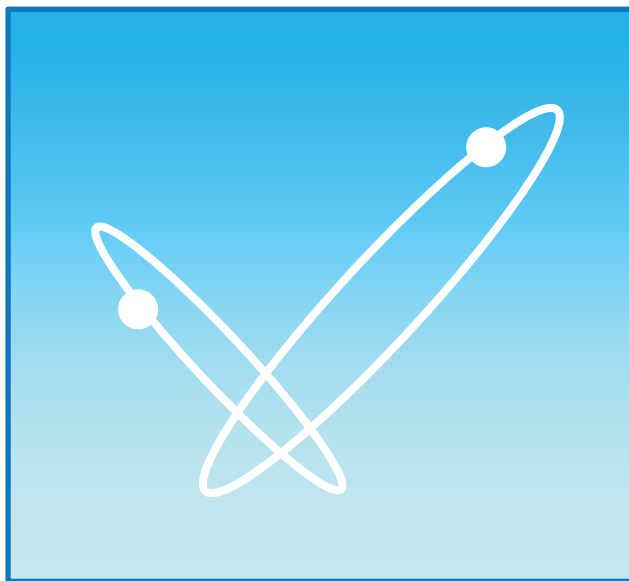


---

# 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

---



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

2015年5月

独立行政法人情報処理推進機構

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



## はじめに

---

いま企業が成長していくためには、新たな価値を提供する製品を生み出し、その価値を確実に利用者に届けることが必須となってきている。逆に、新たな価値を提供する製品を生み出せれば、例えば小さなベンチャー企業も強い影響力を発揮できる時代となってきた。しかしながら、新たな価値を生み出すことも、それを確実に利用者に届けることも決して容易なことではない。本書は、その両方の実現に欠かすことのできない品質の考え方について解説したものである。

現在、新しい価値を提供する製品の多くが、ネットワーク機能を組み込み、また、インタフェースを公開することで、単一の製品では提供できない価値をシステムとして実現している。システムの全てを1社で実現できなくても、複数社が協力することで競争力を発揮できる環境が生じている。

異なる製品同士がつながるために重要なことは、相互につながる製品それぞれがシステムとして使われた時の品質を確保し、その情報を外部に公開することである。さらに、その価値を確実に利用者に届けるためには、利用者を含むすべての関係者が価値を感じるあらゆる側面から品質が考慮されなければならない。そのためには、製品分野を越えた共通的な品質に関する考え方が重要となってくる。これが、つながる製品が強い競争力を持つための源泉となる。

本書では、つながる世界のソフトウェア品質について解説するとともに、つながる品質を整理するための指針として国際規格「SQuaRE シリーズ」についてのリファレンスを提供する。本書が、新たな「つながる」価値を組込んだ製品を届けたいと考えている企業にとって、品質への取組みを見直すきっかけになれば幸いである。

---

---

# ソフトウェア品質ガイド編

---

# ◆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 |
| 1.5 | スマート家電の品質 <b>事例</b> | ..... | P14 |

## 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？      | ..... | P22 |
| 2.2 | SQuaRE の構造と概要   | ..... | P25 |
| 2.3 | SQuaRE の品質のとらえ方 | ..... | P28 |
| 2.4 | SQuaRE の活用      | ..... | P32 |

## 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             | ..... | P36 |
| 3.2 | ユーザビリティを向上させる取組み <b>事例</b> | ..... | P41 |
| 3.3 | セキュリティ/セーフティ               | ..... | P49 |
| 3.4 | 自動車分野 <b>事例</b>            | ..... | P57 |
| 3.5 | 医療・ヘルスケア分野 <b>事例</b>       | ..... | P62 |
| 3.6 | データ品質                      | ..... | P67 |
| 3.7 | クラウドサービス分野 <b>事例</b>       | ..... | P70 |

## 第4章 品質向上に向けた改善のポイント

---

|     |                 |       |     |
|-----|-----------------|-------|-----|
| 4.1 | 開発・運用プロセスへの組み込み | ..... | P78 |
| 4.2 | 品質のトレーサビリティとは   | ..... | P87 |

## 第5章 利用者の安全・安心につながる品質説明

|                 |       |     |
|-----------------|-------|-----|
| 5.1 品質説明と第三者評価  | ..... | P92 |
| 5.2 PSQ 認証制度 事例 | ..... | P95 |

### << コラム >>

|                   |       |     |
|-------------------|-------|-----|
| IoT時代の品質課題        | ..... | P18 |
| 他標準との関係           | ..... | P27 |
| IT サービス品質         | ..... | P69 |
| アジャイル開発や小規模開発への応用 | ..... | P86 |
| 品質を戦略的に考える        | ..... | P89 |
| 海外の品質認証制度         | ..... | P98 |

