

開発文書品質の研究課題についての考察

山本 修一郎[†], 栗田 太郎^{††}, 山本 佳和^{†††}

[†]名古屋大学 情報連携統括本部 情報戦略室

〒464-8601 愛知県名古屋市千種区不老町

^{††}フェリカネットワークス株式会社 開発部

〒141-0032 東京都品川区大崎 1-11-1

^{†††}株式会社デンソークリエイティブ プロジェクトセンター システム 1 室

〒460-0008 愛知県名古屋市中区栄 3-1-1

E-mail: syamamoto@acm.org, taro.kurita@jp.sony.com, yama_y@dcinc.co.jp

あらまし システム開発を成功させるためには、開発文書の品質が重要になる。これまでに、プログラムのソースコードについては品質メトリクスなど多くの既存研究がある。しかし、開発文書の品質については明確な定義や評価基準、ならびに、その活用、改善方法が確立されていない。このため、本稿では、開発文書品質の研究課題を整理するとともに、開発文書品質研究のロードマップ策定に向けた取り組みを紹介する。

キーワード 開発文書, 仕様書, 設計書, 文書品質, 品質基準, 品質メトリクス, 開発工程, 開発手法

Research Issues on Systems Development Document Quality

Shuichiro Yamamoto[†], Taro Kurita^{††}, Yoshikazu Yamamoto^{†††}

[†]Nagoya University, Strategy Office, Information and Communications Headquarters

Furo-cho, Chikusa-ku, Nagoya, Aichi 464-8601 Japan

^{††}FeliCa Networks, Inc., Development Department

1-11-1 Osaki, Shinagawa-ku, Tokyo 141-0032 Japan

^{†††}Denso Create, Inc., Software Development Department

3-1-1 Sakae, Naka-ku, Nagoya, Aichi 460-0008 Japan

E-mail: syamamoto@acm.org, taro.kurita@jp.sony.com, yama_y@dcinc.co.jp

Abstract The quality of document is important to develop successfully systems. Many researches were proposed to evaluate the quality of program codes. However, the definitions of document quality, evaluation criteria, and usage methods are still not established. In this paper, we try to analyze research issues on systems document quality. We also discuss an approach to describe the research roadmap for systems development document quality.

Keyword Systems Development Document, Specification, Design, Document Quality, Quality Criteria, Quality Metrics, Development Process, Development Methodology

1 はじめに

システム開発文書品質研究会 ASDoQ は、文書品質の提案、計測技術の研究、品質向上方法の研究を目的として活動している [1]。文書品質の提案では、文書の品質に関する研究を進め、品質を定義し、その評価指標を提案する。計測技術の研究では、人が文書の意味を読み取りながら計測する方法や計測結果の比較研究ならびに、信頼性向上について研究する。品質向上方法の研究では、品質阻害要因の特定と対策ならびに、開発文書に関する教育について研究する。

著者らが参加している研究会のロードマップ部会では、これらの研究を進める上で、システム開発文書品質についての課題を洗い出し、システム開発技術研究会の短中期的に成果が期待できる研究課題についての議論を進めている。

たとえば、研究会で意見交換会を実施したところ、次のような開発文書の課題が現場にあることが判明した。①経験に基づく文書作成の改善②思考の道具としての技術用語の教育、③何が書いてあるのか分からな

い社内文書の存在、④グローバルなソフトウェア開発のための文書化方法、⑤組込み開発文書品質基準の標準化、⑥顧客ごとに異なる文書体系の存在やそれらへの合わせ込み、⑦開発文書の社内標準の策定、⑧開発文書事例の作成、⑨要求定義書の品質の改善、⑩アジャイル開発における文書管理方法、⑪文書作成時間の確保が困難化、⑫開発文書の活用が困難性。①から④は、開発文書を構成する文章についての課題である。⑤から⑧は開発文書の標準化についての課題である。⑨から⑫は開発文書管理についての課題である。従って、システム開発現場の文書についての課題として、文章表現、標準化、文書管理法という 3 つの課題があることが明らかになった。もちろんこれらの課題については、技術文書作成方法やソフトウェア仕様化技法など、従来から解決策が提案されているものもあり、現場で知られていないだけの技術もある。また、これらの課題の他にも現場には様々な課題があるので、継続的な調査分析が必要である。以下では、まず関連研究について紹介する。次にシステム開発課題と文書の関係について述べる。さらに、考察と今後の展望について述べる。

