

ソフトウェア開発の品質，生産性向上に向けた ISO/IEC 25030 制定の意義

江崎和博（法政大学 理工学部 経営システム工学科）

概要 情報・通信ネットワーク技術の進歩にともない，社会基盤や企業の事業競争力強化にとって，情報システムに組み込まれるソフトウェアの品質確保がますます重要な課題となっている．従来から，ソフトウェアの真の品質，生産性の向上を図るためには「源流管理」，すなわち，ソフトウェア開発上流の要求定義の品質を高め，品質を作り込むことが極めて重要であると言われてきた．一方，要求分析は非定型かつ，設計者の技術的洞察性に負うところが多く，最も困難な作業とされる．要求定義で作成される要求仕様書の品質が低いと，発注時に期待されたソフトウェアが実現されない，開発完了後に要求定義の漏れが露呈するなどが発生する．その結果，ソフトウェアの発注者や利用者との間で多くのトラブルを引き起こすことになっている．これまで，ソフトウェア品質評価技術を提供する規格として，ISO/IEC9126及び14598シリーズ（JIS-x0129，JIS-x0133シリーズ）が存在しているが，品質要求定義のための規格がなかった．このためソフトウェアのライフサイクルで計画－実行－評価というPDCAサイクルが回らないという問題があった．このような状況認識の中で2007年，世界ではじめて要求定義プロセスに踏み込んだ国際規格が日本発の国際標準として制定された．筆者は，この規格のプロジェクトコエディタとして規格の開発を推進した．本論文では，このISO/IEC 25030：「ソフトウェアの品質要求定義」について紹介するとともに，そのソフトウェア開発への適用の効果などについて述べる．

1. はじめに

世界規模で進展している IT 革命による産業・社会構造の変革，グローバル化，地球環境問題などへの対応が必要不可欠となっている．こうした社会状況下で，日本が世界の先進国家として生き残り，持続的に発展していくためには，社会を支える情報システム基盤の品質向上が益々，重要な課題となっている．システムの品質を向上させるためには，発注者と開発者がお互いに共通の認識に立ち，受注・発注から設計・開発・納入・保守・運用までの全ライフサイクルで協力し合うことが重要である．そのためには従来からあるハードウェアの品質管理技術に加えて，システムに組み込まれたソフトウェアの品質確保が極めて重要である．ISO/IEC JTC1

（International Organization for Standardization / International Electro Technical Commission / Joint Technical Committee 1：国際標準化機構・国際電気標準会議合同委員会）SC7 WG6 及び（財）日本規格協会では，過去約 20 年間にわたり，一貫してソフトウェア品質の評価や管理技術の研究・開発及び標準化[3-13]に取り組んでいる．

本論文では，近年，国際標準として制定された ISO/IEC 25030「ソフトウェアの品質要求定義」[3]，[4]について紹介するとともに，本規格のソフトウェア品質改善に向け

た適用の意義について紹介する．

2. ソフトウェアの品質

従来，製品の品質といえば，主に信頼性のことを指していたが，ソフトウェアは多面的な性質を持つため信頼性だけでは，ソフトウェアの品質全体を捉えることができない．そこで，ISO/IEC では，従来あった B.W.Boehm[1] や J.A.McCall[2]のソフトウェア品質構造モデルを利用者の視点から整理し，機能性，信頼性，効率性，使用性，保守性，移植性の 6 つの品質特性[10]として定義している．さらに，利用者はソフトウェア導入後の利用効果にこそ興味を持つことから，利用時の品質を有効性，生産性，安全性，満足性の 4 つの品質特性[10]として定義している．ソフトウェアの品質は，この品質モデルの

ソフトウェアは
国際標準化による
品質保証がカギ

視点から，各品質特性毎に対応する測定法を用いて測定し，定量的に評価することによって，その品質を保証し，改善することができる．

2.1 品質要求定義の問題

ここで，ソフトウェア開発における要求定義の問題点について触れておきたい．これまでのソフトウェア開発では，設計段階でソフトウェアに対する機能要求を洗い出し，機能要求の定義は行われてきたが，非機能要求に含まれる品質要求の定義については必ずしも充分に行わ