# 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
- 1.5 スマート家電の品質 事例

## 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

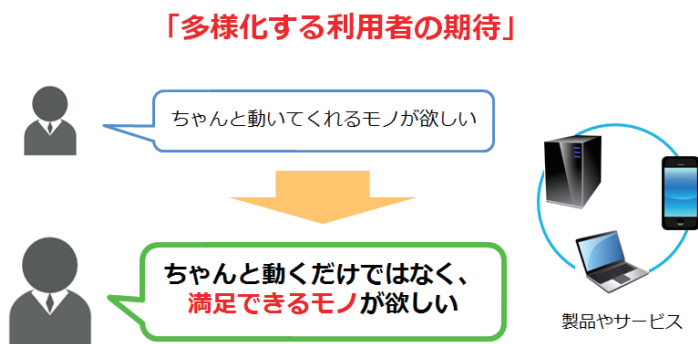


図 1.1-1 多様化する利用者の期待

<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<sup>3</sup>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」

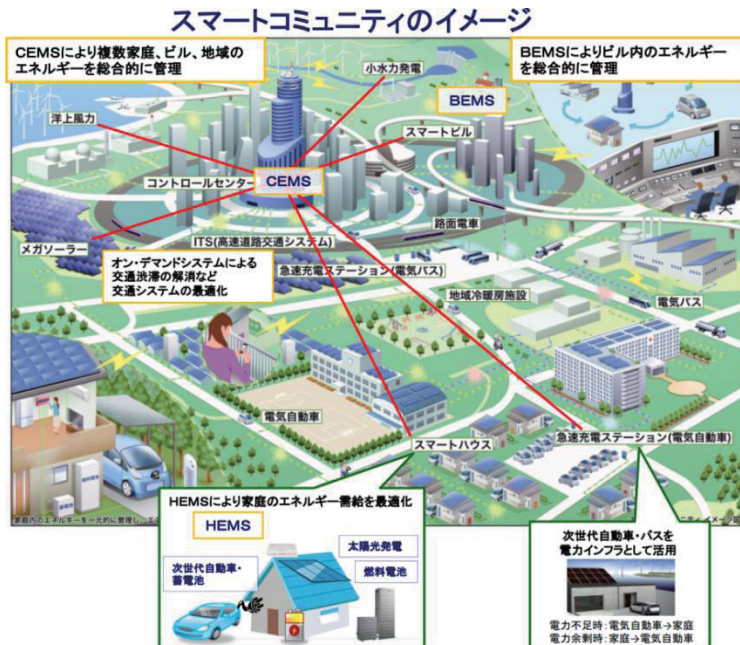


図 1.2-1 製品に対するステークホルダのさまざまな期待



## 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

用者は、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」

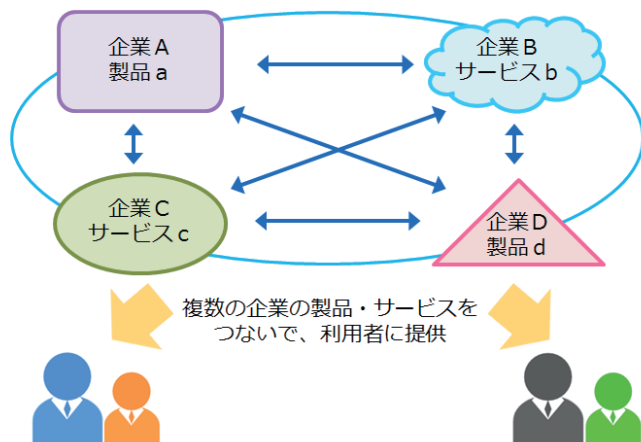


図 1.3-2 「つながる」システムのイメージ

## 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; 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。

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。

## 1.5 スマート家電の品質【事例】

スマート家電は、家電がインターネットなどにつながることで、私たちに新しい価値を提供してくれる。スマートフォンなどの情報端末を使って、家電を操作したり、状態を確認したり、設定を変更したりすることができる。最近の家庭内のインターネット環境やスマートフォンの急速な普及により、浸透しつつある分野である。

例えば、外出先からエアコンをオンすることで、家に着くころには、部屋を暖めたり、冷やしたりすることができる。また、いまの部屋の温度をスマートフォンから確認することができ、室温が高くなると、メールで知らせてくれる機能を使って、部屋に残したペットのために温度調整をすることもできる。