2 関連研究

2.1 開発文書構造

システム開発文書も技術文書の一種である。技術文書については、実は 1980 年代から様々な方法論が提案されている。たとえば、下村 [2] では、「日本人が作成するほとんどの技術文書が、欧米では通用しないという状態が依然として続いている」という文から始まる。第 I 部で日本語による技術文書の作り方の知識とパラグラフ論を提示し、第 II 部で技術英語の特質を述べて日本語と英語の変換についての知識を紹介している。パラグラフ論では、トピックセンテンスを根とする木構造で論述展開することを提唱している。また技術文書の要件として、因果律、事実と意見の区別、識別と限定、物事の定義と論述の論理性、簡潔明瞭性、短文があると述べている。例えば具体的には、一つの事柄を表現するための平均的な文字数は日本語だと 50 字であると指摘している。

日本のシステム開発文書に関して下村氏が指摘した状況は、残念ながら、それから 20 年以上経った現在も継続している。なお、この書籍では当時、ドキュメンテーション・エンジニアリング事業を実践していた日本企業があったことも紹介されている。

ソフトウェア開発文書については、文献 [3] の第 2 章「ドキュメンテーション」で、ドキュメントの重要性、種類、ドキュメントの作成と記載内容、ドキュメントの品質管理、ドキュメンテーション支援ツールについて記述されている。

開発者のためのドキュメントについては、内部仕様書、開発管理・作業手順書、メモ的ドキュメントに分けて記述項目を整理している。

ドキュメントの品質管理については、ドキュメントごとに要求される品質特性が異なり、一概に説明できないことを指摘している。仕様書の検証手順例として、①仕様書の位置づけの理解、②全体構成の見直し、③チェックリストによる内容の検証、④前工程仕様書の詳細化の妥当性確認、⑤修正箇所の再確認があること

を紹介している。

2.2 仕様書テンプレート

要求全体の構造の例としては、IEEE std. 830-1998 [4]、Volere 構造テンプレート [5, 6]、Scenario Plus テンプレート [7] などがある。要求間の関係のひな型には SysML の要求関係のひな型（ステレオタイプ）がある。要求記述項目のひな型の例には、Volere シェル・テンプレート、USDM [8] がある。要求記述文の文法のひな型には EARS [9] がある。

2.2.1 章節構成のひな型の例

要求仕様書の章節構成のひな型の例としては、IEEE std. 830-1998 [1] が代表的である。他にも、Volere 構造テンプレート [2, 3] や Scenario Plus テンプレート [4] がある。

Volere には、図 1 に示すように構造テンプレートと Volere シェル・テンプレートがある。

文書構造テンプレートでは、①プロジェクトの目的、②利害関係者、③制約条件、④命名規則、⑤前提条件、⑥ビジネスデータモデルとデータ辞書、⑦スコープ、⑧機能要求、⑨見た目と様式要求、⑩操作性要求、⑪性能要求、⑫運用と環境要求、⑬保守性要求、⑭セキュリティ要求などを記述する。

Scenario Plus テンプレートでは、①ビジョン、②ステークホルダ、③ゴール、④コンテキスト、インタフェース、振る舞い、⑥シナリオ、否定的シナリオ、⑦品質と制約、⑧根拠、⑨定義、⑩測定基準、⑪優先順位を記述する。

