

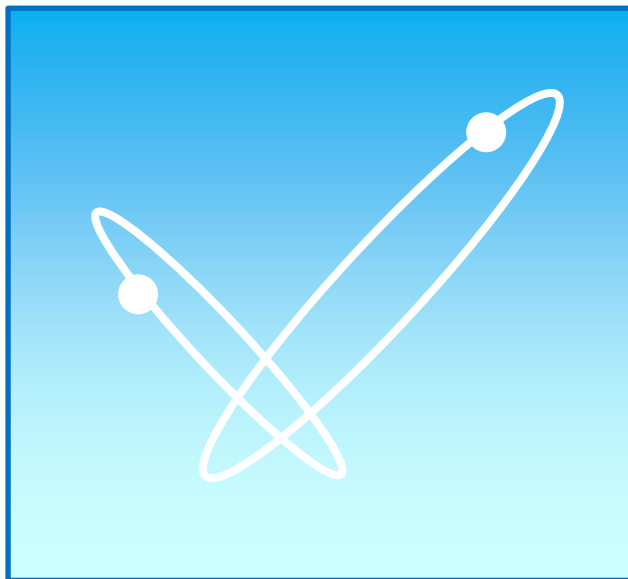
---

# つながる世界の ソフトウェア品質ガイド

あたらしい価値提供のための品質モデル活用のすすめ

---

ダイジェスト版



本資料は「つながる世界のソフトウェア品質ガイド」から一部を抜粋したものです。

掲載されている会社名・製品名などは、各社の登録商標または商標です。

「つながる世界のソフトウェア品質ガイド」  
あたらしい価値提供のための品質モデル活用のすすめ

ダイジェスト版

2015年3月

独立行政法人情報処理推進機構

© Information-technology Promotion Agency, Japan. 2015 All Rights Reserved

# ◆目次

## 第1章 あらためて品質を考える背景

---

1.1	多様化する利用者の期待	.....	P08
1.2	多岐にわたるステークホルダ	.....	P09
1.3	「つながる」システムの拡大	.....	P11
1.4	品質に関する共通言語の必要性	.....	P13

## 第2章 品質を考える時に知っておくべき国際規格

---

2.1	SQuaRE とは？	.....	P16
2.2	SQuaRE の構造と概要	.....	P19
2.3	SQuaRE の品質のとらえ方	.....	P21
2.4	SQuaRE の活用	.....	P25

## 第3章 いま注目したい品質の視点

---

3.1	ユーザビリティと利用時の品質	.....	P28
3.2	セキュリティ/セーフティ	.....	P33
3.3	データ品質	.....	P41



# 第1章

## あらためて品質を考える理由

品質の重要性は、システムやソフトウェアの開発では大変古くから議論されてきた。しかし、今あらためて品質を考え直す必要性が出てきている。この章ではその背景を解説する。

- 
- 1.1 多様化する利用者の期待
  - 1.2 多岐にわたるステークホルダ
  - 1.3 「つながる」システムの拡大
  - 1.4 品質に関する共通言語の必要性
-

## 1.1 多様化する利用者の期待

多くの開発現場では、機能そのものが利用者の期待であるという考えをもとにして、想定している機能が動作しない危惧のみを考え、機能を正しく動くようにすること(システム/ソフトウェアのバグをゼロにすること)が主要な品質であるという考え方が一般的だったのではないだろうか。

しかし、IT を活用した製品<sup>1</sup>の種類や社会における役割が増えるにつれ、利用者の期待は、機能だけにとどまらず、利用目的や利用場面に応じて、安全性・セキュリティはもとより、快適さや楽しさ、また、ビジネスへの高度の貢献といった内容まで含まれるようになってきており、非常に多様化している。必要な機能が使えるというだけではよい製品であるとはいえず、利用者が必要とする機能を「高い満足度」で使えることが強く求められようになっている。

製品を提供する事業者が、安心できる生活基盤を整備し、また競争力のある製品を生み出すには、上記のように利用者の多様な期待を分析し

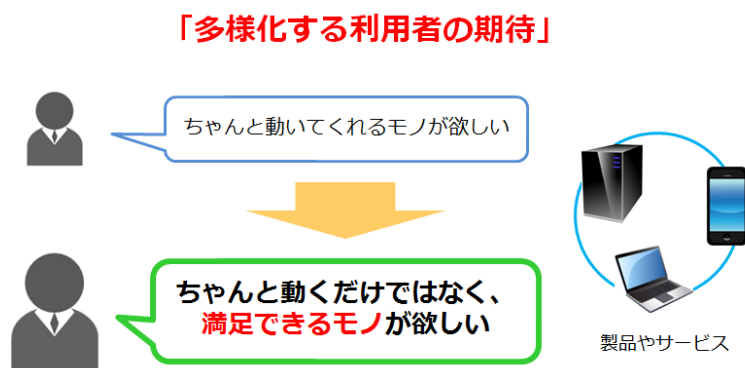


図 1.1-1 多様化する利用者の期待

<sup>1</sup> 量販品、特定用途のシステム、サービスを含む(以下同様)。

た上で、ビジネス上の戦略に沿って実現すべき品質要求<sup>2</sup>を明らかにし、設計・実装・試験・保守によって具現化し、かつ保証することが求められている。

## 1.2 多岐にわたるステークホルダ

現在の製品では、利用者それぞれの期待が多様化しているということに加えて、ひとつの製品やサービスが対象とすべきステークホルダが従来よりも広範囲になってきているという状況がある。ステークホルダ<sup>3</sup>とはその製品に何かしらの関与を持つ人のことを言い、本書では製品の直接的な関係者だけでなく、間接的に影響を受ける人も含む広い意味で用いる。例えばソフトウェアの利用者、開発者、発注者は代表的なステークホルダであるが、その他にも、システムの運用担当者、販売者なども含まれる。以前から、様々なステークホルダが存在していたことには違いないが、製品そのものが、複雑化し、かつ高機能化したことで、開発のサプライチェーンや運用に関わる企業の増加があり、さらに、その製品に関わる多様な利用者の存在など<sup>4</sup>、考慮すべきステークホルダは確実に増えている。これらのステークホルダの存在を正しく理解し、その期待に応えることが、これからの製品開発、ひいてはビジネスの成功には欠かせない。

製品に対する期待はステークホルダの立場によって異なる。例えば、利用者の立場からは必要な機能が使い易く、安全に提供されることが重要だが、運用担当者の立場からは、製品が運用環境下で安定して動作し続けること、その監視や修正が容易であるかどうかが重要である。異なる

<sup>2</sup> 品質に関する具体的なニーズを仕様化したもの。

<sup>3</sup> 「利害関係者」と訳されることもある。

<sup>4</sup> 製品の直接的な利用者だけでなく、間接的な利用者も考慮する必要がある。例えば、公共交通システムの利用者は、交通制御システムの不具合で移動できない等の影響を受ける。

ステークホルダの立場からは、必要とする品質への期待の観点も異なってくる。さらに、観点が異なるだけでなく、場合によってはしばしば矛盾する。一方を優先すると、他のステークホルダの満足度を下げってしまうようなことが発生してしまう。

それぞれのステークホルダがどのような品質を期待するのかは、ステークホルダとのディスカッション等を通じて、ある程度把握することができるが、要求を明確化する方法論が十分に活用されていない状況もあり、正確な把握は難しい。また、ステークホルダ自身が認識せず、製品を実際にある場面で使った時に初めて認識するような、暗黙の期待の存在も無視できない。