れていない。結果的に製品納入後、できあがったシステムで、以下のような問題を引き起こしている。

①効率性の視点を欠くと、ソフトウェア開発完了後に性能問題を引き起こし、開発期間や予算の超過が発生する。

②使用性の視pointsの欠如は、使い勝手の悪い使われないシステムを実現し、投資が回収できないこともある。

□保守性の視点を欠くと、システム機能は動作しても、保守・運用に膨大な労力と費用を費やし、かえって業務効率を悪化させてしまう場合がある。

2.2 源流管理の重要性

一般的に製品の品質は上流から作りこめという「源流管理」の原則が知られており、ソフトウェア品質の確保は、図1に示すように、開発前に製品の品質目標を設定し、開発完了後に品質の目標値と測定値の差異を分析することによって可能である。実際の開発現場では、品質目標の設定と、品質目標を達成するためのソフトウェア構造の設計、設計結果に基づくシステム実現性の検証という「源流管理」が、かならずしも十分に実施されていない場合もある。ソフトウェア構造の不備に起因する性能問題の解決のために、最悪の場合、開発を完了したソフトウェアの作り直しにつながることもある。

したがって、真のソフトウェア品質や生産性向上のためには、開発上流の設計段階で、ソフトウェアに対する品質要求の明確化と実現性の検証を行い、品質を作り込むことによって、PDCAを回すことが極めて重要である。

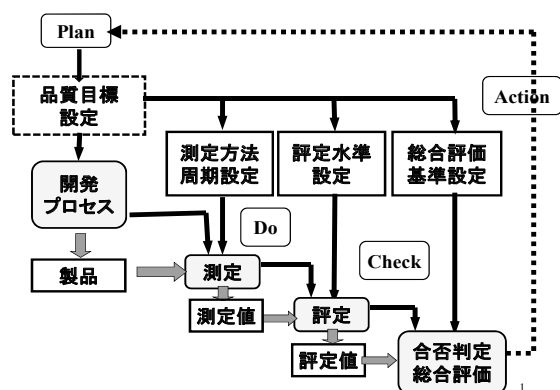


図1 品質目標の設定によるPDCAサイクルの実現

3.ソフトウェア品質の標準化

3.1 ソフトウェア品質の国際標準化

ISO/IEC JTC1/SC7 WG6 では、これまでに、ソフトウェアの品質評価技術の開発と国際標準化[5-13]を進めてきた。ソフトウェア品質評価技術の第1世代標準であるISO/IEC9126 シリーズ(製品の品質, 4分冊)及びISO/IEC 14598 シリーズ(製品の評価, 6分冊)に関しては、JIS X0129 シリーズ[10-13], JIS X0133 シリーズ[9]として、国際規格の制定を全て完了した。

一方、ソフトウェア品質要求定義の研究は立ち遅れて

おり、ソフトウェアを評価するための規格は整備されたが、品質要求仕様書の存在が前提となっており、品質要求の定義には触れてこなかった。

そこで、2003年度から第2世代標準として、従来、カバーで

きていなかった品質要求定義プロセスにもスコープを拡大し、SQuaRE (Software product Quality Requirements and Evaluation, ISO/IEC 25000 シリーズの略) 関連[5]の内、最も新しい提案として、ISO/IEC 25030:「ソフトウェアの品質要求定義」[3][4]の新規開発と制定、スコープの拡大に伴うISO/IEC 25001「計画と管理」[6]の制定、ISO/IEC 25030[3]を前提としたISO/IEC 25040「品質評価参照モデル及び手引き」の開発を進めている。