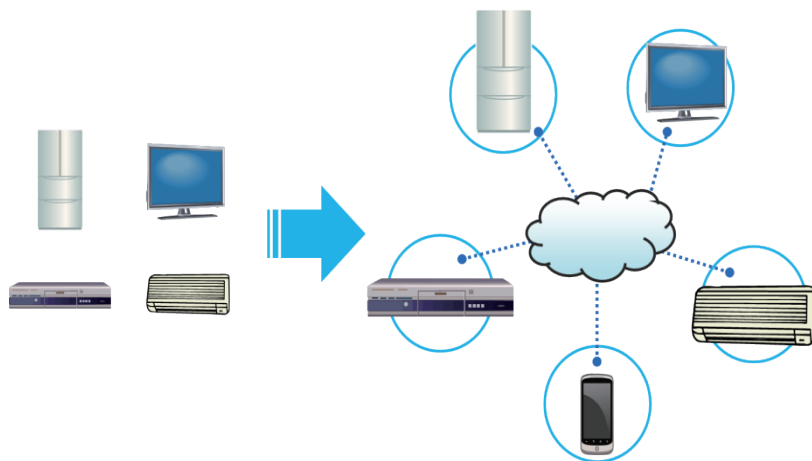


図 1.5-1 スマート家電のイメージ

冷蔵庫は、外出先から庫内の食材をカメラで確認することで、買い物先で、買い忘れや重複して買ったりするのを防ぐことができる。冷蔵庫の冷えが悪いと、メールで知らせてくれたりもする。

血圧計や体重計などは、スマートフォンで測定記録をグラフで確認することも可能となり、日々の健康管理が簡単にできるようになる。

## ■スマート家電の障害事例

家電がネットワークにつながることで私たちの暮らしが便利になる一方で、セキュリティへの対策は、非常に重要になってきている。これまでは、パソコンやサーバーなどがサイバー攻撃の対象であったが、家電も同じような脅威にさらされてきている。現在のところ、家電が重要な機密情報を持つケースは限られるため、情報漏えいが問題になることはあまりない。しかし、攻撃者によって機能が停止させられたり、遠隔で操作されたり、攻撃の踏み台に利用される場合が出てきている。

最近のスマートテレビは、内蔵カメラを持っているが、このスマートテレビがハッキングされ、盗撮も可能であることが大きなカンファレンスで公開された<sup>6</sup>。盗撮された映像は、インターネットを通じて、広く出回ってしまう可能性がある。テレビの画面に流れるテロップに、デマ情報を流すデモも行われ、話題になった。

これまでセキュリティとは無縁と思われてきたコーヒーメーカーにも脆弱性が見つかっている<sup>7</sup>。外部からコーヒーの濃さや水量などを変更することが可能と発表された。これだけであれば、大きな問題にはならなかったかもしれないが、この脆弱性を利用して、ユーザーのパソコンにアクセスが可能となり、問題は広がった。さらに、この製品はソフトウェアにパッチをあてることもできないため、修正することができず、現在も脅威にさらされたままになっている。

<sup>6</sup> <http://www.cnn.co.jp/tech/35035482.html>

<sup>7</sup> <http://jvndb.jvn.jp/ja/contents/2009/JVNDB-2009-004384.html>

## ■スマート家電に必要な対策

スマート家電には、セキュリティ対策が必須であるだけでなく、出荷後に見つかった脆弱性を修正するためのソフトウェア更新のしくみを入れておくことも重要である。

ソフトウェアを更新するしくみは、製品の持つ機能を進化させるという点ではメリットがある。これまでの製品は、工場出荷時に機能が決まり、出荷後に機能を追加することはできなかった。新製品に新しい機能が追加されると、その機能が欲しいユーザーは、製品を買い換えるしかなかった。スマート家電では、ソフトウェアで実現する機能を出荷後にでも更新することができるようになり、欲しい機能が追加される機会も増えてくる。また、ユーザーごとに異なるニーズに対応した機能(ソフトウェア)を入れることも可能となる。

例えば、工場出荷時には発売されていなかった新しい洗剤に対応した洗濯コースを追加したり(洗濯機)、新しい料理に対応したレシピを追加したりすること(電子レンジ)ができる。ユーザーごとに、希望するレシピを入れることもできる。

表 1.5-1 スマート家電の特徴

| スマート家電の特徴 |   |
|-----------|---|
| セキュリティ    | ネットワークを介して、機器の遠隔操作や状態確認などができ、ユーザーの利便性は大きく向上する。しかし、ネットワーク経由で、外部から攻撃される可能性も出てくるため、セキュリティに対する対策は非常に重要となってくる。 |
| 機能追加      | 新しい機能を製品購入後にも追加することができ、ユーザーにとって欲しい機能を手に入れやすくなる。一方、機能追加のしくみに問題があれば、製品が使えなくなる可能性もあり、十分な検証が必要となる。            |