### 「ステークホルダによって期待はさまざま」

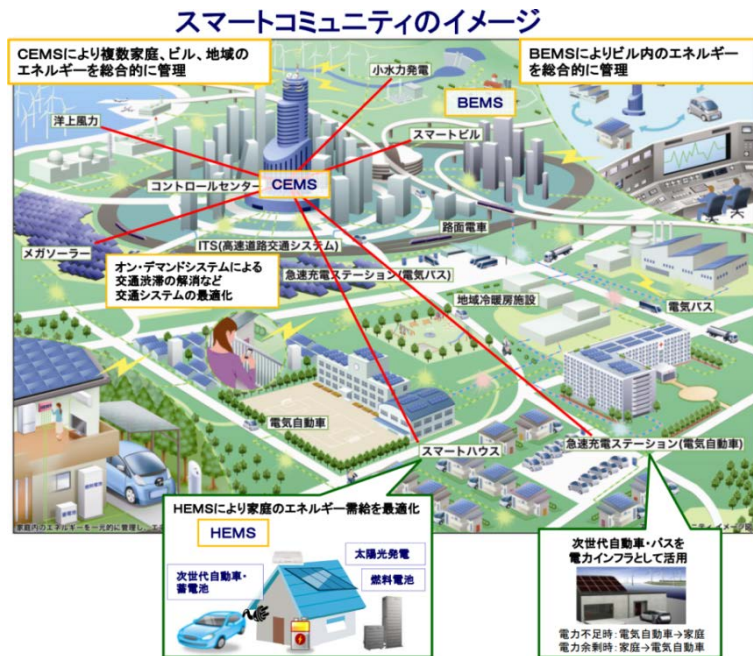


図 1.2-1 製品に対するステークホルダのさまざまな期待



### 1.3 「つながる」システムの拡大

利用者の期待とステークホルダの変化に加えて、これらの変化をさらに加速させる昨今の状況について認識しておく必要がある。それは、これまで単体で提供されてきた製品がネットワークに接続され、かつ相互に連携し合って「つながる」システムとして利用者に価値を提供するような新しい形態が増えているということである。この「つながる」システムには、スマートフォンにインストールされたアプリを通して、センターサーバーやクラウドが提供するサービスを利用するような比較的小さなものから、エネルギー、交通、医療のような社会インフラシステムが相互に連携しながら、全体として効率的な生活基盤システムを提供するスマートコミュニティのような大規模なものまで含まれる。これらの「つながる」システムの普及に対して、利



出典: 経済産業省 [http://www.meti.go.jp/policy/energy\\_environment/smart\\_community/](http://www.meti.go.jp/policy/energy_environment/smart_community/)

図 1.3-1 スマートコミュニティのイメージ

ユーザーは、生活を便利で安全なものにしてくれる、という大きな期待を持っている。同時に、システムの品質に関与するステークホルダも一般的に増加する傾向にある。

さらに、「つながる」システムを開発あるいは提供する事業者にとって、もうひとつ特徴的な問題がある。それは、「つながる」システムにおいては、特定の開発者や運用者が管理できる製品の先に直接管理ができない製品あるいはシステムがつながり、それらシステムの全体として利用者に価値をもたらす状況が広がってきているということである。つまり、開発時点でその利用範囲が明確に決められず、利用者やステークホルダを具体的に定義することが、そもそも難しい場合が頻繁に起こりうる。

このような「つながる」システムの品質を実現するためには、つながっている全ての要素が確実に品質要求を満たしている必要がある。相互の接続に齟齬があると、提供する価値が大きく下がり、場合によっては事故に至ることも危惧される。このため、異なる人・組織の間での製品の品質に関する共通認識や相互理解ができていることは重要である。

### 「『つながる』システム」

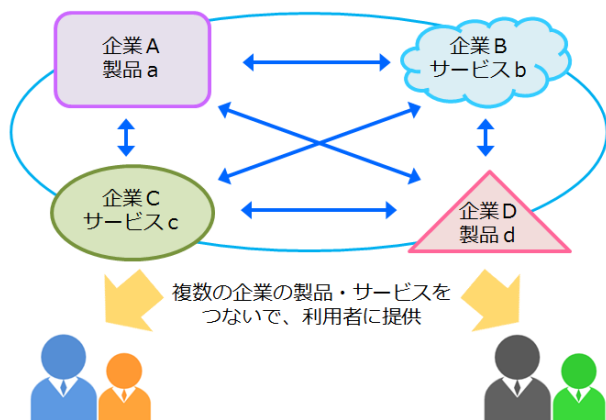


図 1.3-2 「つながる」システムのイメージ

## 1.4 品質に関する共通言語の必要性

ここまで、あらためて品質を考え直す必要性について、その背景を中心に説明した。それでは、事業者は、利用者や多くのステークホルダが期待する品質をどのように具体化し実現すればよいのであろうか。

品質要求の項目は、機能だけでなく、使いやすさ、性能、壊れにくさなど機能以外の項目（非機能項目）も含めて非常に多くの観点から整理する必要があるが、これらを漏れなく洗い出すのは容易ではない。

また、これからの「つながる」システムにおいては、構成する製品がそれぞれ異なる品質要求の下で開発されている場合、それをつなぐ時に、整合していない要求を確認し突き合わせるが必要になる。

この問題に対しては、双方の製品が共通の考え方に基づいて品質保証がされていることが対策の近道となる。それぞれ独立に開発された製品であっても、また事業者が異なる場合であっても、品質を保証する共通の基準があれば、「つながる」システム全体の品質を一貫して確認し説明することが容易になる。言い換えれば、品質に関する共通言語を定義し利用することで、「つながる」システム全体の品質を具体化することが可能となる。

このような課題に対して、現在ではシステム／ソフトウェア開発の品質における共通言語の役割を果たす国際規格／国内規格が体系的に整理されている。本書では、そのような国際規格として、SQuaRE<sup>5</sup>を取り上げ、その考え方と活用方法を紹介する。SQuaREで規定された内容は、「つながる」システムに特化したものではなく、現在の複雑化・多様化した品質を整理するための一つの指針である。また、規格そのものが課題方法を示している訳ではない。しかしながら、このような共通の枠組みを活用するこ

<sup>5</sup> SQuaRE: Systems and software Quality Requirements and Evaluation

とが、「つながる」システムにおける様々な課題に対して今後取り組んでいくための環境作りという意味で、非常に重要であると考えている。

## 第2章

# 品質を考える時に 知っておくべき国際規格

ソフトウェアの品質要求を定義し評価する時の、共通的な考え方を提供するため、国際規格「SQuaRE シリーズ」がある。この章では、規格の概要と基本的な品質の考え方及び活用方法について解説する。

- 
- 2.1 SQuaRE とは？
  - 2.2 SQuaRE の構造と概要
  - 2.3 SQuaRE の品質のとらえ方
  - 2.4 SQuaRE の活用
-

## 2.1 SQuaRE とは？

システム及びソフトウェアの多岐にわたるステークホルダ（利用者、発注者、開発者など）が持つ多様な品質要求を定義し、その実装を評価するための共通の考え方を示す基準のひとつとして、国際規格 SQuaRE シリーズ<sup>1</sup>（以下、SQuaRE）がある。

開発した成果物が要求を満足しているかを確認する場合、候補製品を比較選定する場合、第三者が利用者の立場で客観的な評価を行う場合など、様々な立場のステークホルダが異なる観点から異なる基準で品質に対する要求を出し、評価を行っていたのでは混乱が生じる。

ソフトウェア品質の観点や基準には、唯一無二の正解がある訳ではない。したがって、観点到抜け漏れがないことや理解しやすいこと、基準が実用的かつ客観的な判断につながることなどの基本要件を充足した上で、

### ソフトウェア製品評価標準化の目的と必要性

#### ● 製品評価の目的

- ・ 開発した成果物が要求を満足しているかを確認する
- ・ 候補製品を比較選定する
- ・ 第三者が利用者の立場で客観評価を行う

#### ● 標準化の必要性

- ・ 品質要求に対する合意基盤の確立  
要求を出す側と受ける側の共通理解の促進
- ・ 評価の客観性、反復性、定量性の確保  
結果の信頼性、信憑性の保証
- ・ 受け入れ、選定の拠り所となる評価方法、基準の設定  
評価結果の解釈の一貫性の保証

国際的に広く受け入れられる標準が定められることが重要だとの立場から

<sup>1</sup> SQuaRE: Systems and software Quality Requirements and Evaluation; ISO/IEC 25000 シリーズ、JIS X 25000 シリーズの呼称。スクウェアと読む。

SQuaRE が作られた。

IT システムに対する要求には、システムが果たすべき機能に対する要求事項はそれぞれに異なるが、品質要求はシステムが異なっても比較的同じような観点から求められることが多い。そこに標準化の意味がある。

ソフトウェア品質の国際規格としては、さかのぼれば、1991 年に発行された ISO/IEC 9126<sup>2</sup>があり、(規格を意識せずに使われる場合を含めて) これまでにかかなり広い範囲で活用されてきた。この規格では、ソフトウェア品質の評価観点として六つの品質特性(機能性、信頼性、使用性、効率性、保守性、移植性)が定義され、ソフトウェア品質の要求定義から評価に至る一連の作業の枠組みが規定された。その後、実務への活用のしやすさや、規格利用者の利便性の向上などを考慮して、ISO/IEC 9126 シリーズ(4 分冊)及び 14598 シリーズ(6 分冊)と、それを補完する ISO/IEC 12119 と ISO/IEC 14756 が制定され、それぞれ JIS(日本工業規格)化された。

