4)</sup>があり、十分な効果をあげているとは言いがたい。

従来、ソフトウェア開発文書の品質評価や品質改善には、ソフトウェア開発文書のモデル

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

†2 合同会社 イオタクラフト  
IOTACRAFT,LLC.

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

†4 名古屋大学  
Nagoya University

を起点とした方法がとられている<sup>5)6)</sup>。竹内・荻野・中田・坂本・福岡<sup>5)</sup>は、IEEE<sup>7)</sup>が示す、要求仕様が満たすべき8項目(妥当性、非曖昧性、完全性、無矛盾性、重要度、検証可能性、変更可能性、追跡可能性)に着目し、これらを満たしていない表現の抽出を行っている。ただし、計算機処理によるテキスト分析の技術的な制約があるので、非曖昧性、完全性、検証可能性、追跡可能性の4項目のみに限定している。

また、Arthur・Stevens<sup>6)</sup>は、ソフトウェア開発文書を満たす条件として、正確性、完全性、使用性、拡張性の4種の品質属性を取り上げ、一貫性、可読性など13種の要因に分解し、さらにシステム充足性、文書簡潔性など37種の詳細な観測単位を与えた評価分類を示している。

計算機処理では、妥当性や無矛盾性などの意味的な理解や図表の解釈が困難である。さらに、開発文書を改善するための問題点の抽出を目的としたときに、ソフトウェア開発文書の品質属性を分解したArthurらの観測単位は十分であるか、検証する必要がある。

我々の提案する文書診断法は、構文解析プログラムなどを用いた計算機処理ではなく、診断者が文書を読み、問題点を抽出する。従来一般的に行われているレビューとは異なり、より根本的な問題点の抽出と改善点の指摘までを行う、開発文書に対する総合的な取組みである。

この方法は、計算機を用いた方法に比べ、図表の分析や文章の意味的な解釈が必要な診断において有利であると考えられる。一方で、診断の品質が診断者の経験に依存するという問題点も予想される。

本稿では、開発文書の改善を目的とした文書診断について報告する。企業の開発文書に対して、提案する文書診断の方法を適用し診断した結果を示し、その結果の分析と考察を行い、まとめと今後の展望を示す。

## 2. ソフトウェア開発文書診断の方法

### 2.1 診断観点と診断種別の設定

診断者がソフトウェア開発文書の問題点を的確に指摘するために、我々は、診断観点と診断種別を設定した。

#### 2.1.1 診断観点

ソフトウェア開発プロセスに基づいて開発が行われる場合において、ソフトウェア開発文書には、プロセスに割り当てられた作業成果が書き込まなければならない。すなわち、技術文書に必要な内容を記述するだけでなく、ソフトウェア開発プロセスとの整合性に注意を

払わなければならない。そこで、テクニカルライティング観点とソフトウェア開発観点の二種類の観点によって文書診断を行うことを提案する。

ソフトウェア開発文書に対する診断観点は、テクニカルライティング観点とソフトウェア開発観点の二種類の観点で構成することとした。テクニカルライティング観点は、診断対象の文書が技術文書として普遍的に要求される品質水準に達しているかを判断する観点である。ソフトウェア開発観点は、診断対象の文書がソフトウェア開発プロセスの中間成果物として必要十分な内容として作成されているかを判断する観点である。ただし、両観点に共通する項目はテクニカルライティング観点のみに設定するものとする。

同様に、ハードウェア開発文書の診断に必要な観点は、テクニカルライティング観点とハードウェア開発観点で構成される。また、ソフトウェアとハードウェアを含めた総合的な組込みシステム開発文書の診断に必要な観点は、図2.1のように、テクニカルライティング観点を含み、ソフトウェア開発観点とハードウェア開発観点およびシステム開発観点で構成される。互いに重なる部分は、関連する部分の記述に必要な観点である。例えば、ソフトウェア開発観点とハードウェア開発観点の共通部分の内、テクニカルライティング観点以外の部分は、ソフトウェアが制御するハードウェアの入出力など、ハードウェアとソフトウェア相互に関連する記述を診断するために必要な観点である。