Scenario Plus ではステークホルダとして、所有者、運用者、規制者、競合者、その他（開発者）を挙げている。

2.2.2 要求記述項目間の関係のひな型の例

要求間の関係の代表的な例は、階層分解関係である。ゴール指向要求工学では、ゴール要求をサブゴール要求に階層的に、ゴールを分解していく。

SysML [8] では、①階層関係、②複製関係、③派生関係、④満足関係、⑤検証関係、⑥洗練関係、⑦追跡関係、⑧理由関係がステレオタイプとして定義されている。とくに SysML の階層関係で、要求を階層的に分解することによって、ゴール分解を記述できる。

USDM [9] でも、要求間の関係を記述することができる。後述するように USDM では、要求記述の中で、要求が必要となる理由を説明することができる。USDM では階層関係を要求番号の階層で記述する。

2.2.3 要求記述項目のひな型

Volere シェル・テンプレートでは、①要求番号、②要求種別、③イベント番号、④内容、⑤理由、⑥要求元、⑦適合基準、⑧顧客満足度、⑨顧客不満足度、⑩依存関係、⑪対立要求番号、⑫参考情報、⑬履歴をカード形式で記述する。

USDM では、①要求番号、②要求文、③説明、④理由、⑤要求間の関係を記述する。

SysML の要求図では、①要求識別子、②要求文、③要求関係を記述する。上述したように SysML の要求関係は、要求間の関係で予め定義されている。

図式として定義されることが多いユースケースには、記述項目が定義されている。具体的に言えば、ユースケースでは、機能要求名、主体アクタ、関連アクタ、イベントフロー、代替フロー、事前条件、事後条件、関連する非機能要求を記述することになっている。このようなユースケースの記述項目もテンプレートと考えることができる。また、アーキテクチャ分析では、品質特性シナリオの記述項目が定義されている [10]。

2.2.4 要求記述文の文法のひな型

自然言語で要求を正確に、誤りのない構文で記述することは容易ではない。また、要求を正しく記述するためには、経験や訓練が必要となる。個人によっては、思い入れのある凝った記述になったり、せっかくなかで書いた要求がレビューで手直しされると気分を書いたりすることがあるかもしれない。記述のスタイルが個人ごとに違うこともよくあることである。

要求記述構文テンプレート [11] では、次のように、要求を記述する構文を決めておき、<条件> や <イベント> など、置き換え可能な部分を穴埋め式に具体化するようにしている。

<条件> である場合、
<イベント>が発生して <時間> 以内に、
<主体> が <機能> する必要がある。

このように、構文が決まれていると、文法が統一され、要求の記述を効率化できるだけでなく、要求の理解も容易化できる。EARS テンプレートには、①遍在型、②イベント駆動型、③非期待振舞型、④状態駆動型、⑤選択型という 5 種類のテンプレートがある [12]。このテンプレートだけで航空機エンジンの安全性要求を記述できたという報告がある。

2.3 開発文書の構造品質尺度

モジュール設計の構造品質尺度として、モジュール結合度とモジュール強度が従来から良く知られている [13]。

この構造品質尺度を拡張してオブジェクト指向設計の評価メトリクスが提案されている [14, 15]。これらのメトリクスでは構造尺度に基づいてオブジェクト指向で設計されたソフトウェアの保守性や生産性を推定している。またソフトウェア開発における意思決定に利用できるメトリクスの開発が期待されている [15]。

2.4 ソフトウェア品質特性

ISO 9126-1 では、ソフトウェアの品質特性として、機能性、信頼性、使用性、効率性、保守性、移植性があるとされている [16]。開発文書の場合には、信頼性、使用性、保守性が重要な品質特性となると考える。

Arthur は保守性についての開発文書の品質尺度を提案している [17]。中條は、人的エラー防止の観点からソフトウェア開発文書が満たすべき性質を、情報の観点と容易化の観点から整理している [18]。情報の観点には①情報種別、②文書構造、③書式がある。容易化の観点には、①共通化・集中化、②特別化・個別化、③適合化がある。Dautovic らはソフトウェア開発文書に対する実践的なレビュー規則に基づいて、ワープロで作成された開発文書を自動的に検査するツールを開