その後、ソフトウェア技術の進化、利用形態の多様化、人々や社会に及ぼす影響の増大などを考慮して、これらの規格を再編、強化する取組みが行われてきた。これらは、ISO/IEC 25000 シリーズとして、順次、整備されており、通称 SQuaRE(スクウェア)と呼ばれている。従来の規格からの主要な強化点は次の通りである。

- 1) 品質モデル、品質測定量を用いた品質要求定義のプロセスを規定
- 2) ソフトウェアによって処理されるデータ、ソフトウェアを活用して提供されるサービス<sup>3</sup>について、それらの品質特性、品質測定量を規定。
- 3) 品質測定量(例:規模あたり障害件数)の測定値の算出に用いる測定量要素を規定(例:成果物規模、障害件数)

---

<sup>2</sup> ISO/IEC 9126 は、Information technology - Software product evaluation - Quality characteristics and guidelines for their use というタイトルの国際規格で、1991 年に発行され、1994 年にそれを翻訳した JIS X 0129 が制定された。

<sup>3</sup> サービス品質は、2015 年 3 月現在策定中。

なお、ソフトウェア製品評価標準化の進展の過程で、従来の国際規格並びに対応 JIS 規格は、次世代規格の制定に伴って随時廃止され、置き換えられている。

表 2.1-1 <参考>ソフトウェア製品評価標準化の進展

期 区分	<div style="display: flex; justify-content: space-around; align-items: center;"> <span style="width: 15px; height: 15px; background-color: #FFC0CB; border: 1px solid black; margin-right: 5px;"></span> 新規           <span style="width: 15px; height: 15px; background-color: #ADD8E6; border: 1px solid black; margin-left: 20px; margin-right: 5px;"></span> 改訂         </div>		
	第1期 【1985～1991】	第2期 【1992～2004】	第3期 【2005～】
品質モデル	◆ソフトウェアの品質特性の定義	◆ソフトウェアの品質副特性の定義 ◆ソフトウェアの利用時の品質特性の定義	◆品質モデルの再編 ◆ソフトウェアの利用時の品質副特性の定義の追加 ◆品質モデルの再編 ◆データの品質特性の定義 ◆ITサービスの品質特性と品質副特性の定義(審議中)
品質測定		◆品質測定量の定義	◆品質モデルの追加、再編に合わせた品質測定量の追加、改訂 ◆品質測定量要素の定義
品質要求			◆品質要求事項の仕様化プロセスの定義
品質評価	◆品質評価プロセスの枠組みの定義	◆品質評価プロセスの定義 ◆局面、主体者に依じた品質評価プロセスの定義 ◆品質評価ノウハウ蓄積の枠組み(評価モジュール)の定義	◆品質評価プロセスの定義の改訂 ◆局面、主体者に依じた品質評価プロセスの定義の改訂 ◆回復性の評価モジュールの定義
品質管理		◆ソフトウェア品質測定、評価の組織的な技術管理の規定	◆ソフトウェア品質測定、評価の組織的な技術管理の規定の改訂 ◆一連の規格の活用指針の規定

込山俊博 “システム及びソフトウェアの品質基準の体系化”，情報処理，Vol.55，No.1より引用



## 2.2 SQuaRE の構造と概要

SQuaRE は、大きくコア部分と拡張部分に分かれている。コア部分では、システム及びソフトウェアの品質要求定義と評価に関して、基本的な概念、具体的な品質の定義や測定方法、作業要件等を規定している。コア規格の具体的な構成としては、品質管理、品質モデル、品質測定、品質要求、品質評価という 5 つのカテゴリ(部門という)がある。拡張部分(部門)には、それらの応用規格や特定領域をさらに詳細化した規格が制定・計画されている。SQuaRE の全体構造とそれを構成する個別国際規格の概要については、図 2.2-1 及び表 2.2-1 に示した。これらの国際規格は、それ自身改訂・拡充されているとともに、逐次翻訳され JIS 規格として発行されている。

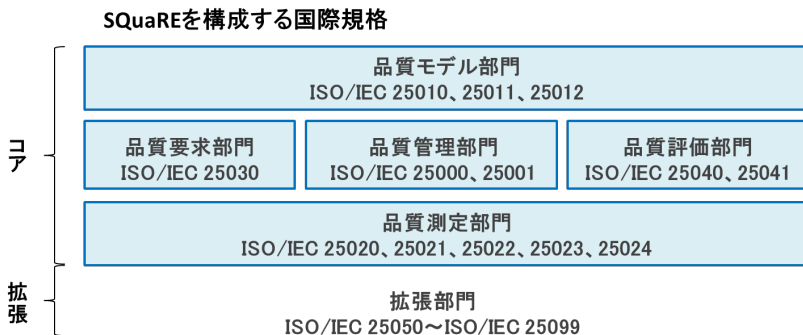


図 2.2-1 ソフトウェア製品評価規格の発行状況(含: JIS)

表 2.2-1 SQuaRE 規格の内容

部門	規格番号 (注)	内容
品質管理部門	ISO/IEC 25000	システム及びソフトウェアの品質要求定義と評価に関する基本概念と用語を定義
	ISO/IEC 25001	SQuaREの考え方に沿って品質要求定義や品質評価を実施する組織が、それらに関する技術やノウハウを蓄積し、標準化し、プロジェクトでの適用を支援し、改善するサイクルを規定
品質モデル部門	ISO/IEC 25010	システム及びソフトウェアの品質の構造をモデル化し、それを構成する品質の観点(品質特性及び品質副特性)を、利用時の品質モデル及び製品品質モデルとして定義
	ISO/IEC 25011 (審議中)	システム及びソフトウェアを用いたサービスの品質の構造をモデル化し、それを構成する品質の観点(品質特性及び品質副特性)をサービス利用時の品質モデル及びサービス製品品質モデルとして定義
	ISO/IEC 25012	システム及びソフトウェアが処理するデータの品質の構造をモデル化し、それを構成する品質の観点(品質特性)をデータ品質モデルとして定義
品質測定部門	ISO/IEC 25020	品質測定の基本概念と作業要件を規定
	ISO/IEC 25021	品質測定量の算出によく用いられる要素(例:規模、欠陥数など)を品質測定量要素(QME: Quality Measure Elements)として定義
	ISO/IEC 25022 (審議中)	ISO/IEC 25010の利用時の品質モデルの品質特性/品質副特性を定量化するのに用いる品質測定量を定義
	ISO/IEC 25023 (審議中)	ISO/IEC 25010の製品品質モデルの品質特性/品質副特性を定量化するのに用いる品質測定量を定義
	ISO/IEC 25024 (審議中)	ISO/IEC 25012のデータ品質モデルの品質特性を定量化するのに用いる品質測定量を定義
品質要求部門	ISO/IEC 25030	品質要求定義の基本概念と作業要件を規定
品質評価部門	ISO/IEC 25040	品質評価の基本概念と作業要件を規定
	ISO/IEC 25041	開発者、取得者、独立評価者のそれぞれの立場に応じた品質評価の進め方を規定
	ISO/IEC 25045	回復性(Recoverability)の実用的な品質測定量を規定
拡張部門	ISO/IEC 25051	品質保証部門や独立評価機関が実施する既製ソフトウェア製品(RUSP: Ready to Use Software Product)の第三者評価を規定
	ISO/IEC 25060～66 (一部審議中)	ユーザビリティの要求定義と評価に用いる様式を規定

(注) カッコ内は 2015 年 3 月時点

## 2.3 SQuaRE の品質のとらえ方

次に、SQuaRE を理解するうえで、重要な考え方である「品質ライフサイクル」と「品質モデル」について説明する。

システム及びソフトウェアの品質要求定義と評価を行うには、対象製品が品質ライフサイクルのどの段階にあるかを考える必要がある(図 2.3-1)。図 2.3-1 に示された大きく循環する矢印は、ステークホルダのニーズに基づいて、利用者視点、開発者視点の品質要求を仕様化し、開発・実現された成果物の品質をライフサイクルの各局面(設計・製造、テスト、運用・保守)で評価して利用者・市場に製品として提供し、その後、ステークホルダの新たなニーズが製品品質の強化・改善にフィードバックされることを表している。