3.2 ソフトウェア品質の国際標準化におけるISO/IEC 25030の位置づけ

現在、第2世代標準として策定を進めているSQuaREが想定するソフトウェア開発プロセスの枠組みを図2に示す。

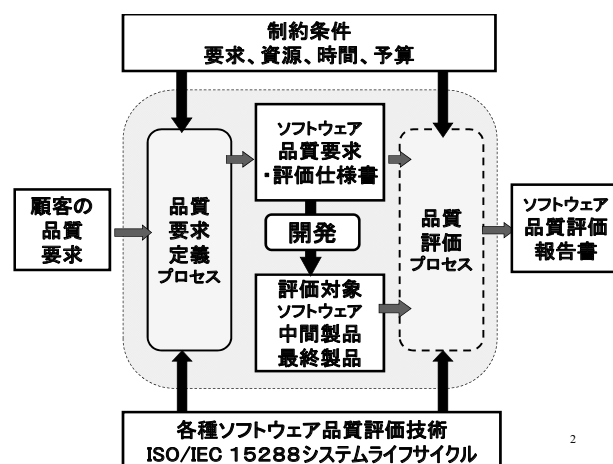


図2 ソフトウェア開発プロセスの枠組み

顧客のニーズは品質要求定義プロセスを経て品質要求仕様書に変換され、さらに品質要求仕様書に基づき開発された評価対象ソフトウェアと品質評価仕様書を入力とし、品質評価プロセスを経て、ソフトウェア品質評価報告書を作成する流れとなる。ここで、評価のための制約条件は顧客のニーズに含まれる開発コストや期間への要求、支援基盤は ISO/IEC 2501n (n は規格の枝番号)、ISO/IEC 2502n 等の品質モデルや測定法から構成される。

ソフトウェア開発プロセス全体における品質要求と評価の枠組み支援するために、現在、第 2 世代標準として策定を進めている SQuaRE 全体の規格の体系は、図 3 に示すように、大きく 5 つの部分から構成されている。

2503n Quality Requirement Division 品質要求部門	2501n Quality Model Division 品質モデル部門	2504n Quality Evaluation Division 品質評価部門
	2500n Quality Management Division 品質管理部門	
	2502n Quality Measurement Division 品質測定部門	

図 3 SQuaRE における品質要求の位置付け

① ISO/IEC 2500n 品質管理部門

ソフトウェア品質の要求と評価技術全体の基本概念を示し、標準の体系、SQuaRE シリーズ共通用語の定義、SQuaRE に基づき、品質の要求定義と評価を行うための計画と管理に対する要求事項などを規定している。

② ISO/IEC 2501n 品質モデル部門

ソフトウェア品質のモデルであり、ソフトウェアを利用者の視点から見た 6 つの品質特性 (機能性、使用性、信頼性、効率性、保守性、移植性)、品質副特性及びソフトウェア利用時の効果を評価する 4 つの品質特性 (有効性、生産性、安全性、満足性) について、現在、見直しを進めている。(現在、開発中の ISO/IEC 25010 制定までは、従来の「JIS X0129-1 第 1 部:品質モデル」、2003. [10] を参照)

③ ISO/IEC 2502n 品質測定部門

ソフトウェアの品質を測定するための内部品質測定法 (Internal Quality Measurement)。ソフトウェアの品質をシステムの振る舞いを通して測定する外部品質測定法

(External Quality Measurement)。ソフトウェアの利用時の効果を測定する利用時の品質測定法 (Quality in Use Measurement) を定義する。(ISO/IEC 25020 以外は現在、開発中であり、制定までは、従来の「JIS X0129-2,3,4」[11-13] を参照)

④ ISO/IEC 2503n 品質要求部門

ISO/IEC 25010 で規定するソフトウェア品質モデル及び、ISO/IEC 2502n [11-13] の品質測定法を利用して、ソフトウェアの品質要求定義を行うための要求事項を規定し、2007 年に ISO/IEC 25030「ソフトウェアの品質要求定義: Quality Requirements and Guide [3] が制定された。

⑤ ISO/IEC 2504n 品質評価部門

品質モデルや品質測定法を用いて、ソフトウェア品質の評価を行う場合の要求事項を規定する。本規格は現在開発中である。

SQuaRE は現在、現行の ISO/IEC 9126 及び ISO/IEC 14598 シリーズの見直し作業を進めており、最終的には 14 分冊で構成される ISO/IEC 25000 (SQuaRE) シリーズとして制定・発行する予定である。

4. ソフトウェア品質の要求定義

4.1 ISO/IEC 25030 の概要

ISO/IEC 25030[3] の目次構成を図 4 に示す。第 5 章にソフトウェア品質要求定義の前提となるステークホルダーやシステムの基本概念、第 6 章に品質要求を仕様化するための基本要素事項を規定しており、以降にその要点を解説する。