発して評価している [19]。

2.5 情報品質の研究から見た開発文書品質

情報品質の研究 [19] では、①品質価値の解明、②品質特性の解明、③品質基準の確立、④品質管理の確立、⑤品質保証の確立、⑥コミュニケーションという課題が挙げられている。

Wang は、情報品質を次のように定義している [20]。

【定義】情報を利用する消費者にとっての適合性を情報品質という。

この定義に基づいて開発文書品質を定義すると、次のようになる。

【定義】開発文書を利用する関係者にとっての適合性を開発文書品質という。

また、Wang の情報品質に基づいて、技術文書の品質を自動的に測定する研究がある [21]。この研究では、技術情報への接近性を GQM (Goal-Question-Metric) によって分析することによりメトリクスを定義し、XML 文書に対する評価実験を実施している。

2.6 コミュニケーション

開発文書は、システム開発の関係者間のコミュニケーションの手段である。目的の有無と、内部と外部の関係に基づいた組織コミュニケーションモデルが提案されている [22]。このモデルに従って、開発コミュニケーションをモデル化すると、図 1 のようになる。

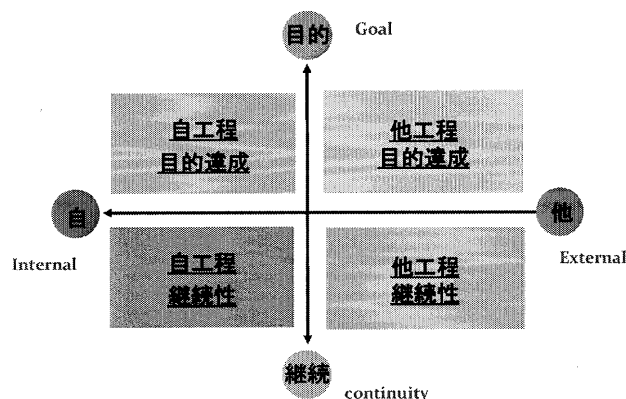


図 1: 開発コミュニケーションモデル

開発コミュニケーションを、①自工程目的達成、②他工程目的達成、③自工程継続性、④他工程継続性に分類できる。この開発コミュニケーションモデルから、次のことが分かる。ある工程担当者が作成する開発文書には、その工程の目的を達成するだけでなく、関連工程の目的達成を支援する必要がある。同時に、その工程の他の担当者とともに、その工程が継続できるだけでなく、関連する他の工程の継続性についても支援する必要がある。

2.7 自然言語と形式言語、図式言語

記述や解釈の自由度が大きい自然言語ではなく、文法や意味が定義された形式言語を用いることにより、より厳密な記述や解釈、コミュニケーションが可能となる可能性がある。例えば、形式手法の一つである形

式仕様記述手法は、数理論理学に基づく科学的な裏付けを持つ仕様記述言語と、仕様の記述や検証を支援するツールを用いることで、仕様や設計を論理的に表し、さらに、制限はあるが、検査や証明等を行うこともできるようになるものである。仕様記述言語は、システムを開発するときに検討しなければならないデータやロジックを表現するために設計されており、自然言語と比較して、素直に設計対象を表現したり、記述を執筆者の意図通りに読み解いたりすることができるようになる。一方、仕様記述言語で記述能力にも限界がある。たとえば、仕様や設計の背景や経緯、理由を記述することには適していない。このため、形式仕様に自然言語で補足することが適切である。また、形式言語を用いるのではなく、表現が制約された自然言語を用いる方法もある。

UML や SysML 等の図式言語もある。それぞれの図は特定の事柄を表現するために工夫がなされており、表現対象や目的と合致すれば、仕様や機能を図で表すことができる。しかし、すべてを図で表現できる訳ではなく、自然言語や仕様記述言語による説明や補足が必要であることが図式の方法に記されている。