組込みシステムに関わる共通の観点である、テクニカルライティング観点による文書診断では、基本的な文書読解能力は必要であるが、ソフトウェアとハードウェア固有の知識は不要である。他方、ソフトウェア開発観点およびハードウェア開発観点による文書診断では、それぞれの専門知識が求められる。診断観点を分けることにより、文書診断に必要な知識を限定され得る。

#### 2.1.2 診断種別

我々は、それぞれの観点に基づいて文書診断を行う際に注目すべき、診断種別を設けることとした。

テクニカルライティング観点では、用字・用語、文法、表記ルール、図表の表現、文章の表現、文書構成の6種類の診断種別を設定した。他方、ソフトウェア開発観点では、表記ルール、図表の表現、文章の表現、文書構成、記述の入出力、根拠の記述、文書体系の7種類の診断種別を設定した。表2.1は、診断観点別に診断種別を示す。“○”は、各観点に含まれる診断種別を表し、“-”は含まれない診断種別を表す。

これらの診断種別は、これまで行ってきた開発文書作成の指導や診断経験を元に、特にテクニカルライティング観点については「日本語スタイルガイド」<sup>8)</sup>を参考にして設定した。

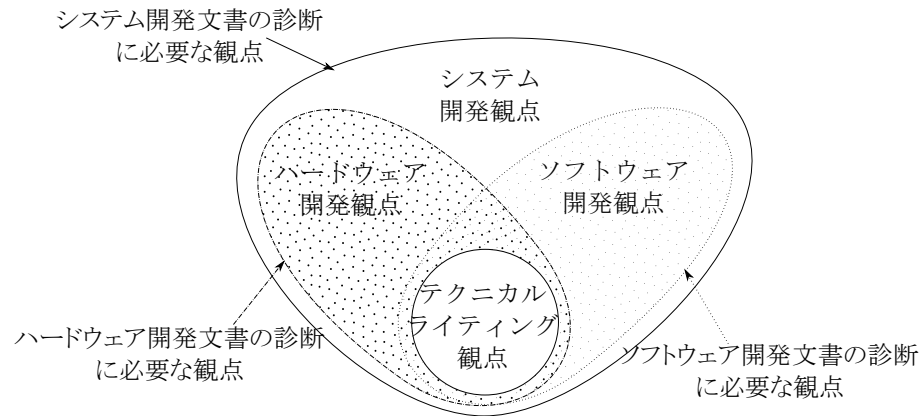


図 2.1 組み込みシステム開発に必要な開発文書観点の関係

Fig. 2.1 Relation between document viewpoints for embedded system development

表 2.1 診断観点別診断種別

Table 2.1 Classification of diagnostic items for each viewpoint

診断種別	テクニカルライティング観点	ソフトウェア開発観点
用字・用語	○	-
文法	○	-
表記ルール	○	○
図表の表現	○	○
文章の表現	○	○
文書構成	○	○
記述の入出力	-	○
根拠の記述	-	○
文書体系	-	○

なお、診断種別は上記の 13 種に限定するものではなく、文書診断で抽出される問題の内容に応じて、追加と修正を可能とする。

## 2.2 診断手順

診断者は診断対象の文書を読み、テクニカルライティング観点とソフトウェア開発観点の双方により、診断種別を基準にして、ソフトウェア開発文書の問題点を指摘事項として抽出する。次に、指摘事項の内容から、問題内容を抽象化した指摘項目としてまとめ、指摘項目ごとに指摘事項の件数を数える。この作業を、診断対象の文書に対して繰り返し実行する。

## 3. 診断実施

### 3.1 診断対象

産業用大型プリンタへの搭載を目的に開発された組み込みソフトウェアの開発文書群（全 6 冊、総ページ数は 110 ページ、文字数は 74,387 字）を文書診断の対象とした。表紙ページ、目次を含む全ページを診断対象とし、図や表も対象とした。