1	Scope	
2	Conformance	
3	Normative references	
4	Terms and definitions	
5	品質要求の基本概念	基本概念
5.1	ソフトウェア及びシステム	
5.2	ステークホルダーとステークホルダーの要求	
5.3	ステークホルダー要求及びシステム要求	
5.4	ソフトウェア品質モデル	
5.5	ソフトウェアの特質	
5.6	ソフトウェア品質測定モデル	
5.7	ソフトウェア品質要求	
5.8	システム要求の分類	
5.9	品質要求のライフサイクルモデル	
6	品質要求に対する要求	基本要素事項
6.1	一般要素事項とその前提	
6.2	ステークホルダー要求	
6.3	ソフトウェア要求	
Annex		
Bibliography		

25030より引用

図 4 25030 の目次構成

ソフトウェアの品質評価では、既に存在するソフトウェアの属性に着目し、ソフトウェアの品質を評価すればよい。一方、ソフトウェアの品質要求定義では、諸々のステークホルダーからの要求が必ずしも、ソフトウェアのみに限定されない。そこで、ISO/IEC 25030[3]では、ビジネスプロセス、システムに対する要求も含めて、ソフトウェアを含む広義のシステム全体の品質モデルの明確化が必要となり、システムの領域に踏み込んで定義した。ISO/IEC 25030[3]に規定された、ソフトウェアを巻き込むシステム全体の品質モデルを図5に示す。

システム全体の品質モデルでは、ソフトウェアをコンピュータシステムの構成要素の一部とみなし、ソフトウェア以外のハードウェアやデータの品質モデル、情報システムを取り巻く人間系業務プロセス、機械系システムなどの品質モデルの相互の位置づけなどを示している。

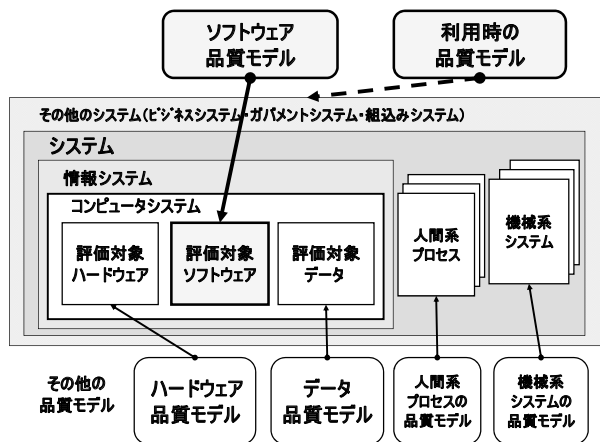


図5 システム全体の品質モデルの概

4.2 ソフトウェアライフサイクルモデル

ISO/IEC 25030[3]に規定されたソフトウェア品質要求と評価のライフサイクルの概念を図6に示す。

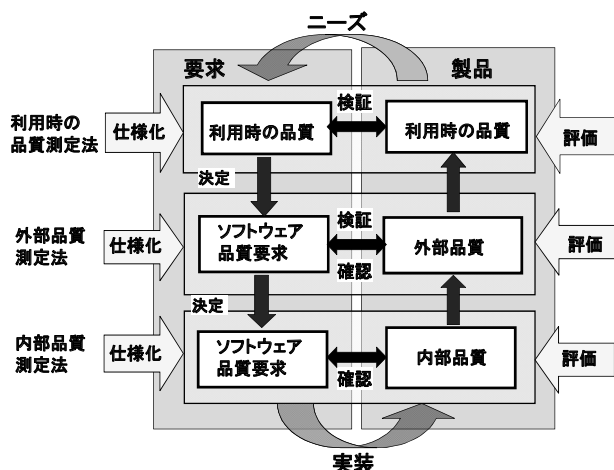


図6 ソフトウェアライフサイクル

ソフトウェア品質の要求と評価には表裏一体の関係がある。開発初期の段階で、下記の3つの測定法に基づき、ソフトウェアに対する定量的な品質目標を定義することができれば、この全く同じ視点から、ソフトウェア品質の評価を行うことができる。