利用者は、例えば「業務効率を 10%アップする」といった利用時の品

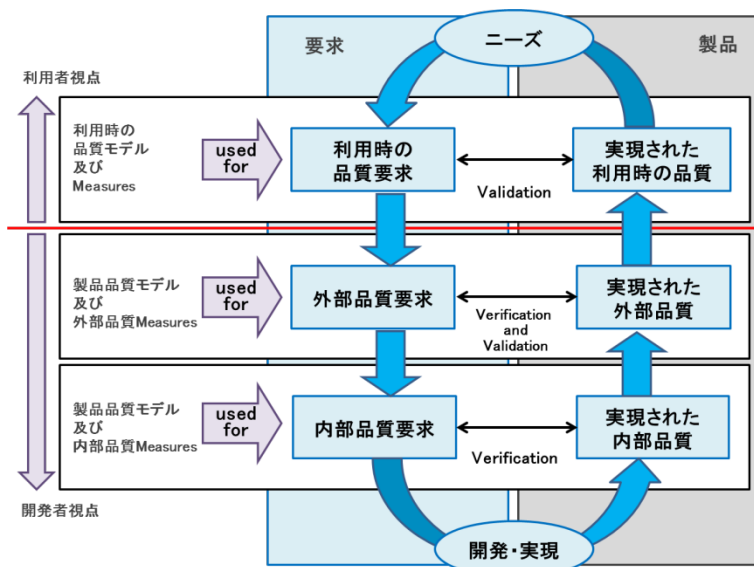


図 2.3-1 システム&ソフトウェア品質ライフサイクル

質の一つである「効率性」に関する要求(品質目標)を出す。それを実現するには、システムソリューションとして機能要求も出す必要があるが、非機能要求である品質要求(外部品質要求)も提出される。性能効率性、使用性、信頼性などの品質特性がそれに貢献することが予想される。さらに外部品質要求を実装レベルで支えるのが内部品質要求である。プログラムの構築やテストの仕方についての品質が問われることになる。

ここで、システム及びソフトウェアの品質は「製品品質」と「利用時の品質」に分類されている(図では製品品質をさらに外部品質と内部品質に分けている。)。機能の豊富さ、操作のしやすさなど、製品自体が備えている特徴から見た品質は「製品品質」と呼ばれ、作業効率の向上、使った時の満足感など、その製品の利用者に対する影響から見た品質は「利用時の品質」と呼ばれる。

SQuaRE では、「製品品質」「利用時の品質」、それぞれに対応する品質モデルを定義している。それ以外にも、システム及びソフトウェアの運用において重要となる「データ品質」に対応する「データ品質モデル」が定義されているが、これについては「3.3 データ品質」で説明する。

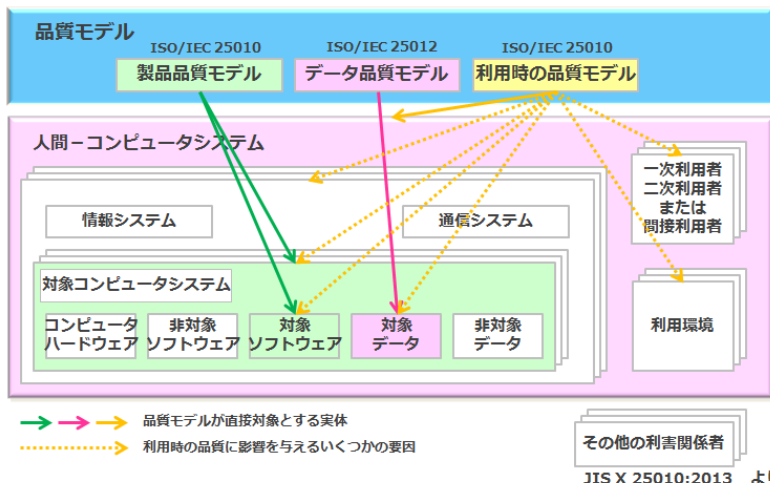


図 2.3-2 品質モデル

図 2.3-2 は、これら 3 種類の品質モデルが対象とする領域を示したものである。また、「製品品質モデル」「利用時の品質モデル」の概略はそれぞれ図 2.3-3、図 2.3-4 のとおりである。

「品質モデル」は、品質要求を定義し評価するための枠組みであり、品質を分類・整理する時の着眼点(観点)を与えるものである。品質モデルでは、品質の概念を木構造に展開する。品質を構成する上位の要素(例えば、「満足性」や「機能適合性」)を品質特性と呼ぶ。この品質特性をさ

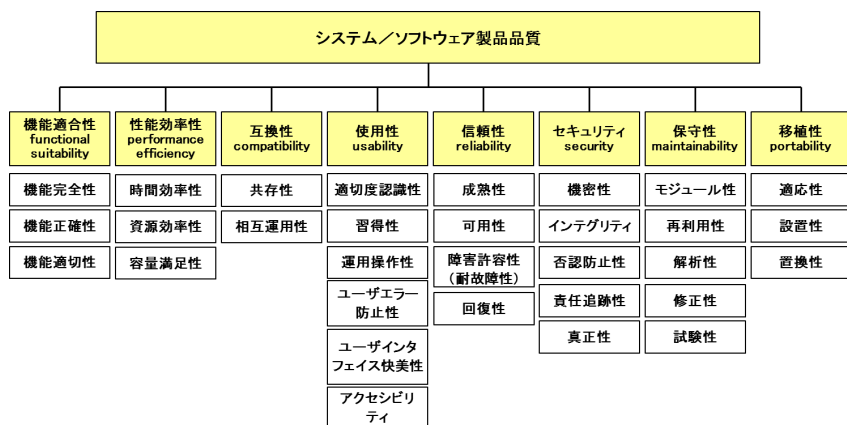


図 2.3-3 システム・ソフトウェア製品の品質モデル

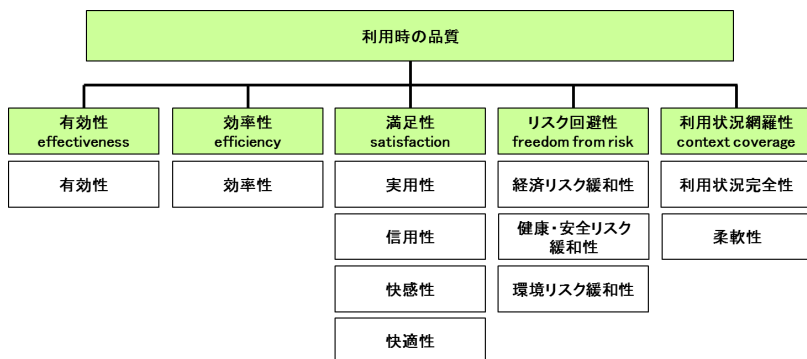


図 2.3-4 利用時の品質モデル

らに詳細化した下位の要素(例えば、「実用性」や「機能完全性」)を品質副特性と呼ぶ。

図 2.3-1 の品質ライフサイクルにおいて、2つの品質モデルは関係があり、一方の品質(副)特性の評価結果が他方の品質(副)特性の評価結果に影響を与えている。「製品品質」は、レビューやテストなどを通して評価し、製品の品質向上に役立てることに加え、利用者から見た「利用時の品質」の予測に用いることができる。また、「利用時の品質」は利用者へのアンケートや利用場面の観察などを通して評価され、「製品品質」の要求定義の入力として用いることができる。

## 2.4 SQuaRE の活用

SQuaRE で規定された品質モデル(品質特性や品質副特性及びそれらの関係)と品質測定量(品質特性や品質副特性の良否判定に用いるメジャー及びその測定方法)及びそれらの活用方法を理解するには、規格文書を読むことのほかに、本書の“SQuaRE 品質モデル活利用リファレンス編”<sup>4</sup>を参考にするのがよい。

規格で規定された事項をすべて正確に実践するのはなかなか難しいが、後述する認証取得のような場合を除けば、規格に完全に準拠する必要は必ずしもなく、まずは、規格の有用な部分を参考にしながら、品質要求定義や評価のプロセスの改善に取り組むのがよい。SQuaRE でも、品質モデルの適用にあたって、品質(副)特性の追加、修正、詳細化など、変更理由を付記してモデルをテーラリングすることを認めている。