### 3.2 診断結果

文書診断は、1 冊の文書に対し 2 名の診断者が行った。診断者は、対象文書の作成および関連するソフトウェア開発作業に関わらない、研究員 2 名（研究員 A と B）である。研究員 A は、ソフトウェア開発文書診断経験 1 年未満、組み込みソフトウェア開発経験 1 年未満、実験用ソフトウェア開発経験 10 年である。研究員 B は、ソフトウェア開発文書診断経験 3 年、ソフトウェア開発経験 10 年以上、組み込みソフトウェア開発文書作成に関する教育講師経験 10 年以上である。なお、今回の文書診断は、診断の経験が少ない研究員 A を研究員 B が指導する目的を含んでいる。

一つの文書に対して、先に研究員 A が文書診断を行い、続けて研究員 B が文書診断を行った。研究員 B は研究員 A の診断記録を見て既指摘事項を確認し、さらに未指摘事項を抽出した。その後、2 名の診断者が指摘について討議し、問題点としての指摘の妥当性を確認し、診断項目ごとに指摘事項の件数を確定した。実施に要した時間は 2 名合計 66 時間、指摘件数は計 487 件であった。

文書診断により抽出した、テクニカルライティング観点の指摘事項 286 件、ソフトウェア開発観点の指摘事項 201 件を診断種別に分類した結果を図 3.1 と図 3.2 に示す。また、診断種別ごとに、抽出した指摘項目の詳細は付録 A.1 に示す。テクニカルライティング観点では、図表の表現、文章の表現に該当する指摘が全体の 57% を占め、ソフトウェア開発観点の指摘では、図表の表現、文章の表現、文書構成に該当する指摘が全体の 74% を占めた。

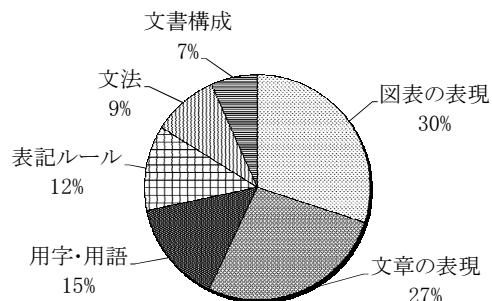


図 3.1 テクニカルライティング視点の診断種別分類  
Fig. 3.1 Result of diagnostic for technical writing viewpoint

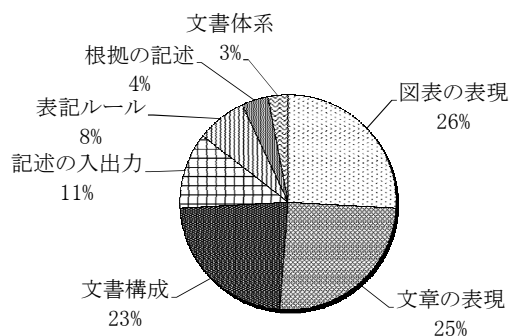


図 3.2 ソフトウェア開発視点の診断種別分類  
Fig. 3.2 Result of diagnostic for software development viewpoint

ともに、図表に関する指摘が目立つ結果となった。

## 4. 考 察

### 4.1 先行研究との関連

実施した診断結果をもとに、抽出した指摘項目および本稿で設定した診断視点と、Arthur らが与えた評価分類との関連を検討した。

Arthur らは、開発文書が持つべき 4 種の品質属性を取り上げ、次のように定義している。

正確性は、ソフトウェア開発文書において全ての要求に対してコードと開発文書の間に一貫性があること。完全性は、ソフトウェア開発文書のために必要な情報が揃っていること。使用性は、可用性・利便性・適切性を備えていること。拡張性は、システム変更に対応性があること。これら 4 種の品質属性を 13 種の要因に分解し、37 種の観測単位を与え、3 階層を持つ木構造の評価分類木を与えている。拡張性については分解がされていないが、開発文書に必要な要素であるため、品質属性の一つとして提示するに留まっていると述べている<sup>6)</sup>。

#### 4.1.1 指摘項目と評価分類の比較

我々が抽出した各指摘項目に対し、Arthur らの評価分類（品質属性/要因/観測単位）を対応づける。例えば、テクニカルライティング視点の診断種別「用字・用語」の指摘項目「用語の不統一」に対しては、品質属性「使用性」の要因「可読性の論理的理解」の観測単位「用語一貫性」を対応づける。