開発文書群全体として捉えると、表形式も含めて、様々な言語や表現方法を用いて文書を記述することが一般的であり、今後より一層、表現の目的に合わせて適切な言語や記法を選択していく必要があるだろう。しかし、具体的にどのように言語を選択して文書を構成していくのかについて、まだ一致した方法、見解は得られておらず、今後の研究テーマとなり得る。

また、記述した文書、記述された文書の妥当性をはじめとして、書かれたことが検証されていて、正しければ良いというわけではない。例えば、開発が不可能な規模の仕様書に基づいて開発を行うべきではない。理解性、冗長性、複雑性などの品質メトリクスは文書全体に対して意味があるものであり、表現するための言語や方法によらず、形式と自由度の制御を行いながら、必要に応じて測定できなければならない。また、これらの性質を議論する上で、ソースコードとの比較も課題となる。大人数でのコミュニケーションの基盤とするのには適していない。ソースコードと比較して、自然言語で書かれた抽象的な文書は、チームやステークホルダとの意思疎通に向くが、厳密性等に問題が生じるかもしれない。開発におけるコミュニケーションの意味と、合意形成しなければならない人数、文書の役割、言語や記述の抽象度の選択も研究課題の一つとなるだろう。

2.8 ソフトウェア度量衡学

ソフトウェア度量衡学 (Software Metrology) [23] は、開発文書からソフトウェア規模を見積もる手法から発展した。ソフトウェア度量衡学では、計測方法を定義するためには、以下の 3 項目が必要であるとされる。

①計測目的、②計測対象のメタモデル、③計測すべき属性の特徴。計測方法の定義に対して数量的計測規則が対応づけられる。

3 システム開発課題と文書の関係

3.1 開発工程と文書

ソフトウェア開発プロセスの高信頼化では、後工程

の開始時に、前工程の生産物としての開発文書を検査して、品質が不足していると判断されると、後工程を開始せず、前工程に戻るというフォードバック型プロセスが適用されている [24]。また、開発組織が開発文書を検査するのではなく、第三者が独立して妥当性を確認する方法も注目されている。

開発プロセスにおいては、それぞれの工程ごとに意味がある、あるいは意味を持たせなければならない。また、工程ごとに開発文書を作成することが一般的である。このような視点において、以下のような課題がある。①工程毎の活動（工程そのもの）や成果物の定義、②要求の調整や仕様の変更、ソースコードや文書の変更管理、③ステークホルダとの関係性の確立や折衝、プロジェクトの内外への文書の二重作成、④レビューやテストによる品質確保、⑤派生開発や運用・サポートまでの考慮等である。

これらの課題は、開発文書の外側にある課題や位置付けの問題であり、開発文書そのものの書き方やレビューの問題ではない。しかし、開発を、プログラムの記述も含めて、文書の作成を中心として捉えるのであれば、以上の課題と文書作成の方法を合わせて考える必要があり、研究課題となる。また、例えばインクリメンタルな開発プロセスによっては、様々な工程の成果物が徐々に出来上がり、開発の最終段階において、要求からシステムテストまでに関わる一連の文書が最後に完成する場合もある。このような開発における文書の作成方法も研究課題である。

3.2 開発手法と文章

2.7 節でも触れたが、さまざまな開発手法は、様々な言語や記法を用いて開発対象を表現すること提案するが、文書全体の構造までには踏み込まない。また、説明や補足のための自然言語をどのように用いるのかまでは言及しない。従って、各種の手法を用いるときには、手法において提案されている言語や記法を活用すると同時に、手法で提案されていない自然言語を含む言語、記法も用いて、如何に構造化した文書、文書体系としていくのか、ということ議論しなければならない。このときに重要なことは、①各開発手法と、記述やモデル、文書の対応関係の明確化、②メタ記述、メタモデルの考案とそれらを用いたテンプレート、③記述の意図と読解の視点の整理と合致、④様々な手法や記法が混在した文書群から様々なアスペクトの抽出方法、⑤抽象度の制御等であり、これらが研究課題となる。