SQuaRE のように体系的に整理された基準を用いることで、品質要求定義の際に考慮すべき観点の漏れを防ぐことに役立つ。また、「発注者と受注者」、「設計者とプログラマ」、「システム開発者とソフトウェア開発者」、「A 分野の事業者と B 分野の事業者」など、つながるシステムにおいて、異なる役割を持つステークホルダのコミュニケーションを潤滑にし、品質に対する共通の認識を与え、それぞれの立場における品質向上の取り組みを促し、最終的に完成した製品品質の安定につながると期待できる。

典型的な利用の主体は、第一者(開発者)、第二者(取得者)、第三者(独立評価者)に分類できる。

開発者は、システム及びソフトウェアの開発を進めていく中で、品質要求定義や評価に SQuaRE を用いる。要求定義の段階では、SQuaRE の品質モデルに照らして品質要求を仕様化し、品質測定量を用いて評価の基準を設定する。その後、開発の主要マイルストーンで実績値を算出し、

<sup>4</sup> 本資料(ダイジェスト版)には掲載していません。

基準に照らして工程終了や出荷可否の判定を行う。それによって、品質要求を多角的かつ定量的に定義し、要求された品質の実装度合いを客観的に評価し、必要な修正を施して、品質を作り込んでいくことが可能となる。

取得者は、取得する製品が要求に見合うか否かを判断するのに、又は候補となる製品のうち、どれが最も要求に合致するかを選定するのに用いる。自身の品質要求を品質モデルに照らして定義し、品質測定量を用いて基準を設定し、レビューやテストの結果に基づいて実績値を算出し、基準に照らして判定を行う。それによって、期待される品質の水準に満たない製品のスクリーニングや用途に最も合致した製品の選択をシステムティック行うことが可能となる。

独立評価者は、開発者や利用者から独立した立場で、客観的に製品を評価するのに用いる。例えば、品質保証部門が開発者とは異なる観点からテストを実施して製品の品質を保証したり、認証機関が国際規格の要求事項に準じて製品品質を認証したりするのに **SQuaRE** が活用できる。



# 第3章

## いま注目したい品質の視点

この章では、SQuaREで定義された品質の中で、特に注目しておきたい品質を取り上げ詳しく解説する。

- 
- 3.1 ユーザビリティと利用時の品質
  - 3.2 セキュリティ／セーフティ
  - 3.3 データ品質
-

## 3.1 ユーザビリティと利用時の品質

### ■ユーザビリティの定義と考え方

ソフトウェアやシステムは、どんなに完成度が高くても意図した利用者にとって役に立たなければ意味がなく、また操作がわかりづらい、時間がかかってしまうなどユーザビリティに問題があると使ってもらえなくなる。このようにソフトウェアやシステムの企画・開発においてユーザビリティが重要であることは明らかなが、ではどのようにしてユーザビリティを向上できるか、悩むことが多いのではないかと思う。ここではそのユーザビリティの定義を解説し、利用時の品質の考え方を紹介する。なお、ユーザビリティと利用時の品質の関係はのちほど説明する。

ユーザビリティは曖昧なものにとらえられることが多いが、人間工学の国際規格 ISO 9241 シリーズにて次のように定義され、測定できるものとされている。「ある製品又はあるシステムが、指定されたユーザーによって、指定された利用の状況下で、指定された目的を達成するために用いられる際の有効さ、効率及びユーザーの満足度の度合い」(ISO 9241-210:2010)

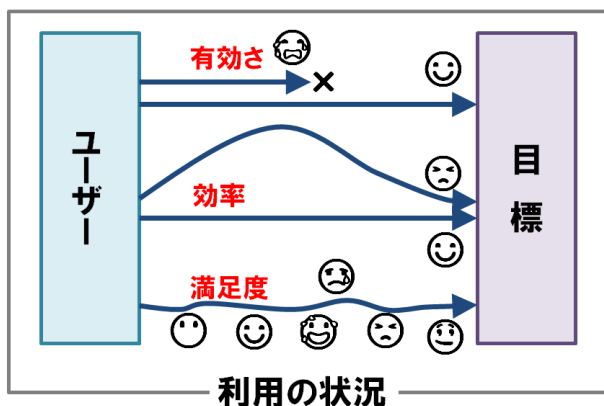


図 3.1-1 ユーザビリティの定義より(ISO 9241-11、9241-210)

これはユーザビリティを考える場合、ユーザー、ユーザーの目標、利用の状況の3つを決める必要があること、そして3つの観点(有効さ、効率、満足度)で測定できることを示している。図3.1-1にこの定義の模式図を示す。言い換えると、この3つが決まらなるとユーザビリティは測れない、ということであり、またこの3つのうち一つでも変化するとユーザビリティの良し悪しに変化する可能性があるということである。

例えば、身近な例としてある携帯電話とスマートフォンのユーザビリティを比較するとしよう。ユーザーは80歳男性、目標は電話番号を打って孫に電話をかける、場所は自宅の居間とする。携帯電話(フィーチャーフォン)は数字キーと通話キーを押せばよく、30秒で話ができた。一方スマートフォンは電話というアプリを探すのに時間がかかり3分かかってなんとか電話ができた。どちらも電話をかけることはできたので、有効さにはほぼ差がない。しかし効率は30秒でかけることができた携帯電話の方が3分かかったスマートフォンより明らかによいといえる。それも6分の1の時間という数値で比較することができた。満足度はいろいろな測定法があるが、例えば7段階の評定尺度(使いやすさの主観評価シート)への記入によって測ってみると携帯電話は6点、スマートフォンは2点と明らかな差があった。以上のようにユーザーとその目標、利用の状況を定めることで、2台の機器のユーザビリティの良し悪しを3つの観点で明確に比較できたことがわかるだろう。ここで目標を“Webサイトで情報を調べる”に変えた場合、ユーザーを20歳の大学生女子に変えた場合、異なった結果になることは予想できると思う。

また利用の状況に関しては、例えば自宅の居間ではなく、“街灯のない夜道”とした場合、携帯電話はキーが光らないタイプだったため、暗がりでは番号が見えず、電話をかけることができなかった。一方、ディスプレイに数字キーが表示されるスマートフォンでは多少時間がかかっても電話をかけることができた。携帯電話よりも“有効さ”において明確な差があり、自宅の居間とは逆の結果となった。さらに利用の状況は、電波状況やま

わりの騒音の大きさなど電話をかける上で影響を受ける様々な要因が存在する(図 3.1-2)。

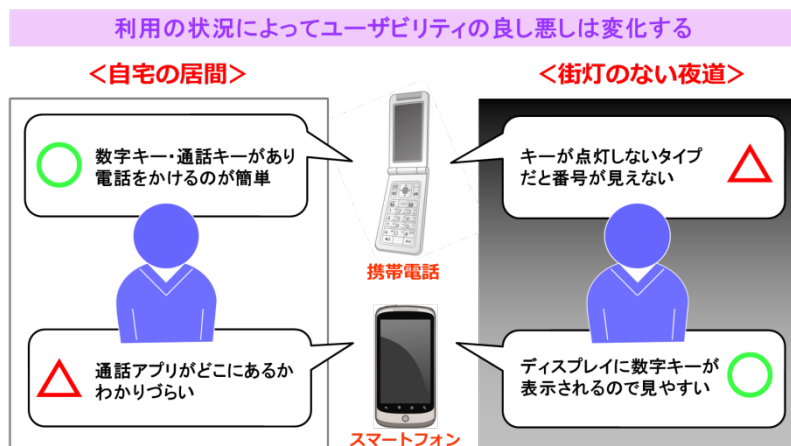


図 3.1-2 ユーザビリティの良し悪しは変化する

ユーザビリティが、曖昧ととらえられるのは、この 3 つをきちんと定義せずに議論することが原因である。特に利用の状況は刻々と変化するため必要な項目をきちんと定義しておく必要がある。

SQuaRE 規格でも同じ考え方が採用されており、ソフトウェア開発においてさらに有効となるように拡張されている。SQuaRE では利用時の品質を次のように定義している。

「特定の利用状況において、特定の目標を達成するために、特定の利用者が彼らのニーズを満たすために、有効性、効率性、リスク回避性及び満足性を満足して製品又はシステムを使用することができる度合い」(JIS X 25010:2011)。

ユーザビリティの定義における有効さ、効率、満足度の度合いに加えて、リスク回避性があげられており、より広い概念となっている。SQuaRE では、本書の第 2 章で紹介があったように、利用時の品質モデル(図 2.3-4)があり、上記 4 つに利用状況網羅性を加えた 5 つを利用時の品質モデル