- ① 「利用時の品質測定法」による利用時品質の評価
- ② 「外部品質測定法」による外部品質の評価
- ③ 「内部品質測定法」による内部品質の評価

ISO/IEC25000 シリーズに 品質モデルの すべてが網羅

4.3 システム要求の構造

システム全体の品質モデルの概念に基づき、ISO/IEC 25030[3]に規定したシステム要求の構造を図7に示す。

諸々のステークホルダーが持つ問題点や課題から派生したシステムに対する明示的、暗黙的ニーズから、ソフトウェアへの要求を抽出する。

ソフトウェアに対する要求は、機能要求と非機能要求に分解することができる。非機能要求は、さらにソフトウェアに対する品質要求と、それ以外の価格や納期などの要求に細分化できる。

システムへの要求	ソフトウェアへの要求	ソフトウェア製品への要求	ソフトウェア固有の特質への要求	機能要求	
				ソフトウェア品質要求	利用時品質への要求
					外部品質への要求
					内部品質への要求
	開発への要求	付加された特質への要求	価格、納、製品の将来性、供給ベンダーなどに対する管理的な要求		
		開発プロセスへの要求			
ソフトウェア以外に対する要求	開発組織への要求	開発組織への要求			
		コンピュータハードウェア、データ、機械部品、人的ビジネスプロセスを含む要求など			

図7 システム要求の構造

4.4 品質要求定義のプロセス

ISO/IEC 25030[3]に規定されたソフトウェア品質要求定義プロセスでは

- ① まず、諸々のニーズ全体を洗い出す。
- ② 次に図 8 に示すように、洗い出したニーズを 25010「ソフトウェア品質モデル」[10]の視点から仕訳し、システムの品質に対する要求と、システム品質以外のソフトウェア機能に対する要求、人間系業務プロセスに対する要求などに分解する。
- ③ 次に、図 8 に示すように、抽出したシステム品質に対する要求を ISO/IEC 2502n「ソフトウェア品質測定法」[11-13]の視点から、ソフトウェアの品質に対する要求と、ソフトウェア品質以外のハードウェア品質に対する要求などに識別する。
- ④ ソフトウェア品質要求については、25010「ソフトウェア品質モデル」[10]の 6 つの品質特性、副特性の視点から要求の過不足を明確化すると共に、要求の網羅性を確保し、ISO/IEC 2502n[11-13]の品質測定法に基づいて、各品質特性、副特性毎に具体的な測定法を選択し、定量的な品質目標と評定水準を定義する。
- ⑤ 各品質特性、副特性に対応する測定法毎に設定した具体的な品質目標を、品質要求仕様書として仕様化する。

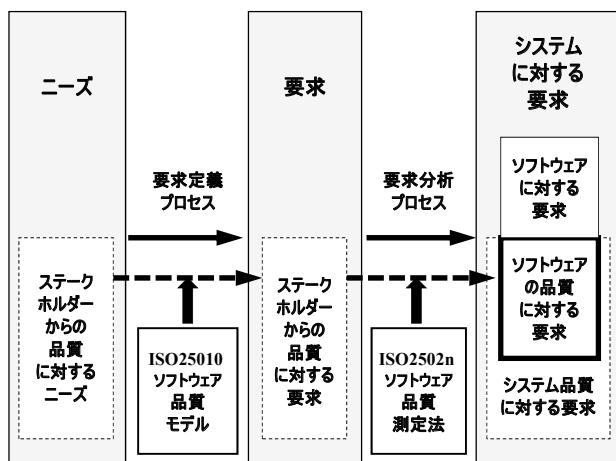


図 8 品質要求定義のプロセス

以上が、要求定義プロセスの概要であり、ISO/IEC 25030[3]には、各アクティビティ毎に、ソフトウェア品質モデルの視点から、適切なソフトウェア品質要求定義のための具体的な要求事項を規定している。

また、本規格はソフトウェア品質要求定義プロセスとして、ISO/IEC 15288「システムライフサイクルプロセス：System Life Cycle Processes」[14]に規定されたシステムの要求定義や分析プロセスも参照している。

4.5 品質評価への ISO/IEC 25030 の適用

一般的に、ソフトウェア品質の評価は、製品の取得者や開発者、第三者の評価者によって、製品の開発完了後に行われる。一方、「源流管理」の考え方に基けば、開発初期の設計段階でも、取得者と開発者との間における