これまでは製品単体で機能していたものが、ネットワーク環境を含めたシステムとして、機能を実現するようになり、システム全体で、品質を確保する必要がある。システムの一部分がうまく動かないと、全体として、機能しなくなることもある。また、一つの会社だけでシステムを提供するとは限らず、接続検証が重要になってくる。

今後、スマート家電は、ますます機種が増えてくることが予想されている。家中の家電がつながり、スマート家電からスマートホーム、そしてスマートシティへ展開されてくると、IoT の一翼を担う重要な生活基盤になってくる。品質への要求は、さらに高くなってくる。

## コラム

## IoT時代の品質課題

昨今、ITの分野において、IoT (Internet of Things) という言葉が頻繁に使われるようになった。日本語では“モノのインターネット”と訳され、今のところ、広く認知された定義はないが、概ね以下のような意味で使われることが多い。

「機械、自動車、医療機器、家電、日用品など様々なモノが、インターネットに接続し、個体情報やセンサー情報をもとに、他のデバイスやサービスとつながることで、必要な情報の入手や、あるいは制御を可能とする技術、機器/デバイス、サービス等の総称」

今後、IoTは、生産や流通などビジネスの仕組みだけでなく、エネルギーや交通システムのような社会インフラや、一般消費者の身近な生活環境を大きく変える可能性があると言われており、新たなビジネス機会の拡大や、人々の生活を豊かで便利なものにしていくという期待が大きい<sup>8</sup>。

一方で、様々な機器やサービスの普及に伴うリスクの増大も懸念されており、我が国においても、セキュリティやプライバシーの問題を中心に、今後の対策を具体化する動きが進められている。

<sup>8</sup> 「IoT時代に対応したデータ経営 2.0 の促進」(経済産業省 産業構造審議会 商務情報流通分科会 資料)

[http://www.meti.go.jp/committee/sankoushin/shojo/johokeizai/pdf/001\\_03\\_00.pdf](http://www.meti.go.jp/committee/sankoushin/shojo/johokeizai/pdf/001_03_00.pdf)



PC がインターネットを介してウイルスに感染したり、乗っ取られたりする危険があるのと同様に、あらゆる IoT デバイスが攻撃の対象となり、悪質な攻撃により社会システムが機能不全になる可能性もある。また、IoT デバイスを使う個人のプライバシーにかかわるデータが、本人の認識や承諾なく収集され、あるいは流出する危険性が指摘されている<sup>9</sup>。

IoT に関しては、様々な標準化の動きも活発化しており、国内外の事業者による民間レベルのコンソーシアムが複数存在するほか、ISO/IEC JTC1 において国際標準化の取組みもスタートした<sup>10</sup>。

---

<sup>9</sup> 「我が国のサイバーセキュリティ推進体制の機能強化に関する取組方針」(内閣サイバーセキュリティセンター)

<http://www.nisc.go.jp/conference/seisaku/dai41/pdf/houshin20141125.pdf>

<sup>10</sup> <https://www.itscj.ipsj.or.jp/oshirase/2015/01/20150122.html>



## 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(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考えから 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が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>1</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6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、強化する取り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、整備されてきている。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>2</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>1</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>2</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

■ 新規    ■ 改訂

| 区分    | 期    | 第1期<br>【1985～1991】 | 第2期<br>【1992～2004】  | 第3期<br>【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<br>◆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<br>◆ソフトウェアの利用時の品質副特性の定義の追加<br>◆品質モデルの再編<br>◆データの品質特性の定義<br>◆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<br>◆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<br>◆局面、主体者に応じた品質評価プロセスの定義<br>◆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定<br>◆局面、主体者に応じた品質評価プロセスの定義の改訂<br>◆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<br>◆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 SQaRE の構造と概要

SQaRE 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。コア部分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。SQaRE 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、それ自身が改訂・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。