対応づけを分析した結果、要因と観測単位が対応付けられない指摘項目の存在が判明した。我々が抽出した指摘項目の内、10 項目に対して要因が、14 項目に対して観測単位が対応付けられなかった。観測単位において対応付けられなかった指摘項目の例として、ソフトウェア開発視点の診断種別「記述の入出力」の 4 つの指摘項目「何を記述するかが不明確」、「入出力が不明確」、「次工程への入力情報が不十分」、「同意の表現の使用や表現のコピー」は、品質属性「完全性」の要因「文書関連性」に対応付けたが、観測単位には対応付けられるものがなかった。このことは、我々が提案する文書診断法が、Arthur らの現在の評価分類に従う限りにおいては見逃される、問題のある記述を検出し得ることができることを意味する。

Arthur らは、4 種の品質属性および 13 種の要因に当てはまる新たな観測単位の追加を今後の研究対象にあげており、現在の評価分類が完全ではないことを認めている。他方、当てはまる要因がなかった事に関しては改善の必要性などを明言していない。しかし、要因は観測単位と品質属性を繋ぐ架け橋であると述べられていることから、観測単位の追加によって要因を追加する必要がある。

#### 4.1.2 診断視点と品質属性の比較

我々が定義したテクニカルライティング視点およびソフトウェア開発視点と、Arthur らが定義した品質属性の関係を比較する。図 4.1 は、文書診断で抽出された 487 件の指摘事項を、4 種の品質属性に分類し、テクニカルライティング視点とソフトウェア開発視点の別に集計した結果である。我々は、テクニカルライティング視点の使用性に相当し、ソフトウェア開発視点の正確性と完全性に相当すると予想していた。その理由は、Arthur らが、正確

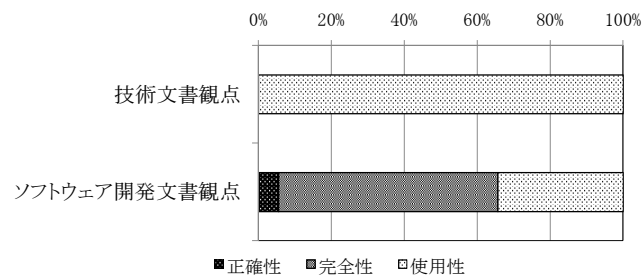


図 4.1 観点ごとの品質属性の分布  
Fig. 4.1 Rate of quality attribute for each viewpoint

表 4.1 「文書表現」に関する指摘項目  
Table 4.1 Defects in “sentence representation”

テクニカルライティング観点	ソフトウェア開発観点
あいまい表現 表現がわかりにくい 冗長表現 表現の不足 文体の不統一 誤字・脱字 表記内容の対応が取れていない 一文一義でない	開発上の表現の不足 開発上の冗長表現 開発上の表現が不明確 条件漏れ・記載漏れ 未決事項の存在・解決すべき事項が不明確

性と完全性をソフトウェア開発文書が必要とする性質として定義しており、使用性を技術文書一般に必要なとされる性質として定義しているためである。

分類の結果、テクニカルライティング観点では、全ての指摘事項が使用性に分類された。他方、ソフトウェア開発観点では、正確性あるいは完全性に分類された指摘事項を合わせて全体の 65.7%にとどまり、使用性に分類される指摘事項が全体の 34.3%を占めた。上記の我々の予想に反した理由として、診断観点が明確に分解できていないことが挙げられる。

ソフトウェア開発観点到分類され、使用性に分類された指摘項目は 7 項目あった。その中で、最も指摘事項の件数が多い指摘項目は、診断種別「図表の表現」の指摘項目「図表の位置づけが不明確」であり、23 件であった。この指摘項目は、テクニカルライティング観点の指摘と捉えることも可能である。しかし、ソフトウェア開発文書では、特にソフトウェア開発プロセスの作業を明確にすべきであると考え、我々はソフトウェア開発観点到分類した。