として定義している。SQuaRE ではこの5つの特性の下に副特性を定義しており、それらを用いて測定する枠組みを与えている。利用時の品質の測定は様々な方法で測定されるが、まずここに含まれている特性、副特性さらにそれらを構成する特徴をベースに考えていくとよい。

## ■ユーザビリティ向上のための人間中心設計(HCD)

ユーザビリティや利用時の品質は、ユーザーや利用の状況によって変化するためこれらを考慮した具体的な設計(画面遷移や画面設計など)で苦勞することが多い。対象ユーザーの特徴が異なると最適な画面遷移やデザインが異なるからである。一方で使いやすいシステムを設計するためのプロセスや手法はどのようなプロジェクトでも概ね同じである。その枠組みを与えるものとして人間中心設計(HCD<sup>1</sup>)がある。ユーザー中心設計(UCD<sup>2</sup>)とも呼ばれている。1999年にHCDに関する最初のISOが発行され、2010年には最近の動向にあわせた改訂版ISO 9241-210が発行された。2010年の改訂では、HCDの目標としてユーザーエクスペリエンス(UX<sup>3</sup>)向上が追加された。UXはユーザビリティも含む概念であり、有効さ、効率、満足度はもちろん、よりよい体験(嬉しい、感動)なども包含するものとして使われている。

図3.1-3にHCDのプロセスを示す。ユーザビリティや利用時の品質を考える際、重要となる利用の状況や対象ユーザーのことを開発の上流で調査によってきちんと把握すること、デザインコンセプトや要求事項を明確にすること、そして設計(プロトタイプング)と評価を繰り返すことがエッセンスとなっている。HCDを推進するためには、プロセス、手法、スキルの3つが柱になっており、図3.1-3のプロセスをベースに、適切な時期に適切な手法を用いること、そしてそれらをきちんと実行できるスキルをもつ人材(専門家)が参画することが重要となる。専門家の例としてはHCD-Net認

<sup>1</sup> HCD: Human Centered Design

<sup>2</sup> UCD: User Centered Design

<sup>3</sup> UX: User eXperience

定の人間中心設計専門家や日本人間工学会認定の人間工学専門家などがある。

SQuaRE の拡張部門には、この HCD プロセスを推進するための規格も含まれている。第 2 章の表 2.2-1 に示すように ISO/IEC 25060 番台として審議されている。これまでに 25060 と 25062 が発行されており、現在、利用状況の記述やユーザー要求報告書、ユーザー評価報告書などに関する規格が策定中で、発行されると HCD プロセス推進のよい手引き書となるであろう。

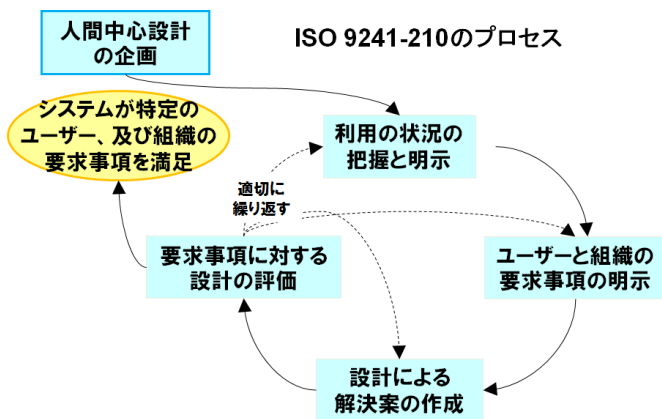


図 3.1-3 ISO 9241-210 のプロセス

(注) SQuaRE (ISO/IEC 25010 等)では、“usability”という用語が、製品品質の 8 つの特性のひとつとして用いられており、SQuaRE の JIS (JIS X 25010 等)においてその日本語訳は“使用性”を用いている。一方、人間工学分野の ISO 9241 シリーズでは、現在“usability”の訳は、“ユーザビリティ”を用いており、SQuaRE の“利用時の品質”の概念に近い(例外として人間工学分野の JIS Z 8521:1999 だけは“usability”を“使用性”と訳している)。

## 3.2 セキュリティ／セーフティ

日々の報道等にも表れているように、IT システムのセキュリティやセーフティ(安全性)に関わる対策が不十分であることに起因する障害や事故が頻繁に発生している。例えば、海外の原子力プラントの制御システムにマルウェアが感染し、システムがダウンした。また、インターネットに接続された家電機器が攻撃を受け、プライバシー情報が外部に流出する状態になっていた、等の事故が報じられ、人々に衝撃を与えた。セーフティに関しても、放射線治療装置のソフトウェアエラーによる治療事故等の事例が数多く報じられている。今や、日常生活で使われる機器、社会生活や産業で使われる機器、ネットワーク上の各種サービスといった機器・サービスの多く<sup>4</sup>が IT で支えられており、今後も、その誤動作や、外部からの攻撃、情報漏えいなどが大きなインパクトを人々にもたらす可能性がある。ソフトウェア、システム、サービスが様々に組み合わせられ利用されることや同じプラットフォームで複数のソフトウェアが動くこと、またネットにつながってなくてもオフライン入力データに巧妙に攻撃が仕込まれる可能性

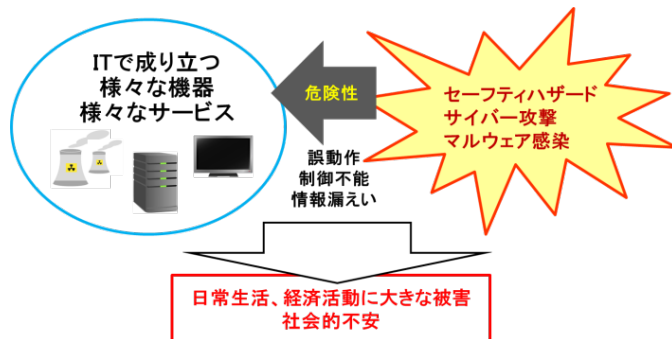


図 3.2-1 セーフティとセキュリティの重要性

<sup>4</sup> 具体的には、モバイル機器、家電、自動車、医療機器、交通機関の制御、プラントの制御、ネットバンキング、行政サービスなどのセキュリティ、セーフティ課題が話題として取り上げられている。

などを考慮すると、一見、クリティカルでないシステムでも、必要なレベルでセキュリティやセーフティの品質に留意することが必須であるといえる。しかし、現状を見ると、開発サイドに、その深刻さの認識や対策が十分であるとは言い難い。

セキュリティ／セーフティの課題は差し迫った課題であるため、それらの課題だけに集中して対応策や要求事項を考える傾向もあるが、実際にはこれらの課題はシステム全体の品質を確保する中で考慮すべき課題である。また逆に、特定のセキュリティ／セーフティ上の機能の実現により、課題を果たさなければならないことも多くある。こうしたことを考慮すると、あるシステムが全体としてセキュリティ／セーフティの課題に適切に対応できるためには、システムのセキュリティ／セーフティに関わるアーキテクチャが明確である必要がある。アーキテクチャの大きな観点として、通常は、システムの「本来機能」(利便性などの実現)を実現するためのソフトウェア／システム部分と、専門的にセキュリティやセーフティを担保するための「特定機能」部分を区分することが多い(図 3.2-2 を参照)。システムの「本来機能」の側面とは、その機能がシステムの本来行いたいことを目標通りに実現することであり、セキュリティ／セーフティ品質としては、意図せず埋め込まれるセキュリティの穴などが問題となる。もう一つの側面である、

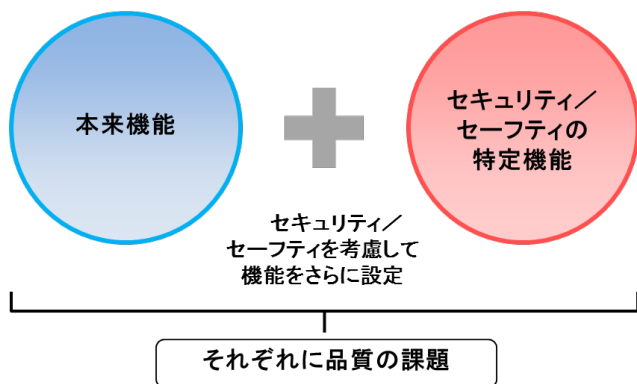


図 3.2-2 本来機能と特定機能



セキュリティ／セーフティを担保するための「特定機能」部分としては、本来機能だけでは実現できないセーフティ機能(例えば安全制御装置のソフトウェア機能)や、セキュリティ面から必要とされるセキュリティ固有機能(例えば暗号化機能)の適切な実現が第一に問題である。そして、付随的にそれらの機能自体での実現上の欠陥の回避が課題となる。そのほかに、アーキテクチャ上の観点として、セキュリティ／セーフティを管理する厳密さのレベルを区別した複数の領域を設定し、それら領域間での情報の移動を制限するといった考え方がある。