4 考察

4.1 開発文書品質の評価次元

Wang は情報品質特性として、固有品質、利用品質、文脈品質、表現品質の 4 項目を挙げている [25]。これらの品質特性ごとに、以下に示すように複数の次元を示している。

固有品質の次元には正確性、客観性、信用性、評判性がある。利用品質の次元には接近容易性、セキュリティがある。文脈品質の次元には、適切性、付加価値、適時性、完全性、適量性がある。表現品質の次元には、解釈性、理解容易性、簡潔性、一貫性がある。

開発文書品質についても、同様の品質特性と、対応する次元を定義すべきである。

IEEE std .830 では、要求仕様が満たすべき性質として、①正当性、②無曖昧性、③完全性、④一貫性、⑤優先順位付け、⑥検証容易性、⑦修正容易性、⑧追跡性を挙げている。

前述した ISO 9126 の品質特性も考慮することにより、開発文書品質の次元を定義することが重要になる。

4.2 開発文書コミュニケーションのモデル化

図 1 に示したように、開発文書の位置付けと品質について、関係者を識別し、関係者同士がどんな情報をやり取りするのかという視点で考えることができる。開発文書は、関係者間のコミュニケーション手段の一つであり、コミュニケーションでは相手と個々の目的が存在すると考えられる。また、文書という形のあるもので残せるという特徴から、記録の仕方や内容を工夫すれば、コミュニケーションの継続性を確保できる。

そこで、付表 1 のように、ソフトウェア開発の世界での関係者と、それぞれの関係者同士の間でかわされる情報を整理することができる。付表 1 では、横方向がコミュニケーション情報の発信者で、縦方向が受信者である。これらの交点が、それぞれの関係者同士で受信されるべき情報のないようであり、目的達成のための情報と、継続性のための情報にわけて整理している。これらの情報を整理することで開発文書の目的が明確になり、対応する品質特性を定義できるから、その文書品質特性を評価する方法も定義できるようになる。

5 開発文書品質研究の展望

5.1 研究課題の整理

研究会が取り組む開発文書品質の課題である①文書品質、②計測技術、③品質向上方法に対して、情報品質の研究で挙げられている①品質価値の解明、②品質特性の解明、③品質基準の確立、④品質管理の確立、⑤品質保証の確立、⑥コミュニケーションという 6 課題を比較すると、表 1 のような 3 行 6 列の行列として研究課題を整理できる。

表 1: 開発文書品質研究課題

開発文書品質の課題	技術		
	文書品質モデル	計測技術	品質向上技術
品質価値の解明	品質価値の効果	品質価値の計測	価値向上方法
品質特性の解明	品質特性	特性測定方法	品質特性の展開方法
品質基準の確立	標準化、評価指標	基準値の定義方法	品質基準の準拠方法
品質管理の確立	品質目標分析法	品質管理外リクス	品質管理プロセス
品質保証の確立	品質保証モデル	品質保証外リクス	品質保証プロセス
コミュニケーション	品質欠陥モデル	可視化方法	リスク対策プロセス

5.2 開発文書コミュニケーションで交換される情報

開発文書を使った関係者同士のコミュニケーション活動で交換される情報を付表 1 のように整理することにより、それぞれの目的に合わせて交換される情報の品質特性や評価の方法を定義できる。ただし、今回の整理で定義した関係者はあくまで、一般的な開発組織を考えた場合のものである。このため、実際の組織で

