

# 組込みシステム開発における 「利用品質」導入の課題

平 沢 尚 毅

## 1. はじめに

技術の成熟化に伴い製品性能による差別化が難しくなっていく中で、利用者にとっての使いやすさが製品の訴求点として着目されるようになってから久しい。使いやすさを包含した概念として、ユーザビリティやユーザエクスペリエンスなどの用語が浸透しつつある現在、開発プロセスの中に、それらを実現するための方策が試行錯誤されながら展開されている。

これらの方策としては、ISO13407（現9241-210）[1]に代表されるようなプロセスアプローチから浸透していったと考えられる。プロセスアプローチは、現行のプロセスを分析し、ユーザビリティを高めるために必要なプラクティスを改善あるいは付加していく。この結果、ユーザ要求定義や、ユーザビリティ評価といったプロセスの導入が進められるようになっている。

一方、前述のユーザビリティやユーザエクスペリエンスは、利用者から見た品質概念である。この利用者から見た品質は、利用品質（Quality in use）と呼ばれている。ISO9126[2]においては、ソフトウェア品質の特性は、利用品質とソフトウェア製品品質に大別されており、利用品質はソフトウェア品質における重要な位置づけにある。

ところで、ソフトウェア品質に関しては、自動車の安全問題など、社会に与える影響の大きさが認識されつつある。その結果、ソフトウェア品質監査制度[3]を確立する動きや、経産省の指導による情報システムの信頼性プロジェクト[4]が見られる。更に、国際規格では、ISO9126やISO14598[5]などのソフト

ウェア品質に関するいくつかの規格がISO/IEC 25000 -Software product Quality Requirement and Evaluation (SQuaRE) シリーズ[6]として統合されている。このようなソフトウェア品質に関する動勢に対して、様々な概念が錯綜し、結果として開発現場に適応する上で混乱を生じさせる可能性が生まれている。本報告では、このようなソフトウェア品質に関する新たな動きを受けて、利用品質を高めるという観点から、組込みシステム開発に利用品質を導入する場合の課題を考察することを目的とした。

## 2. 利用品質の位置づけ

ISO25010[7]において、利用品質は、下記のように定義されている。

Quality in use is the degree to which a product or system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use.

この定義によれば、利用品質の主体者は特定の利用者であり、その品質特性は効果性、効率性、リスクからの自由度、そして満足度によって構成されることになる。これらの利用品質の特性は、システムに依存した形式になっていない。

一方、ISO25010においては、システムあるいはソフトウェアの製品品質は、次の8つの特性によって構成される。

- ① functional suitability
- ② performance efficiency
- ③ compatibility
- ④ usability

- ⑤ reliability
- ⑥ security
- ⑦ maintainability
- ⑧ portability

これらの特性を、ISO25010の前規格であるISO9126と比較した場合、compatibilityおよびsecurityが増えていることがわかる。

ソフトウェア品質の改善には、ソフトウェアそのもの、それを生み出すプロセスおよび開発基盤等の特徴を客観的に測定し評価することが求められる。ソフトウェアの測定可能な特徴は属性と定義されている。ソフトウェア属性を考えるには、製品の品質・規模・利用効果の他、プロセス、プロジェクト、組織の特徴等の視点があげられる。属性を測定する方法と測定尺度を合わせた概念はメトリックと呼ばれている。メトリックとは、属性を特定の尺度上の数または名前へ計算または分類によって対応づける方法と「測定尺度」の両方を併せた概念である。対応づけによって得られる値を「測定値」、属性の予測や見積もりに使う測定値は「指標」と呼ばれている。

継続して改善を進めるには、利用品質もメトリックを設定する必要がある。例えば、効率性であれば、システムの利用の際の達成時間を想定できる。ISO9126-4[8]には、利用品質メトリックの事例が紹介されている。理論的には、ソフトウェア品質のメトリックから設定される品質指標の根拠は、利用品質の各メトリックの指標との対応がとられている必要がある（図1）。例えば、ユーザインタフェース設計においてユーザビリティの指標として画面遷移のステップ数が規定されているとすると、ここで設定される基準は、利用品質の基準を満たしている必要があることになる。逆に、利用品質の指標が曖昧であるとすれば、ソフトウェア品質の指標の根拠を明確に説明できないことになる。