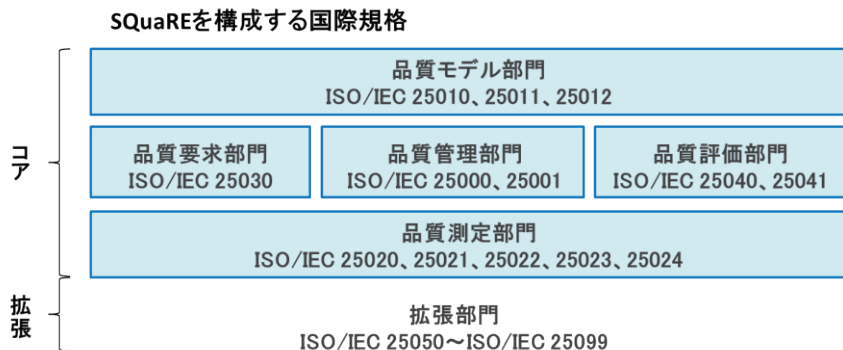


図 2.2-1 ソフトウェア製品評価規格の発行状況(含:JIS)

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<br>(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<br>(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<br>(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<br>(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<br>(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



## 他標準との関連

システム及びソフトウェアの開発から運用・保守に至るライフサイクルプロセスを規定した『ISO/IEC 15288』、『ISO/IEC 12207』、『共通フレーム』などでは、品質要求定義と評価に関連するプロセス、アクティビティ、タスクの概要が定義されている。SQuaRE は、それらのプロセス、アクティビティ、タスクに対して、システム及びソフトウェアの品質の視点から、より詳細かつ具体的な作業要件や進め方を提供している。

プロセスアセスメントを規定した『ISO/IEC 15504 シリーズ』、『ISO/IEC 33000 シリーズ』、『CMMI』などでは、プロセスの能力や組織の成熟度の評価や改善に用いるアセスメントモデルの中で、要求定義と評価に関連するプロセス領域、プラクティスが定義されている。SQuaRE は、それらのプロセス領域、プラクティスを品質モデルや品質メトリクスを用いて、より効果的に実践するための方法を提供している。

品質マネジメントシステムを規定した『ISO/IEC 9001 及び 9000 ファミリー』では、その要求事項の1つとして製品及びプロセスの測定、分析及び改善の仕組みを構築し運用することを求めている。SQuaRE は、その要求に応じて品質モデルを用いて、システム及びソフトウェア(製品)の品質測定、分析及び改善を実行する方式を提供している。