この他にも、診断者は、片方の観点を指摘を行う際に、他方の観点を指摘事項を発見したことや、指摘事項をどちらの観点到分類すべきかに迷う経験をしたことを報告している。このことは、両者の観点が明確に分離していない可能性を示唆している。これらの文書診断における問題を解決するために、診断観点と診断種別を含めて診断手順の見直しが必要である。

## 4.2 診断手法の評価

### 4.2.1 観点を分けることの評価

テクニカルライティング観点的診断では、ソフトウェア開発技術者でなくても修正が可能指摘が抽出された。他方、ソフトウェア開発観点的診断では、ソフトウェア開発技術者で

なければ修正が難しい指摘が抽出された。

診断種別「文書表現」を例にして、両観点的の違いを述べる。なお、「文書表現」は両方の観点到含まれる。「文書表現」に該当する指摘項目を表 4.1 に示す。テクニカルライティング観点的では、内容を正確に表現するために必要な指摘が行われ、ソフトウェア開発観点的では、ソフトウェア開発プロセスに関わる指摘が行われた。次に、具体的に「冗長表現」を例に取り上げ、両者の違いを述べる。

まず、テクニカルライティング観点的における「冗長表現」の例を示す。ただし、実際には具体的な製品名や部品名が記載されていたが、機密保持のために改変して記載する。なお、注視すべき箇所に下線を引いた。

本体背面のメインスイッチを OFF にしたとき、パラメータ内容の保護については、X の仕組みにより、パラメータ内容は保障される。

この例では、上記 2 箇所の下線部が「パラメータ内容の保護」を重複して述べており、冗長である。例えば、以下のように改訂すべきである。

本体背面のメインスイッチを OFF にしたとき、X の仕組みにより、パラメータ内容は保護される。

次に、ソフトウェア開発観点的における「開発上の冗長表現」の例を示す。前出の例と同様に、機密保持のために改変している。なお、注視すべき箇所に下線を引いた。

『〇〇ソフトウェア仕様書』における記述：

インク消費量低減のため、専用洗浄液によるポンプチューブ洗浄を 新たに追加する。

上記の例では、上記下線部より、ポンプチューブ洗浄機能の追加が『〇〇ソフトウェア仕様書』で定義されるように読み取ることができる。一般的に、組込みソフトウェア開発における機能要求の定義は、要求仕様書で述べるべきであるので、この文が要求仕様書に書かれて

いるならば、問題は無い。しかし、診断対象のこの文書はソフトウェア仕様書である。機能を実現するための設計が記述されるべきであり、機能を新たに追加することを再度記述することは冗長である。

これらの事例は、テクニカルライティング観点は文書読解力を用いて文書を診断するものであり、ソフトウェア開発観点はソフトウェア開発に関する知識を用いて文書を診断するものであることを示す。また、診断結果の指摘を元に文書の改訂を行う場合に、テクニカルライティング観点では指摘を受けた文を局所的に修正することが可能であるが、ソフトウェア開発観点では関係する文書との関連性を踏まえて修正を行う必要がある。

#### 4.2.2 診断種別の評価

今回、診断者が診断後に互いの診断結果を分析するために討議した所、診断種別があることで問題のある箇所を見つける際の参考になった。特に、診断経験が少ない診断者の方が、その傾向が強かった。ただし、診断者 A が診断を行った後の、診断者 B の診断により、同一文書から診断者 A が抽出していない指摘も多数抽出されている。診断種別は、全ての問題を抽出するためのチェックリストの役割を果たすものではなく、より広範囲な問題点の抽出を可能にする。

なお、診断種別に分けて指摘項目ごとの指摘事項の件数を示すことで、文書診断結果の傾向を把握することが可能になった。すなわち、診断対象の文書が、そもそも文法的に問題が多いか否かということや、文法的には問題がないが文章表現に不備があり曖昧であることなどの傾向を掴むことができた。