しかしながら、ソフトウェアの外部品質の指標は経験則に基づき根拠が明示されない場合が多く、その結果、利用品質メトリックを設定する問題意識は高いとは言えない状況にある。

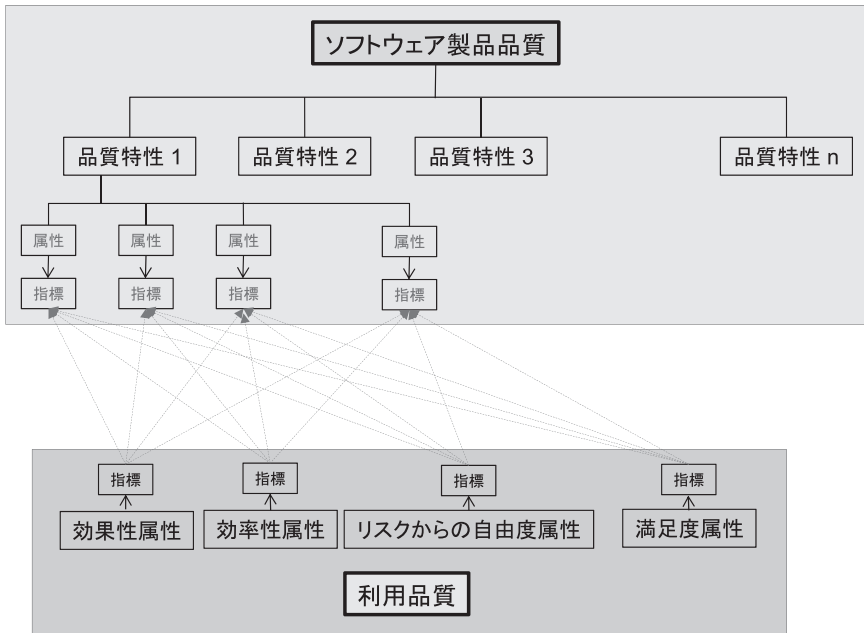


図1 ソフトウェア製品品質と利用品質との対応

### 3. 利用品質をマネジメントするプロセス

IPAのソフトウェアエンジニアリングセンターは、組込みシステム開発プロセスのモデルとしてESPRを提案している(図2)。このプロセスモデルでは、利用品質をマネジメントするプロセスを、製品企画に位置づけ、ESPRモデルのスコープ外としている。一方、SQuaREの品質モデルでは、製品品質、データ品質、利用品質が、ソフトウェア品質モデルとして同レベルに位置づけられている。すなわち、ソフトウェア品質として、3つの品質がマネジメントされる必要性が示されている。

しかしながら、自動車や航空機のように大規模な組込みシステムは、そのシステム構成要素が増大し、利用品質の位置づけが複雑になる。システム構成要素の品質特性は、その下位のソフトウェア構成要素に対して直接に関連するの

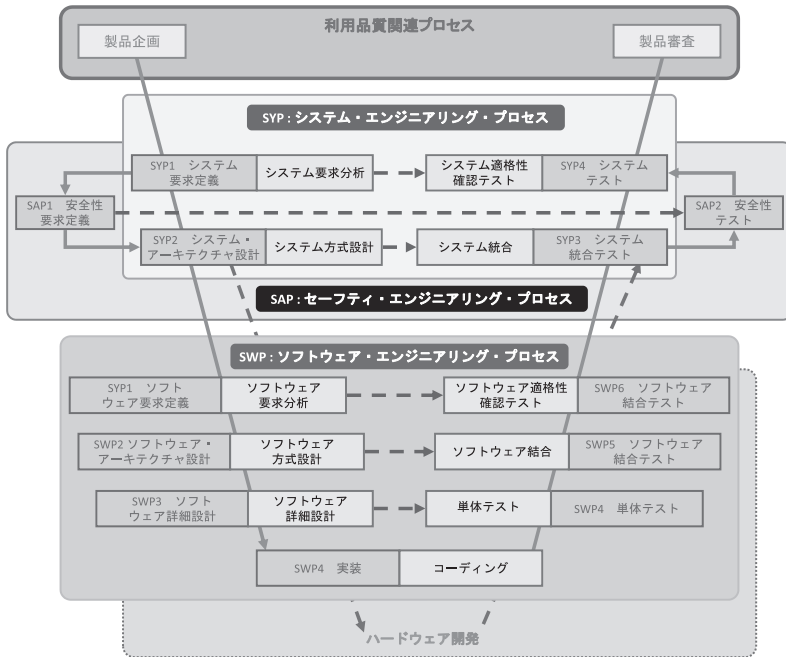


図2 システムライフサイクルプロセスにおける利用品質の位置づけ (ESPR (2007))