## 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

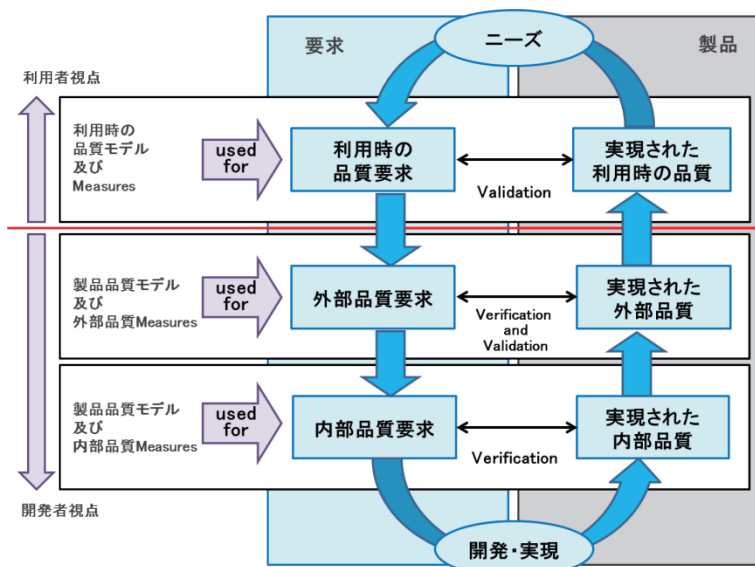


図 2.3-1 システム&ソフトウェア品質ライフサイクル

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「3.6 データ品質」で説明する。

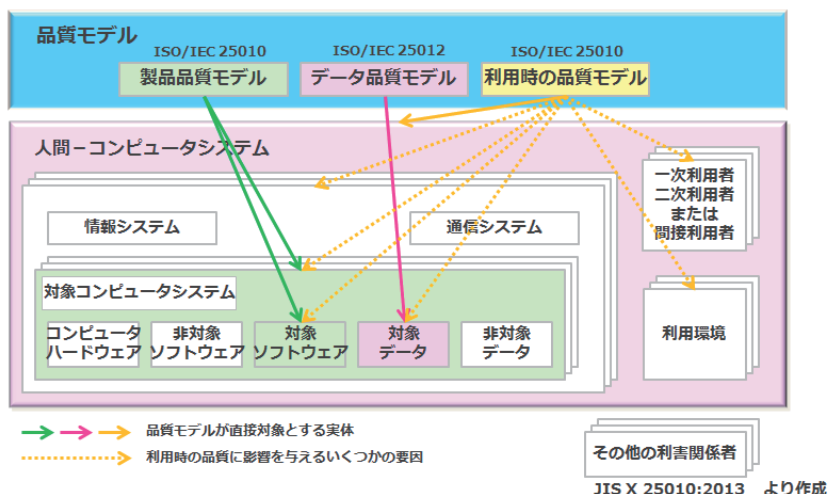


図 2.3-2 品質モデル

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

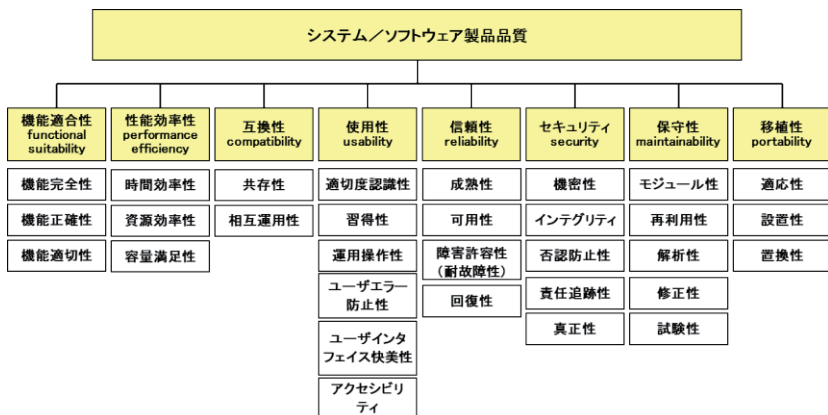


図 2.3-3 システム・ソフトウェア製品の品質モデル

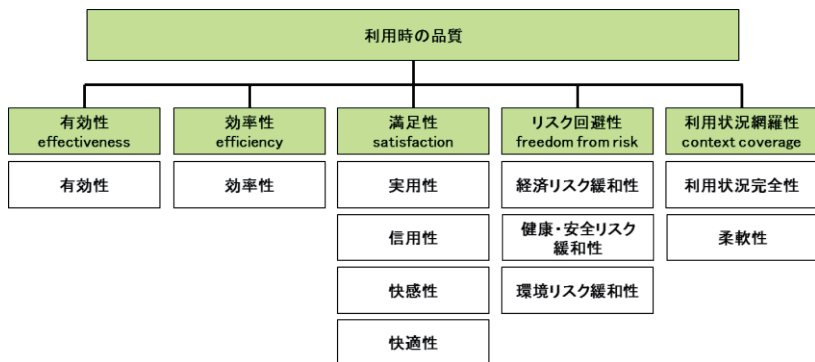


図 2.3-4 利用時の品質モデル

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 SQaRE の活用

SQaRE 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“SQaRE 品質モデル活用リファレンス編”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(詳しい説明は第4章を参照)。SQaRE 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。

SQaRE 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 SQaRE を用いる。要求定義の段階では、SQaRE 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、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。また、その品質に関連が深い産業分野における品質向上の取組み事例を紹介する。

- 
- 3.1 ユーザビリティと利用時の品質
  - 3.2 ユーザビリティを向上させる取組み **事例**
  - 3.3 セキュリティ／セーフティ
  - 3.4 自動車分野 **事例**
  - 3.5 医療・ヘルスケア分野 **事例**
  - 3.6 データ品質
  - 3.7 クラウドサービス分野 **事例**
-

## 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)

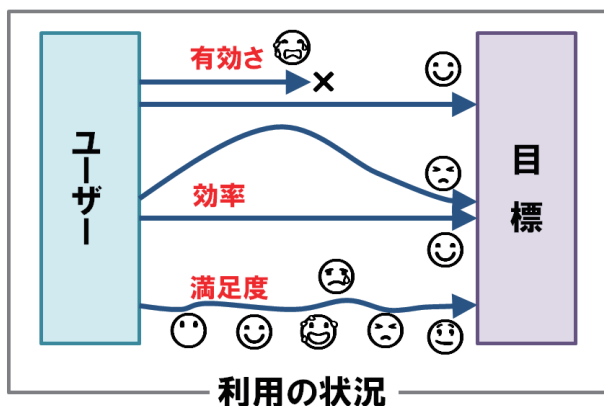


図 3.1-1 ユーザビリティの定義より(ISO 9241-11、9241-210)

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(フィーチャーフォン)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 参照)。



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 ISO 9241-210 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