#### 4.2.3 人による診断の評価

計算機処理による診断では分析が困難な、図表に関する問題やソフトウェア開発プロセスに関わる問題が抽出できる。例えば、「記述の入出力」は「前工程からの引継ぎ事項や引用等の入力は何であるか、またこの工程で決定する事項等の出力は何であるか、が明確であるか」、また「文書体系」は「ソフトウェア開発プロセスに沿った、文書の位置づけが明確であるか」に注目する診断種別である。これらの診断種別に関する指摘は、診断対象とする文書のプロセスにおける位置づけを理解しなければ抽出は不可能であるので、計算機処理での分析が困難である。今回の文書診断では、それらの診断種別の指摘を抽出した。

しかし、診断者の経験によって、抽出する指摘の網羅性に違いが生じた。この課題を解決するために、まず、一定水準の診断能力の確保と統一的な診断基準の確立が必要である。そのために、今後、文書診断者のコミュニティを形成し、診断法の検討や診断技術者の育成を行うことや、診断者の能力を評価あるいは認定する試験制度の設立に関する検討などが望ま

れる。

#### 4.3 実用性の評価

本稿で実施した文書診断では、110 ページの開発文書の文書診断に要した工数は、66 人時間であった。すなわち、1 時間あたりの文書診断速度は 1.7 ページである。しかし、今回は、文書診断の経験が豊富な研究員が未熟な研究員の指導を行うことを目的の一つに加えていたので、文書診断だけに費やされた時間が計測されておらず、生産性の評価を直ちに行うことはできない。

ただし、従来のレビューよりも高い精度で指摘を抽出し、詳細に分類することを目的とした文書診断は、ある程度の時間を必要とすることが想定される。したがって、企業で実用的に文書診断を用いるためには、複雑な要求機能や重要な要求機能に関する設計文書といった、診断対象とする文書を適切に選択する必要がある。

なお、我々は単に文書の品質向上に留まらず、開発プロセスの改善につなげることで、製品品質の向上につながることを期待しており、そのための評価や計測の方法の検討は今後の課題である。

#### 4.4 教育的効果の期待

文書診断により検出される問題点の傾向は、診断対象である開発文書を作成した技術者に欠如する能力の傾向を示唆する。例えば、テクニカルライティング観定の指摘が多い場合は、技術文書を書くための基礎力に問題があり、ソフトウェア開発観定の指摘が多い場合は、ソフトウェア開発プロセスに割り当てられた作業を遂行する技術力に問題があることを示唆する。さらに、テクニカルライティング観定の指摘においても、診断種別の分析から、文法力が不足しているのか、あるいは文法的には正しいが分かりやすい文を書く能力が欠如しているかなどを確認できる。

必要に応じて、能力を向上させる教育を集中的に行うことで、効果的に技術者を育成できる可能性が高まる。また、指摘された各問題点を執筆者が修正することは、自ら問題点を理解し、作業自体を明確に遂行することになるので、ソフトウェア開発の品質を高めることにつながる。さらに、ソフトウェア開発に留まらず、以前に受けた指摘を意識して文書を作成し、論理的に物事を考えることにより、技術者に必要な基礎能力の育成が期待できる。

## 5. ま と め

ソフトウェア開発文書診断法を提案し、ソフトウェア開発文書に対する文書診断と評価を行った。診断の結果、全 6 冊のソフトウェア開発文書から、487 件の指摘事項が抽出され

た。本診断法は、品質属性の分解によって示された評価分類に無い問題点を抽出できていることを確認した。さらに、診断時における観点の分類作業では、一意に決定しにくい指摘事項が検出された。ソフトウェア開発文書に対する診断の観点として、テクニカルライティング観点とソフトウェア開発観点を定めることで、診断に必要な知識、また指摘された問題の解決に必要な知識を分けることができた。診断種別を設定することによって、文書診断の指摘の傾向を把握することができた。また、計算機処理による診断では十分に分析することが難しい、図表や意味的な判断を必要とする問題点を抽出できた。

観点を定めることは、診断を絞り込んで行うことになり、診断者の注意を集中させ、診断の品質を向上させることに有効である。しかし、観点が明確に分離していない可能性もあるので、診断観点および診断種別、また診断手順の見直しが必要である。本診断法は、人手による手法であるため、診断者の経験の違いによって診断結果に差異が生じる欠点がある。そのため、一定水準の診断能力の確保と統一的な診断基準を確立することが必要である。また、文書品質が向上することにより、プロセス品質が向上、ひいては製品品質が向上することを評価および計測する方法が必要である。