に対して、利用品質は、必ずしもシステム構成要素の品質特性とは1対1に関連しない。自動車の例で言えば、走る止まる曲がると言った運転制御のシステムと、ナビゲーションシステムとでは、利用品質は、別々に定義することは難しい。したがって、システムの利用品質をマネジメントする場合は、利用状況とシステム構成要素を対応づけてマネジメントする必要がある。すなわち、利用品質は、複数のシステム・エンジニアリング・プロセスと連動してマネジメントする必要がある。結果として、システム開発を担当するエンジニアにとっては、利用品質がマネジメントスコープの中に設定しにくいことになる。

さらに、利用品質マネジメントに求められる専門性には、心理学や社会学と言った人間科学系のものが含まれるため、一般的なエンジニアにとっては馴染みのないものである。システム・エンジニアリング・プロセスを担当する実務

者にとって、品質に関する現実的な要求は、設計を制御する品質指標の具体的な数値であり、その根拠を明示することは、直接の業務で求められることは少ない。

以上のように、組込みシステムにおいては、利用品質をマネジメントするプロセスが曖昧なことで、現行の開発関係者にとって専門外の知識が要求され、さらに直接の影響を受けないことから、利用品質マネジメントが現行の開発プロセスに導入されることは非常にまれな状況となっていると考えられる。

#### 4. 品質要求

一般的には品質管理であれば、製造プロセスにおいて、予め設定した品質基準に製造物が適合しているかどうかを管理する。また、品質保証であれば、製品が顧客に対して守るべき品質基準を満たしているかどうかを監視することになる。両者ともに、品質基準は既定されている。

一方、製品開発の場合は、従来には無い新たな付加価値を産出しなければならない。新たな付加価値を評価するためには、新たな品質基準を設定する必要がある。ISO25030[10]は、製品開発プロセスにおける品質要求についての規格であり、システム要求定義プロセスの中に、品質要求定義を取り込み、品質要求を抽出することを提案している（図3）。

しかし、組込みシステム開発におけるシステム要求定義プロセスに品質概念を導入することは混乱を与えることになる。それは、品質要求定義と非機能要求定義の関係についてである。

近年、組込みシステムにおける要求定義では、機能要求だけでなく、非機能要求の重要性が強調されるようになってきている。機能要求とは、そのシステムに備わっているべき役割や働きを指し、「～できること」というような表現で記述できるものである。それに対して、非機能要求は、役割や働きの様態を指し、機能をどのようにして実現するかを明示したものである。問題は、この非機能要求とは、品質を含んだ概念であることである。すなわち、非機能要求定義は、

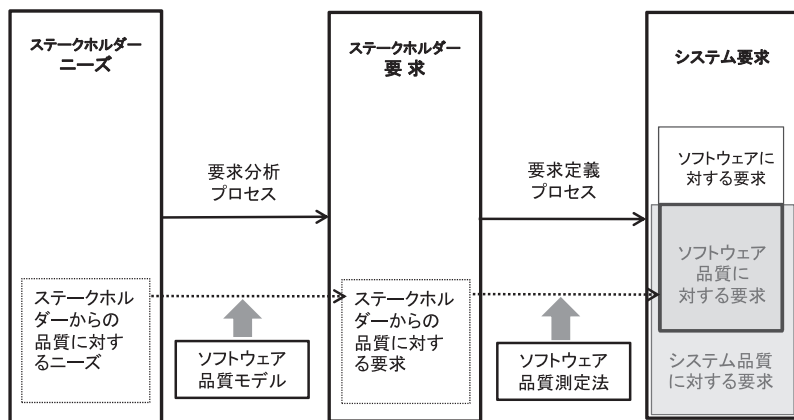


図3 品質要求定義プロセス (ISO 25030 (2007))

品質要求定義と明確な区別がつかない。また、開発現場で使われる「性能」という言葉は、機能を指すのか、非機能を指すのかが曖昧になっている。開発現場では、厳密には、性能、機能、非機能を使い分けているわけではなく、状況によって機能や非機能のニュアンスを意図して使われていると考えられる。