は、その業務ドメインの特性や、組織特有の事情により、関係者の構成が異なるケースが考えられる。この表を使えば関係者間で交換される情報単位で整理できるので情報レベルの関連性を定義できる。これにより、各組織の事情に応じて開発文書の構成や、各文書の構造を定義する際の指標を定義できる。つまり、組織ごとに、ひとつの文書単位とするための「情報の塊」を決定することが容易になる。各組織では、それぞれの情報の重要性や、つながりから、その結合度、凝集度を評価することにより、各組織で最適な文書単位を決定できるようになる。

5.3 現代ソフトウェア開発における文書の範囲

現代の規範的なソフトウェア開発プロセスにおいては、全ての工程のあらゆる側面の文書を作成することになっている。この場合、すべての開発の課題は文書に投影されている、あるいは投影されなければならない。または、文書がない、文書に書いていないことが課題となる。

開発におけるある課題を、文書の問題と捉えるのか、あるいは開発における本質的な問題と捉えるのかで改善活動の視点やその適用範囲が異なってくる。開発における課題の全てを、文書の観点からだけで解決できないことも明らかである。一方で、純然たる文書化の問題もあり、整理が必要である。

さらに、昨今、ソフトウェア開発の状況や方法が劇的に変わりつつある中で、これまで以上に大規模な開発を効率よく行い、同時に品質を確保することが求められている。このときの開発手法は、伝統的な開発プロセスやアジャイル手法ではないかもしれない。ソフトウェア開発者やプロジェクトマネージャの記録（開発の成果、記録、プロジェクトの規範、記録、ミーティングの記録、メール等のデータ）はどのようなものであるべきなのか、成果物や記録をどのように活用できるのかについて、今後のソフトウェア開発の方向性を見定めなければならない。この議論の観点には、現代におけるエンジニアのライフスタイル、多くの拠点に多様なメンバが集まるプロジェクトのスタイル、オープンソースの活用、開発環境、ツール、セキュリティ、品質、コストなどがある。

6 まとめ

本稿では、開発文書品質についての関連研究動向を整理し主な研究課題について述べた。しかし、開発文書品質研究のロードマップとしてまとめるまでには至っていない。

今後は研究会のロードマップ部会でこれらの課題について継続的に整理しロードマップを明らかにしていく予定である。

参考文献

- [1] システム開発文書品質研究 (ASDoQ), <http://asdoq.jp/>
- [2] 下村耕平, これからの技術文書—国際化時代の技術者・翻訳者必携, 競争に勝つドキュメンテーションの論理と方法論, アルカ, 1987
- [3] 花田收悦, ソフトウェアの仕様化と設計, 日科技連, 1986

[4] IEEE std. 830-1998, Recommended Practice for Software Requirements Specifications, 1998

[5] Volere, <http://www.volere.co.uk/template.htm>

[6] Volere, <http://www.volere.co.uk/>

[7] Scenario Plus, <http://www.scenarioplus.org.uk/>

[8] OMG SysML, <http://www.omgsysml.org/>

[9] 清水吉男, 要求の仕様化入門, pp. 1-53, ソフトウェアピープル, vol. 4, 技術評論社, 2004

[10] Len Bass, Paul Clements, Rick Kazman, Software Architecture in Practice, 前田他訳, 実践ソフトウェアアーキテクチャ, 日刊工業新聞社, 2005

[11] Elizabeth, H., Jackson, Dick, J., Requirements Engineering, Springer, 2002

[12] EARS, Easy Approach to Requirements Specification, RE2010, 2010

[13] 国友義久, 効果的プログラム開発技法, 近代科学社, 1983

[14] Bansiya, J., Davis, C.G., A Hierarchical Model for Object-Oriented Design Quality Assessment, IEEE Trans. On Software Engineering, Vol. 28, No. 1, pp. 4-17, 2002

[15] N.E. Fenton and M. Neil, Software Metrics: Roadmap, Future of Software Eng., A. Finkelstein, ed., ACM Press, 2000

[16] ISO 9126 - Software Engineering Product Quality-part1: Quality Model, 2001

[17] Arthur, J. and K. Stevens, "Assessing the Adequacy of Documentation Through Document Quality Indicators," *Proceedings of the International Conference on Software Maintenance*, 1989 中條武志, 情報の流れに着目した設計開発プロセスの標準化, Journal of Japanese Society for Quality Control, Vol.35, No.2, pp.142-149, 2005.