さらに、文書診断の対象となる開発文書を作成した技術者にとって、文書診断および文書診断による改善作業によって、基礎能力やソフトウェア開発能力などを主とした、能力育成に有効であることが期待できる。

## 付 録

### A.1 文書診断結果

各診断観点で抽出された指摘項目を、診断種別ごとに示す。“[ ]”を用いた記述は指摘項目が該当する Arthur らの評価分類であり、[品質属性/要因/観測単位]の書式で示している。要因が2段階に分かれているものは“:”で分けている。“\*”は、レベルに当てはまる要因、観測単位がなかった事を表している。

#### A.1.1 テクニカルライティング観定の指摘項目

テクニカルライティング観定の診断種別に当てはまる指摘項目は39項目に分類された。以下に、診断種別ごとの指摘項目を示す。

##### (1) 用字・用語

- 意味的な誤用 [使用性/可読性:論理的理解/標準順守]
- 用語表現がわかりにくい [使用性/可読性:\*)]
- 使い分け不備 [使用性/可読性:\*)]

- 用語の不統一 [使用性/可読性:論理的理解/用語一貫性]
- 用語定義なし [使用性/可読性:論理的理解/用語集の十分性]
- 用字のあいまい性 [使用性/可読性:論理的理解/用語一貫性]
- 用語・用字の表記ルール違反 [使用性/可読性:論理的理解/標準順守]

##### (2) 文法

- 安易な受動態の使用 [使用性/可読性:\*)]
- 主語、述語、目的語が不明 [使用性/可読性:論理的理解/標準順守]
- かかり受けが不明確 [使用性/可読性:論理的理解/標準順守]
- 助詞、句読点の誤用 [使用性/可読性:論理的理解/標準順守]
- ねじれ文 [使用性/可読性:論理的理解/標準順守]
- 接続詞が不適切 [使用性/可読性:論理的理解/標準順守]

##### (3) 表記ルール

- 引用・参照の基本ルール違反 [使用性/可読性:論理的理解/標準順守]
- 表記のあいまい性 [使用性/可読性:物理的理解/書式一貫性]
- 箇条書きの表記ルール違反 [使用性/可読性:物理的理解/書式一貫性]
- ページの付け方不備 [使用性/可読性:物理的理解/書式一貫性]
- 記号の定義なし [使用性/可読性:物理的理解/\*]
- 目次の不備 [使用性/追跡可能性/目次の十分性]

##### (4) 図表の表現

- 列や行の見出しに対する記述の一貫性なし [使用性/可読性:物理的理解/書式一貫性]
- 図と本文との区別なし [使用性/可読性:論理的理解/標準順守]
- 図の解説の不足 [使用性/可読性:論理的理解/標準順守]
- 図表のあいまい表現 [使用性/可読性:論理的理解/標準順守]
- 図の凡例の不備 [使用性/可読性:論理的理解/標準順守]
- 図表の基本的な表記ルール違反 [使用性/可読性:論理的理解/標準順守]
- 図表なし [使用性/可読性:論理的理解/文書簡潔性]
- 図表表現の不統一 [使用性/可読性:物理的理解/書式一貫性]
- 図表の基本的な表記ルール違反 [使用性/可読性:物理的理解/\*]

##### (5) 文章の表現

- あいまい表現 [使用性/可読性:\*/]
- 表現がわかりにくい [使用性/可読性:\*/]
- 冗長表現 [使用性/可読性:論理的理解/冗長の適切性]

- 表現の不足 [使用性/可読性:\*/]
- 文体の不統一 [使用性/可読性:論理的理解/標準順守]
- 誤字・脱字 [使用性/可読性:論理的理解/標準順守]
- 表記内容の対応が取れていない [使用性/文書内完全性/領域充足性]
- 一文一義でない [使用性/可読性:論理的理解/文書簡潔性]