この問題は、後述する組織的な問題と関連することであるが、重要なことは、エンジニアの開発行動中にいかにして品質に関連する行動を取り込むかということである。ここで言う、品質関連の行動とは、アイデアを出しながらフィードフォワード的に進める開発行動に対して、基準を定め評価することによって、開発活動にフィードバックを与えることである。著者らのヒアリングによれば、このような活動は決して目新しいことではなく、通常の開発においては暗黙的に進められている。一般的な開発の上流プロセスでは、デザインレビューがフィードバック活動を担うこととなるが、優れた開発組織の場合、十分な経験を持った開発関係者が的確なフィードバックを与えていると言われる。

したがって、品質要求に関連する本質的な課題は、暗黙となっている開発知識を可視化し、それを共有しながら、開発関係者の開発能力を深化させてゆくことにあると考えることもできる。

## 5. 機能安全マネジメントとの関係

ソフトウェア品質は、図4のように次の手順で評価され、PDCAサイクルが実施されるとされる[11]。

- (1) 開発プロジェクトに先立って品質計画を立案し、品質モデル、品質測定のためのメトリックを設定する。これに基づいて、評価水準、総合評価基準などを設定し、品質目標を設定する。
- (2) 開発プロセスごとの中間成果物について、設定した測定方法によって測定する。
- (3) 測定した測定値を評価水準から評価する。
- (4) 評価された評価値を総合評価基準から評価する。
- (5) 評価結果に基づいて、改善等の対応をする。

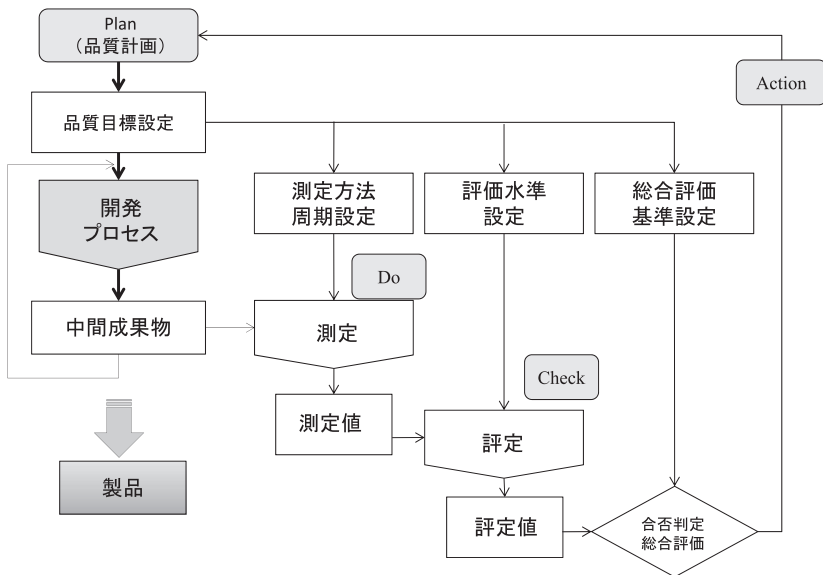


図4 品質目標に基づいたPDCAサイクル ([11]を改編)



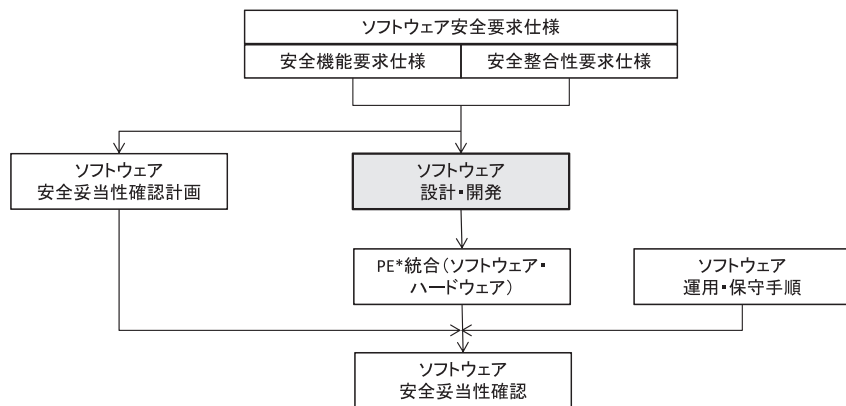
(6) 開発プロジェクト終了後、品質評価結果を総合し、次期の品質計画に反映する。