[18] Andreas Dautovic, Reinhold Plösch, Matthias Saft, Automatic Checking of Quality Best Practices in Software Development Documents, 2011 11th International Conference On Quality Software, pp.208-217, 2011

[19] 関口恭毅編著, 情報品質の研究, 日本情報経営学会叢書, 2009

[20] Wang, R.Y., A Product Perspective on Total Quality Management, CACM, Vol.41, No.2, pp.58-65, 1998

[21] Anna Wingkvist, Morgan Ericsson, Rudiger Lincke and Welf Lowe, A Metrics-Based Approach to Intrusion Detection System Evaluation for Distributed Real-Time Systems, International Parallel and Distributed Processing Symposium, 2002

[22] 山本修一郎, CMC で変わる組織コミュニケーション, NTT 出版, 2010

[23] Alain Abran, Software Metrics and Software Metrology, IEEE Computer Society, 2010

[24] 山本修一郎, 独立検証及び妥当性確認と形式手法がもたらすソフトウェア開発プロセスの高信頼化, SEC journal Vol.6 No.3, pp.34-37, Oct.2010

[25] Richard Y. Wang, A Product Perspective on Total Data Quality Management, COMMUNICATIONS OF THE ACM February 1998/Vol. 41, No. 2, pp.58-65, 1998

付表 1: ソフトウェア開発における関係者と, 受発信される情報の例

受発信	顧客(ユーザ)	開発管理者	分析者	設計者	実装者	評価者	
顧客(ユーザ)	目的	・自分が必要としているものの探究 ・背景情報の整理	・体制 ・開発者の能力 ・開発の流れ	・目的の理解内容 ・システム導入後の顧客(ユーザ)に対する変化(効果) ・制約/条件に基づくトレードオフ項目 ・将来性に対する理解内容			・評価による保証範囲と、結果 ・合格結果のエビデンス
	継続	・開発に至った経緯、背景					
開発管理者	目的	・顧客制約/条件 - 予算、期間など - 開発方法(指定があれば)	・予算、開発進捗状況(開発継続判断) ・開発に関連するリスク		開発/体制の見直し	・作業進捗	・評価可否判定
	継続		・開発記録(作業内容、時間経過、コスト経過、リスク変化など)				
分析者	目的	・開発の目的/背景 ・対象組織の特徴/文化 ・開発対象の用途	・顧客が望むシステムの追及 ・対象システムの将来性の評価	・理解内容と、その結果(要求への準拠度合)			
	継続		・開発の背景情報と、その考慮ポイント ・判断結果までの思考過程(連続性)	・設計上の制約			
設計者	目的	・動作環境 ・利用環境 ・対象ユーザ ・利用可能資源	・顧客制約/条件 ・組織制約/条件	・開発システムの目的 ・満たすべき条件 ・遵守すべき制約	・対象システムを制約条件下で成立させる工夫 ・対象システムの拡張性の検討	・理解内容と、その結果(設計への準拠度合)	・評価結果
	継続				・設計の前提条件 ・設計時の背景情報		
実装者	目的			・実現したい機能 ・技法/工法 ・機能実現、実行に際しての制約/条件 ・外部インターフェース	・作業ルール(コーディング規約など) ・内部インターフェース ・上位設計/意思の順守	・不遵守事項の内容	・評価結果詳細
	継続						
評価者	目的	・顧客制約/条件 - 予算、期間と、設計予算、期間 - 開発方法(指定があれば)	・条件/制約情報 ・評価基準	・評価対象と、項目	・内部構造 ・使用開発環境	・評価計画 ・評価レベル/ポイント ・評価手順	・評価の一貫性(再現性の確保)
	継続						