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

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

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

HCD-Net 認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。

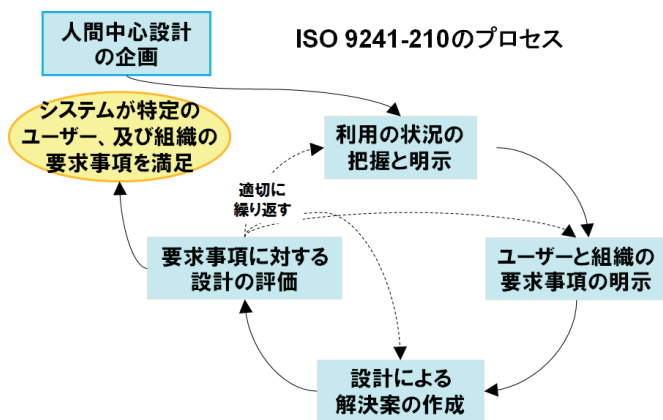


図 3.1-3 ISO 9241-210 のプロセス

(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 ユーザビリティを向上させる取組み【事例】

### ■「利用時の品質」が高いとは

例えば、初心者が銀行の ATM で振り込み手続きをする際に、ATM から提示された情報や操作手順を迷うことなく理解でき、誤った操作をすることなく、期待した時間内にスムーズに手続きを終えられるなら「利用時の品質」は高いといえる。結果として利用者の作業時間、待ち時間を最小限に抑えることができれば、銀行としてのサービス(ここでは利用者の拘束時間を短くする能力)が高いことを示し、顧客満足度を高め、顧客の離反を招く不満要素の削減に繋がる。このようなユーザビリティを向上させ、かつ、その良し悪しを測るにはどう取り組みれば良いだろうか。

### ■「利用時の品質」のユーザビリティ観点を作り込むために

SQuaRE では「利用時の品質」と関連が深い製品品質の特性として「使用性」副特性が定義されており、これを十分に理解し活用することが大切である。

一方、ユーザビリティ観点での検討項目としては、有名な Jakob Nielsen

### 車載機器HMI領域における使用性の検討項目例

| 大分類          | 中分類  |
|--------------|--|
| A. ユーザー特性    | A1. 初心者一般 A2. 熟練者一般 A3. シルバー世代<br>A4. 視覚、聴覚、身体障害 (ほか10特性)  |
| B. 「知覚」領域の特性 | B1. 視認性 B2. 可聴性 B3. 触覚   |
| C. 「判断」領域の特性 | C1. 適切な表示・表現(用語・メッセージ) C2. 構造化<br>C3. 記憶負担の軽減 C4. エラーへの対処 C5. 一貫性<br>C6. 学習性(ほか1特性) C7. 誘導性(ほか1特性) |
| D. 「操作」領域の特性 | D1. 身体適合 D2. 効率性 D3. 安全 (ほか4特性)  |
| E. 魅力性       | E1. 快適性 E2. 意匠性  |

表 3.2-1 車載用 HMI の検討項目例(株式会社 U'eyes Design による)

による「ヒューリスティック評価」の 10 原則<sup>4</sup>や、国際規格 ISO9241-110「人間工学—人とシステムとのインタラクション—対話の原則」<sup>5</sup>内で規定された 7 原則などがある。対象とする分野において、ユーザビリティを具体的に考察するためには、これらの手法を参考にしながら、多岐に渡る検討項目を分類整理して取り組む必要がある。

表 3.2-1 は、株式会社 U'eyes Design による、車載用 HMI の分野における検討項目の分類例である。この例では、車載用 HMI の検討では安全運転に必須の「知覚・判断・操作」を中心的項目として掲げている。またユーザーを明確に定めることを重視し、ペルソナ<sup>6</sup>の設定の仕方に関する質の良し悪しを評価する「ユーザー特性」を加え、さらに、商品性として欠かせない「魅力性」を加えた 5 つの大分類で構成している。

次に、中分類としては、先にあげたヒューリスティック評価 10 原則を扱  
**SQuaRE「使用性」副特性とユーザビリティ検討項目との対応関係**

◎副特性に強く関係する検討項目  
 ○副特性に関係する検討項目  
 △事象によって関係する検討項目  
 -関係がない、もしくは薄い項目

A1. 初心者一般 A2. 熟練者一般  
 A3. シルバー世代 A4. 視覚、聴覚、身体障害  
 B1. 視認性  
 B2. 可聴性  
 B3. 触覚  
 C1. 適切な表示・表現(用語・メッセージ)  
 C2. 構造化  
 C3. 記憶負担の軽減  
 C4. エラーへの対応  
 C5. 一貫性  
 C6. 学習性  
 D1. 誘導性  
 D2. 身体適合  
 D3. 安全性  
 E1. 快適性  
 E2. 魅力性