これらの活動は、開発プロセスを包含する形で実施される。

一方、IEC61508[12]に代表される機能安全を確保するためのプロセスも、同様に開発ライフサイクルを包含したプロセスになっている(図5)。したがって、品質マネジメントのライフサイクルマネジメントと機能安全ライフサイクルは近似したものであると考えることができる。

実際に、安全に関する概念は、ISO25010では、利用品質に含まれるものとなっている。安全要求を創出することは、利用品質の要求する過程であるとも言える。ISO25010には、IEC 60050-191[13]において定義されている dependabilityとソフトウェア品質との対応表があるが、ISO25010の品質モデルがdependabilityの定義を広く包含した形となっている(表1)。

したがって、機能安全を進めるだけでは、安全を確保することは難しく、総合的にソフトウェア品質を確保することが機能安全を強化することにもつながることがわかる。



\*E/E/PE: electrical/electronic/programmable electronics system

図5 機能安全ライフサイクル (IEC61508)

表1 ソフトウェア品質とdependabilityとの対応 (ISO25010 (2010))

	ISO/IEC 25010	IEC 60050-191
利用品質	Effectiveness	*
	Efficiency	*
	Satisfaction	*
	Freedom from risk	Safety
ソフトウェア製品品質	Context coverage	*
	Functional suitability	*
	Performance efficiency	*
	Compatibility	*
	Usability	*
	Reliability ・ Availability	Reliability ・ Availability
	Security ・ Confidentiality ・ Integrity	Confidentiality Integrity
	Maintainability	Maintainability
	Portability	*

## 6. ソフトウェア品質をマネジメントする組織

前述のように、開発活動と品質マネジメント活動ではモノづくりへのメンタルモデルが根本的に異なる。一方は、フィードフォワードであり、もう一方は、フィードバックである。本質的には、それぞれの働きを持つ組織あるいは役割が明確に分離されていることが望ましい。理想的には、図6のように、開発組織と並行して、検査（Inspection）を担う組織があることにより、品質を組織的に統制された形でマネジメントできる。

また、ソフトウェアの信頼を高めるためには、V&Vあるいは、IV&Vという考え方が提唱されている。V&V（検証と妥当性確認）とは、所与の活動の成果物とその活動に対する要求を満足しているかどうか、およびソフトウェアが意図された用途や利用者のニーズを満足しているかどうかを決定するプロセス

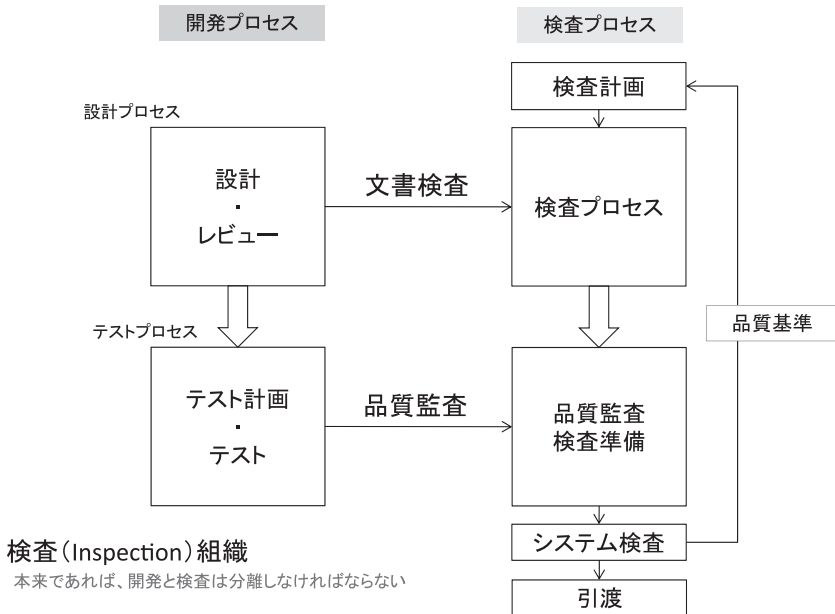


図6 開発プロセスと検査プロセス

であり、Verification（検証）とValidation（妥当性確認）という観点の異なる二種類のプロセスの総称である。一方、IV&Vは、開発組織と品質組織を分離したものである。IV&Vとは、開発組織から技術面、管理面、および財務面で独立した組織が実施する検証と妥当性確認のことである。