### ■セーフティ課題

セーフティは、そのシステムを利用して生活や業務を行う際に、それらの実施経過や結果が、安全であり、安心できるものであることを要請している。

セーフティは、組込み系システムや制御システムなどの物理的な機器の動きに関わるシステムが利用者の誤操作を防止する機能を備えていることや、鉄道や航空機の管制システムがあらゆる状況においても事故を回避する設計がされていること、などが含まれる。従来から安全性が重要なテーマとなってきた交通等の分野では、過去のトラブルや事故の分析から得られたノウハウの蓄積がある。一方で、ソフトウェアによる制御の比重が高まったことや、「つながる」システムの環境が増えてきたことから、いままであまりセーフティに関与しないと思われてきた分野でもセーフティを問題としなければならない状況が生まれている。例えば、従来は単に人に対する情報提供サービスとして扱われていたものが、「つながって」自動化され、装置等が自動的に稼働するようになると、セーフティにかかわるシステムとなる。地図情報システムや災害情報システム(それらの機能内容、通知タイミングや誤報防止などの性能、及びデータ内容)などがそうした例に挙げられる。

ところで、安全性を深く掘り下げた国際的な一般基準としては、機能安

全という考え方がある<sup>5</sup>。機能安全の考え方では、安全の実現のために、開発プロセスの各場面において、順守すべき事柄が必要事項として列挙され、深く追求すべきことが述べられている。しかし、設計上の考慮漏れや、機能を実現するソフトウェアの全般的な品質確保が不十分であれば、列挙された事項に取り組むといっても、それは絵に描いた餅になる可能性がある。思いがけない品質の落とし穴を生み出さないように、品質確保の幅広い視点をもつことと、セーフティの基本的な課題理解とが着実に実行されることの両方が必要である<sup>6</sup>。

機能安全の規格では、どのような水準でセーフティを組込むのかについて、システムの目的やリスクを想定し、求められるレベルに応じた対策を行うという考え方として、安全性インテグリティレベル(SIL)を用いることが

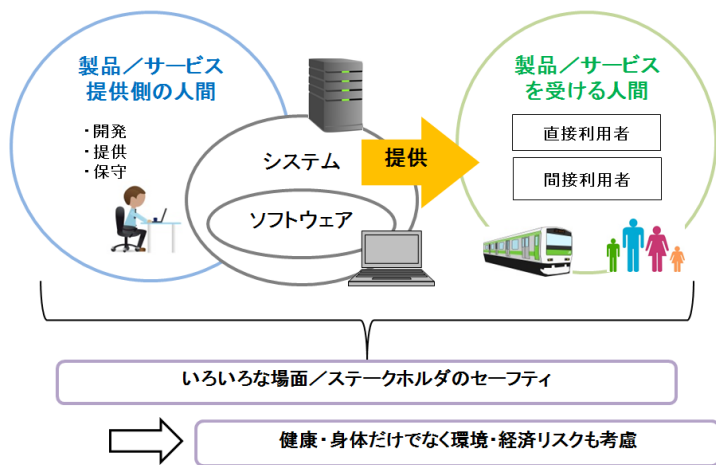


図 3.2-3 システムのセーフティとさまざまなステークホルダ

<sup>5</sup> IEC 61508(機能安全)やその分野別規格である ISO 26262(自動車分野)などがある。これらの規格は、装置やシステムが果たすべき安全性について、システム全体の構造などで実現する安全と、保護装置や安全機能装置などで実現する安全との区別を行ったうえで、後者(機能安全)についての基準を定めている。

<sup>6</sup> 多くのプログラムが並行して動作する環境では、例えばあるプログラムの動作が他のプログラムの動作を妨害するといった事象はしばしば出会うことであるが、そうした共存性の話題もセキュリティ上の可用性などの課題につながることもある。

多い。安全が重要だといっても、過剰な安全設計は開発期間やコストの増加に直結するため、品質要求の設定では、求められる必要十分な機能を明確にすることが重要である。

セーフティ課題を考える上で見逃してはならない点の一つに、システムに関わる多くの利用者(あるいはステークホルダ)がいるという点がある。システムを直接操作するいわゆる利用者のほかにも、システムを継続的に利用できる状態にする役割の保守要員も関与する。また、交通系のシステムの正常な運用が妨げられたときに、交通機関の利用者に大きな影響が及ぶことを考慮すると、システムの間接的な利用者も想定しなければならない。セーフティにかかわる要件定義やテストでは、そうした広い利用者を考慮することが重要である(図 3.2-3 参照)。

図 3.2-3 に示されているように、安全に関わるリスクには、健康・身体にかかわるもののほかに、より広く、社会やビジネスにとってのリスクとして、「環境リスク」や「経済リスク」の視点もある<sup>7</sup>。環境リスクとは、システムの不具合・不十分のために関連する生活環境・労働環境が汚染されるリスクが、適切に緩和・回避されているかを問題としている。また経済リスクは、同様に、システムの不具合・不十分のために関連する企業や個人がビジネス上／金銭上のデメリットを被るリスクが、適切に緩和・回避されているかどうかを問題としている。これらも、多様な利用者の観点から、システムの利用の最終的結果としてリスクを免れられている程度を品質の一環として問題としている。

---

<sup>7</sup> SQuaRE では、広く安全にかかわるリスクに対応する品質特性を「リスク回避性」といい、その詳細化として「経済リスク緩和性」、「健康・安全リスク緩和性」、「環境リスク緩和性」を取り上げている。セーフティはソフトウェア単独で実現するものではなく、システム総体として実現するものという考え方で、利用時の品質モデル内の「リスク回避性」という品質特性を設けている。一方、セキュリティは、ソフトウェアの品質が直接関与するとの立場で製品品質に含めるとともに、データ品質の面でも機密性などの特性を課題としている。

## ■セキュリティ課題

「つながる」システムでは、その複雑なシステム構成を突いて、またシステムの複雑さに隠れた開発者の単純なミスも突いて、セキュリティ上の攻撃が行われる。セーフティと異なるのは、セキュリティの攻撃者は目的をもって執拗に攻撃をかけること、また一度成功した攻撃手順は攻撃者の知識として蓄積され、攻撃は次第に高度化することである。セキュリティ攻撃の目的にはいたずらのものから政治的なものなど多岐にわたるが、特に、金銭的な利益を得ようとするものについては、攻撃者の数と被害額を大きくする要因となっている。

ところで、セキュリティ評価の国際標準であり政府調達等にも使われるコモンクライテリア規格<sup>8</sup>のセキュリティ機能要件の中にも、SQuaRE<sup>9</sup>でいう品質特性(副特性)にあたる、機密性、インテグリティ、真正性といった項目の対応項目が含まれる(図 3.2-4 参照)。コモンクライテリアは、IT 製品を取り扱う際に、そのセキュリティ品質の確立された総合評価方式として国際的に認知された方式である<sup>9</sup>。それほど厳密さを要せず、コモンクライテリア認証の対象としないソフトウェア/システムの開発でも、コモンクライテリアでのセキュリティ機能要件やその保証要件の考え方は、セキュリティ対策を漏れなく進めるために参考となる。その際、SQuaRE を活用し、



図 3.2-4 コモンクライテリアに含まれるセキュリティ機能のいろいろな側面

<sup>8</sup> セキュリティを社会的に問題とする際に、組織のセキュリティを対象とするセキュリティマネジメントシステム (ISMS、ISO/IEC 27001) 的な取り扱いと、製品のセキュリティを対象とするセキュリティ評価基準 (コモンクライテリア、ISO/IEC 15408) 的な取り扱いとがある。ここでは、後者の側面について、SQuaRE のテーマとの関係を見る。