#### (6) 文書構成

- タイトル, 見出しの不備 [使用性/可読性:論理的理解/標準順守]
- タイトルと内容の不一致 [使用性/追跡可能性/項目一貫性]
- 章・節・項・段落の構成の不備 [使用性/可読性:論理的理解/簡潔性:モジュール性]

#### A.1.2 ソフトウェア開発観点の指摘項目

ソフトウェア開発観点の診断種別に当てはまる指摘項目は 27 項目に分類された。以下に、診断種別ごとの指摘項目を示す。

#### (1) 表記ルール

- 開発上の引用・転載の不備 [完全性/文書関連性:関連/\*]
- 参照文書の記載の不備 [完全性/文書関連性:関連/\*]
- 改訂履歴の記載の不備 [完全性/文書関連性:分解/改定履歴]

#### (2) 文章の表現

- 開発上の表現の不足 [完全性/領域網羅性/網羅性]
- 開発上の冗長表現 [完全性/文書関連性:\*/]
- 開発上の表現が不明確 [完全性/文書関連性:\*/]
- 条件漏れ・記載漏れ [完全性/領域網羅性/網羅性]
- 未決事項の存在, 解決すべき事項が不明確 [完全性/文書関連性/未決事項]

#### (3) 記述の入出力

- 何を記述するかが不明確 [完全性/文書関連性:関連/\*]
- 入出力が不明確 [完全性/文書関連性:関連/\*]
- 次工程への入力情報が不十分 [完全性/文書関連性:分解/\*]
- 同意の表現の使用や表現のコピー [完全性/文書関連性:分解/\*]

#### (4) 根拠の記述

- 記述対象の位置づけが不明確 [正確性/要求・設計トレーサビリティ/\*]
- 根拠の記述なし [正確性/一貫性:概念一貫性/構想の不変性]

#### (5) 図表の表現

- 図表の位置づけが不明確 [使用性/追跡可能性/項目一貫性]

- 図表の解説の不足 [使用性/\*]
- 図表の構成の不備 [使用性/\*]
- 開発文書としての図表の基本的な表記ルール違反 [使用性/可読性:論理的理解/空白]

#### (6) 文書構成

- 文書冒頭での文書の目的・位置づけが不在 [使用性/可読性:論理的理解/モジュール性]
- 章・節の冒頭での記述の目的・位置づけが不在 [使用性/可読性:論理的理解/モジュール性]
- 文書構成の全体像が見えにくい [使用性/アクセス性]
- 開発作業構成として, 目次や段落の構成が不適切 [完全性/文書関連性:分解/\*]
- 開発文書としての文書タイトル, 章・節・項・段落のタイトル, 見出しの付け方が不適切 [完全性/文書関連性:分解/\*]
- 開発上の趣旨, 論点があいまい [完全性/文書関連性:分解/\*]

#### (7) 文書体系

- 「要求」の存在が不明確 [正確性/要求・設計トレーサビリティ/要求・実装等価性]
- 上位文書の存在が不明 [完全性/文書関連性:関連/\*]
- 開発プロセス内の文書の位置づけが不明確 [完全性/文書関連性:関連/\*]

#### 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) 経済産業省独立行政法人情報処理推進機構：経営者及び事業責任者向け調査, 2009年版組込みソフトウェア産業実態調査報告書 (2009).
- 3) 塩谷敦子, 菊池百代：ソフトウェアドキュメンテーション, 株式会社デンソークリエイト (2000).
- 4) 経済産業省独立行政法人情報処理推進機構：プロジェクト責任者向け調査, 2009年版組込みソフトウェア産業実態調査報告書 (2009).
- 5) 竹内広宣, 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).
- 6) Arthur, J. and Stevens, K.: Assessing the adequacy of documentation through document quality indicators, *Proceedings of the Conference on Software Maintenance*, pp.40-49 (1989).
- 7) IEEE: IEEE Recommended Practice for Software Requirements Specifications, *IEEE Std 830-1998* (1998).
- 8) 一般財団法人テクニカルコミュニケーター協会：日本語スタイルガイド, 一般財団法人テクニカルコミュニケーター協会 (2009).