ソフトウェア品質要求に関する合意の形成、開発者による品質目標の目的適合性と実現性の見極めを実施する必要がある。

ここで、開発前の設計審査段階では、品質要求仕様書に仕様化された品質目標の妥当性を利用時の品質測定法、外部品質測定法、内部品質測定法の視点から評価することができる。

表 1 に、ISO/IEC 25030[3]に基づいて定義する定量的な品質目標と品質測定法の例を示す。品質要求定義では、この例に示すような品質測定法に基づく定量的な品質目標の明確化と仕様化が可能である。

また、品質評価では、評価対象ソフトウェアを品質評価仕様書に記述された品質測定法を用いて測定し、設定した品質目標値と測定値との差異を分析することによって可能である。

表 1 定量的な品質目標の例

品質特性	品質目標の説明	品質測定法による品質目標の事例
機能性	ソフトウェア製品への要求に対する機能の充足度	マニュアルと動作の合致度(%) =一致する機能数/マニュアルに記述された機能数
信頼性	ソフトウェア製品の潜在バグの少なさ	障害密度: fault density(件/KLOC) =抽出障害件数/生産物の規模
使用性	ソフトウェア製品の操作や理解のし易さ	エラーメッセージ充足率(%) =エラーメッセージ数/コマンド入力数
効率性	ソフトウェア製品の応答時間や処理時間	レスポンスタイム(sec) =ボタン押下から画面表示までの平均時間
保守性	ソフトウェア製品のメンテナンス作業のし易さ	ソースコードコメント率(%) =コメントの数/プログラムのソースコード数
移植性	ソフトウェア製品のインストールのし易さ	インストール(sec) =製品のインストールにかかる平均時間

次に、ISO/IEC 25030[3]に準拠して、品質要求定義が行われた場合のソフトウェア開発で想定できる品質評価のためのプロセスを示す。

① 評価要求の定義

ソフトウェア品質の評価に対する要求の定義では、図 9 に示す、a. ソフトウェア評価目的の確立、b. ソフトウェアへの品質要求の取得、c. 評価対象ソフトウェアの特定、d. 品質評価の厳格性の定義などを実施する必要がある。

る。

このプロセスでは、b.c.d.のアクティビティで、既に、ISO/IEC 25030[3]に準拠して設定された品質目標の存在を前提とした作業が可能となる。この結果、評価仕様の作成が容易になり、大幅な作業精度の改善と効率化が期待できる。

② 評価の仕様化

品質評価の仕様化では、e. 測定法の選定、f. 測定のための判定基準の明確化、g. 評価のための判定基準の明確化を実施する。このプロセスでは、e.f.g.の全てのアクティビティで、ISO/IEC 25030[3]を活用した定義済の品質目標、品質測定法を再利用することができる。

③ 評価の設計

品質評価の設計では、h. 品質評価計画の作成を行う。品質評価計画書には、前記、評価の仕様化で明確化した品質目標や評価項目、品質測定法、判定基準などが含まれる。

④ 評価の実施

品質評価の実施では、前記、評価の仕様化で明確化した品質要求や評価項目、品質測定法、判定基準などを活用して i. 測定、j. 測定のための判定基準の適用、k. 評価のための判定基準の適用を行う。