V&VもIV&Vも、組織的な統制が必要である。したがって、ソフトウェア品質マネジメントを発展させるには、組織運営コストが必要となることは容易に想像できる。安全性の社会的影響を考え、戦略的に段階を踏まえて組織的な変更を展開することが現実的と考える。

まずは、開発プロセスを定義する段階で、戦略的に品質活動をどのように位置づけるかを明らかにし、その上で、中長期的に組織的な対応をどのように進めるかを検討していくことによって無理なく進めることになる。表2には、組込みシステム開発における品質マネジメント組織的な成熟段階の試案を示す。

表2 品質マネジメントのための組織の成熟度

段 階		状 態
第1段階	混沌とした段階	開発プロジェクトにおいて、品質マネジメントの実態が曖昧な状態。一般的に、テストだけは行われている状態。
第2段階	役割の分離	開発プロジェクトにおいて、設計と評価の役割が明確に分かれている体制によって実施されている。開発中は、それぞれが関与しない。
第3段階	組織上の分離	開発組織と品質マネジメントの組織が分離して存在する。それぞれが関連しながらも、独自のライフサイクルを持つ。
第4段階	検証プロセスの高度化	開発組織よりも高度な検証プロセスがあり、そのための専門家が育成され、手技法が開発されている。通常の開発プロセスでは見過ごすような問題も検出できる。

## 7. ま と め

近年のソフトウェア品質に対する新たな動勢を踏まえて、組込みシステム開発に利用品質を導入する際の課題について検討してきた。利用品質を導入することによって、システム製品の品質に対する説明力は格段に向上することが期待できる。しかしながら、開発現場にとって、利用品質をマネジメントすることの必然性、緊急性に対する理解を得ることは非常に難しいことが確認できた。

同時に、実際の利用品質をマネジメントするには、現行の技術者のコンピタンスでは難しく、新たな人材の育成か、外部リソースを利用せざるを得ない。

また、国内の組込み開発現場では、組織的な体制が整備されることは早急には期待できないが、開発に対するマインドセットを変革しながら、品質要求という考えを踏まえて、開発現場のコンピタンスを段階的に拡張してゆくことは可能である。

いずれにせよ、国内の組込み開発プロセスに利用品質を導入することには大きな壁があることは否めない。開発現場の事情に合わせて、新たな品質ビジョンを設定し、スモールステップで進めて行かざるを得ないことは確かである。

## 参考文献

- [ 1 ] ISO/IEC 9241-210:2010 Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems, 2010
- [ 2 ] ISO/IEC 9126-1:2001 Software engineering - Product quality - Part 1: Quality model, 2001
- [ 3 ] 独立行政法人情報処理推進機構, ソフトウェア品質説明力強化のための制度フレームワークに関する提案 (中間報告), <http://sec.ipa.go.jp/reports/20110930.html>, 2011
- [ 4 ] 経済産業省, 情報政策「情報システムの信頼性」, [http://www.meti.go.jp/policy/it\\_policy/softseibi/](http://www.meti.go.jp/policy/it_policy/softseibi/)
- [ 5 ] ISO/IEC 14598:1999 Information technology - Software product evaluation, 1999
- [ 6 ] ISO/IEC 25000:2005 Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE, 2005
- [ 7 ] ISO/IEC 25010:2011 Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - System and software quality models, 2011
- [ 8 ] ISO/IEC TR 9126-4:2004 Software engineering - Product quality -Part 4: Quality in use metrics, 2004
- [ 9 ] 独立行政法人情報処理推進機構 ソフトウェア・エンジニアリング・センター (SEC) : 組込みソフトウェア向け開発プロセスガイド, 翔泳社, 2007
- [ 10 ] ISO/IEC 25030:2007 Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Quality requirements, 2007
- [ 11 ] 江崎和博, ソフトウェア開発の品質, 生産性向上に向けたISO/IEC25030制定の意義, 情報処理学会デジタルプラクティス, Vol.1, No.2, 20110
- [ 12 ] IEC 61508 ed2.0 (2010-04), Functional safety of electrical/electronic/programmable electronic safety-related system, 2010
- [ 13 ] IEC 60050-191-am2 ed1.0 (2002-01) Amendment 2 - International Electrotechnical Vocabulary. Chapter 191: Dependability and quality of service