<sup>9</sup> コモンクライテリアは、米国や日本での IT 製品の政府調達基準の一部として採用されている。

幅広の品質特性群を統一的な視点で点検してみることも有益といえる。

セキュリティは多様な側面を持っており、例えば図 3.2-4 で示すような課題がシステムの品質として評価される。このうちで、例えば責任追跡性とは、セキュリティ上の事象(例えば、あるファイルが閲覧されたこと)について、それを行った主体が具体的に A さんだと特定できることをシステムの品質特性として表している<sup>10</sup>。この場合、あるシステムのセキュリティがしっかりしているとは、そうした追跡が確実に行われることを意味している。従来の一般的なシステムでは、追跡能力が低いことが多く見られたが、「つながるシステム」では、多少とも追跡性の高度化を考慮していくことが求められる。また、図 3.2-4 で示した真正性やインテグリティ<sup>11</sup>に関して、現在では、通常のアプリケーション開発でも、OS が持つセキュリティの仕組み(ユーザアカウント制御や、サンドボックスといった仕組み)や、データベースのセキュリティの仕組みを十分に使いこなすことが必須となっている。最近では、開発するシステムに応じて、どのようなセキュリティ対策を考慮及び実装すべきか、ある程度整理された情報も公開されており、開発者はこれらの情報を参考に、品質要求を明確化することが必要である<sup>12</sup>。

## ■リスクの取り扱い

セキュリティの課題もセーフティの課題も、セキュリティ/セーフティ上の事件・事故が発生しないようにする、つまり、リスクに対する防衛策であるという面を持っている。したがって両者とも、どのようなリスクを想定し、そ

<sup>10</sup> 事象の主体が特定できるようにすることは機能要件とみられることもできるが、従来はこのような要件は機能とはみなされないケースがあったことを背景に、SQuaRE では品質特性として扱っている。

<sup>11</sup> 真正性とは、コンピュータ処理上の主体(人やプログラムなど)が、ただしく本物であることの確からしさのことで、認証性ということもある。インテグリティとは、情報やデータが不正に書き換えられない程度のことで、セキュリティの用語としては完全性ともいう。

<sup>12</sup> また、特に Web サイトの開発では、OWASP TOP10 といったデファクト基準 (<http://www.owasp.org/>) や、IPA で提供している「安全なウェブサイトの作り方」 (<https://www.ipa.go.jp/security/>) 等を参考にすることが有効である。

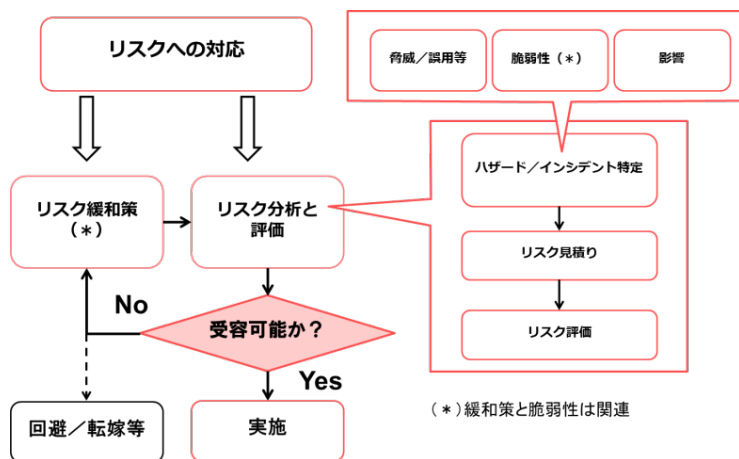


図 3.2-5 セキュリティ/セーフティのリスク評価とリスク緩和の反復プロセス

れにどのように対処するのかというストーリーが明確にならなければならない(図 3.2-5 を参照)。

セキュリティやセーフティの品質要件を定める際には、程度の差はあっても、こうしたリスク分析に基づいた要件策定が基本的な形となると思われる。コストの考慮といった面も含めリスク分析・評価の結果としてリスクが明示的に受容可能となるまでリスク緩和策の実装・実現を追求すべきだということに注意が必要である。

また、セキュリティ/セーフティを追求する上での、これとは異なる観点として、セキュリティ/セーフティを、「困ったことを緩和するためにやむをえず考慮しなければならない追加機能」として考えるのではなく、適切なセキュリティ/セーフティ方策を採用することによってそれまでの発想では実現困難であった価値を実現するといった積極的な品質要件とする立場も考えられる<sup>13</sup>。

<sup>13</sup> 例えば、電子署名の導入により、従来危険として試みられなかった情報保存や通信経路の設定等が可能になりシステムのイノベーションが起こることがある。

### 3.3 データ品質

システム及びソフトウェアの品質を考えると、その処理対象となるデータの品質を併せて考えることが重要になっている。システムやソフトウェアの品質に問題がなくても、データの品質に問題があれば、期待する結果が得られず、利用者からは、その製品やサービスの品質が不足しているように見えることに変わりない。

年金、預金など個人の財産に関するデータ、行政サービスや課税の基礎となる国民、住民の基本データ、事業上の主要な意思決定に用いる市場動向等に関するデータなど、品質を考慮すべきデータには多種多様なものが考えられる。本書が注目する「つながる」システムにおいても、機器やシステムの間で都度やり取りされる情報や、膨大に蓄積されたデータそ



図 3.3-1 ビッグデータの例

<http://www.soumu.go.jp/johotsusintokei/whitepaper/h24.html>

のものが新たな価値を生み出すといわれるビッグデータなど、データの品質がシステム全体の品質に与える影響は、今後も増大すると思われる。

システム及びソフトウェアと同様に、データに関しても品質の多様性に対する考慮が必要である。データに欠落がないこと、データ相互に矛盾がないこと、プライバシーに対する配慮がなされていることなど、データ品質にも様々な特性が求められる。このようなニーズに対応して、SQuaREでは、ISO/IEC 25012 で表 3.3-1 に示すデータ品質モデルを規定している。

表 3.3-1 データ品質モデル

特性	データ品質		
	固有	システム依存	
正確性 (Accuracy)	○		固有の視点からのデータ品質特性
完全性 (Completeness)	○		
一貫性 (Consistency)	○		
信ぴょう (憑) 性 (Credibility)	○		
最新性 (Currentness)	○		
アクセシビリティ (Accessibility)	○	○	固有の視点及びシステム依存の視点からのデータ品質特性
標準適合性 (Compliance)	○	○	
機密性 (Confidentiality)	○	○	
効率性 (Efficiency)	○	○	
精度 (Precision)	○	○	
追跡可能性 (Traceability)	○	○	
理解性 (Understandability)	○	○	システム依存の視点からのデータ品質特性
可用性 (Availability)		○	
移植性 (Portability)		○	
回復性 (Recoverability)		○	

SQuaRE のデータ品質モデルの特徴は、データ品質特性を固有の視点からのものとシステム依存の視点からのものに分類していることである。例えば、データの正確性や一貫性は、コンピュータシステムを介さず、紙に打ち出された状態でも評価できるが、バックアップなどを前提とした回復性は、当該データを処理、保管するコンピュータシステムを介さなくては評価することができない。加えて、2つの性質を併せ持つ、固有の視点及びシステム依存の視点からのデータ品質特性もある。例えば、データ



の機密性は、暗号化などによって機密保護されているデータを仕様から特定することができるとともに、機密保護対象データを保管しているシステムに仮想的な攻撃を試みてデータの機密性を評価することもできる。



---

「つながる世界のソフトウェア品質ガイド」  
あたらしい価値提供のための品質モデル活用のすすめ  
ダイジェスト版  
編著者(敬称略)

独立行政法人 情報処理推進機構 ソフトウェア高信頼化センター  
ソフトウェア高信頼化推進委員会  
ソフトウェア品質説明力向上・普及 WG

主査	白坂 成功	慶應義塾大学大学院
委員	込山 俊博	日本電気株式会社
	佐藤 敏明	株式会社リコー
	柴田 浩	株式会社デンソー
	中山 貴史	パナソニック株式会社
	野中 栄一	株式会社ナーツ
	藤井 洋一	日本ナレッジ株式会社
	伏見 諭	合同会社ソフデラ
	星 光行	株式会社ジェイテック
	山崎 隆	オリンパスソフトウェアテクノロジー株式会社
	吉武 良治	芝浦工業大学
	鷲尾 一憲	日立アイ・エヌ・エス・ソフトウェア株式会社
オブザーバ	鱗原 晴彦	株式会社 U'eyes Design
編集者	宮崎 義昭	独立行政法人情報処理推進機構
	細目 紀子	独立行政法人情報処理推進機構

---





〒113-6591 東京都文京区本駒込 2-28-8  
文京グリーンコート センターオフィス  
電話 03-5978-7543 FAX 03-5978-7517  
<http://www.ipa.go.jp/sec/>