⑤ 評価の終結

評価の終結では、l. 評価結果の審査、m. 評価報告書の作成、n. 評価データの蓄積を行う。

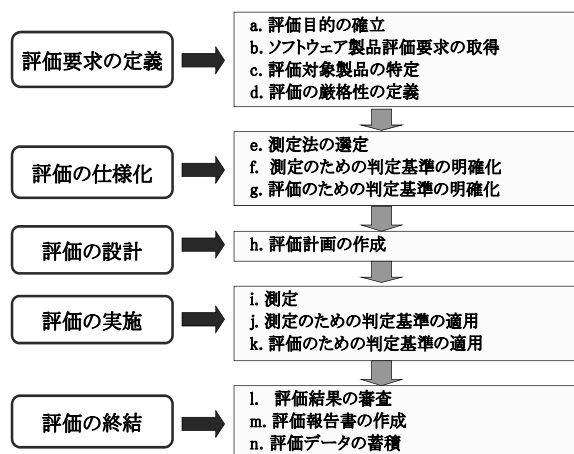


図9 ソフトウェア品質評価のプロセス

上記のプロセスにより、ISO/IEC 25030[3]に準拠して仕様化された品質要求仕様書の存在を前提とする品質評価作業が可能となる。

5. おわりに

近年、ソフトウェアの品質・生産性の向上策として、ソフトウェア開発のプロセスに着目し、プロセスを改善する取り組みが活発である。しかしながら、開発プロセスの改善は、本質的にプロセスに対する入出力プロダクトの品質と開発プロセスの関係を説明することで、はじめて可能と考えられる。そこで、ソフトウェアの真の品質や生産性の向上に向けて、今回、紹介した品質要求定義及び評価技術の実社会への適用、普及促進による「源流管理」の実現が不可欠と考えられる。また、そのためには今後、ISO/IEC 25001「計画と管理」[6]に基づく組織的な品質要求と評価管理のフレームワークをソフトウェアの開発組織に定着させて行く必要がある。

さらに、ISO/IEC 25030[3]は非機能要求に含まれる品質要求の定義を主目的とするが、その基本概念は、機能要求の分析、網羅性の確保などにも適用できると考えられ、近年、取り組みが進められている要求工学の手段としても有効性が高いと考えられる。

る。

ソフトウェアの品質評価技術の開発及び標準化への取り組みは 1985 年にドイツのミュンヘン会議で開始されてから早 20 数年を経過したが、未だ発展の途上である。

今後 100 年の大計を睨み、ソフトウェア開発における品質管理技術の進歩が IT を基盤とする人類社会の形成と安定的な運営には無くてはならないものとして引き続き研究と標準化活動を継続して行く必要がある。

謝辞 本論文の作成にあたり、早稲田大学理工学部の東基衛教授並びに情報処理学会情報規格調査会 SC7/WG6 の委員の皆様様に深謝いたします。

参考文献

- 1) Boehm, B.W. et al: Quantative Ev. of Software Quality, 2nd ICSE, pp.596-605 (1976).
- 2) McCall, J.A. et al: Factors in Software Quality, RADIC, TR-77369 (1977).
- 3) ISO/IEC 25030: Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Quality requirement, Int'l Organization for Standardization (2007).
- 4) Boegh, J.: A New Standard for Quality Requirements, IEEE Computer Software, vol. 25 no. 2, pp.57-63 (2008).

- 5) ISO/IEC 25000: Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE, Int'l Organization for Standardization (2005).
- 6) ISO/IEC 25001: Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Planning and Management, Int'l Organization for Standardization (2007).
- 7) ISO/IEC 25020: Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Measurement reference model and guide, Int'l Organization for Standardization (2007).
- 8) 日本規格協会:「JIS X0129 ソフトウェア製品の評価:品質特性及びその利用要領」(1994).
- 9) 日本規格協会:「JIS X0133-1 第1部:全体的概観」(1999).
- 10) 日本規格協会:「JIS X0129-1 第1部:品質モデル」(2003).
- 11) 日本規格協会:「JIS X0129-2 第2部:内部品質測定法」(2006).
- 12) 日本規格協会:「JIS X0129-3 第3部:外部品質測定法」(2006).
- 13) 日本規格協会:「JIS X0129-4 第4部:利用時の品質測定法」(2006).
- 14) ISO/IEC 15288: Systems engineering - System life cycle processes (2002).

江崎和博 (えさき かずひろ) (非会員)

E-mail: kees959@hotmail.com

元(株)荏原製作所 情報・通信・制御事業本部

品質保証部長, IT 戦略統括部 企画室 副室長

国内・海外情報システム導入プロジェクト管理, 各種情報システムの企画, 設計, 開発, ソフトウェア品質要求定義・評価技術, ソフトウェア工学, 人的要因, ITガバナンスなどの研究を進めている.

技術士 (経営工学部門), 博士 (工学)

ISO/IEC JTC1/SC7 プロジェクトコエディタ,

(財) 日本規格協会 INSTAC 委員,

投稿受付: 2009 年 12 月 18 日

採録決定: 2010 年 2 月 8 日

メンタ: 守安 隆 (東芝ソリューション)