|            |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 適切度認識性     | ○ | ○ | ○ | - | - | ◎ | ○ | - | - | - | - | - | - | - | - | - | - | - |
| 習得性        | ◎ | ○ | △ | △ | △ | △ | △ | - | - | ◎ | ◎ | ◎ | △ | △ | - | - | - | - |
| 運用操作性      | ◎ | ○ | ◎ | ◎ | ◎ | ◎ | ◎ | - | ○ | - | - | ◎ | ◎ | ◎ | - | - | - | - |
| ユーザーエラー防止性 | ◎ | ○ | △ | △ | △ | △ | △ | ◎ | △ | △ | △ | △ | △ | - | △ | - | - | - |
| ユーザーI/F快美性 | ◎ | ○ | △ | △ | △ | △ | - | - | - | △ | - | - | △ | - | - | - | ◎ | ◎ |
| アクセシビリティ   | ○ | ◎ | △ | △ | △ | △ | - | - | - | - | - | - | △ | - | - | - | - | - |

表 3.2-2 「使用性」副特性とユーザビリティ検討項目との対応関係  
 (株式会社 U'eyes Design による)

<sup>4</sup> <http://www.nngroup.com/articles/ten-usability-heuristics/>

<sup>5</sup> JIS Z 8520

<sup>6</sup> <http://www.usability.gr.jp/lecture/ud-and-persona.html>

<http://www.usability.gr.jp/lecture/think-persona-psychologically.html>



い易くした「構造化ヒューリスティック<sup>7</sup>」を採用している。これらの項目は、それぞれの製品領域や開発部門の文化に沿うよう、また、新規開発の場合や部分的な改善を行う開発にも対応できるよう、構造的に分類しておくとう用である。

この分類例に関して、各検討項目とSQuaREの「使用性」副特性項目との対応関係は表 3.2-2 のようになる。この表で、ユーザー特性に分類した各項目は、すべての副特性に共通して検討すべきものとして位置づけている。

### ■「利用時の品質」を測る

「利用時の品質」はユーザビリティに取り組んだ成果であり、それを測る手法がユーザビリティの専門領域にて提唱されている<sup>8</sup>。代表的なものとして、タスク達成度評価、NEM<sup>9</sup>、SUS、CIF があり、NHTSA<sup>10</sup>から提唱された Distraction Guide lines がある。「利用時の品質」を開発プロセス内で予測するためには、模擬的に実際の利用環境に近い環境において、実ユーザーが商品を利用する状況を再現して測定する。本節では「使用性」との関係性を確認できる NEM(どの操作機会で問題が発生しているか、および、その程度を確認できる定量的評価指標)を例に「利用時の品質」の「効率性」の改善事例について紹介する。

<sup>7</sup> "問題発見効率の高いユーザビリティ評価法-1.構造化ヒューリスティック評価法の提案-" ヒューマンインタフェースシンポジウム(1997) p481-p488

<sup>8</sup> ユーザビリティハンドブック(共立出版)、ユーザビリティテスト(共立出版)

<sup>9</sup> Novice Expert ratio Method

<sup>10</sup> NHTSA: National Highway Traffic Safety Administration (米国運輸省道路交通安全局)

## ■「銀行 ATM 振込操作」における効率性の問題点

図 3.2-1 は、銀行 ATM を利用して物品の購入費を販売者へ振り込む場面に対して NEM を適用した例である。利用者の振り込み処理における一連の操作手順(キャッシュカードを挿入し、複数の操作機会を経て振込み作業が完了する)を横軸に示し、縦軸は各操作に要した操作時間である。棒グラフは、設計者(熟練者)の操作時間を 1 として一般利用者(初心者)の操作時間との比  $R_i$ (以下、NE 比<sup>11</sup>)を表したものである。NE 比が 4.5 倍以上ある操作機会にはユーザビリティ上、致命的な欠陥があると判定<sup>12</sup>される。操作ステップ 2 の NE 比が 10 倍を超えたのは、この操作画面内の用語が不適切でユーザーに意味が伝わらず、次の操作ステップに容易に遷移できなかつたことが理由である。

この対策としては、表 3.2-2 で示した「C1 適切な表示・表現」を考慮する必要がある。この事例では、銀行が商品サービス用語として独自に考案した名称が画面に表示されていたことが原因であった。どの用語を選べば「C1 適切な表示・表現」が改善するかは画面仕様設計の段階で検討・調査し、NE 比の値が下がる用語を採用する。

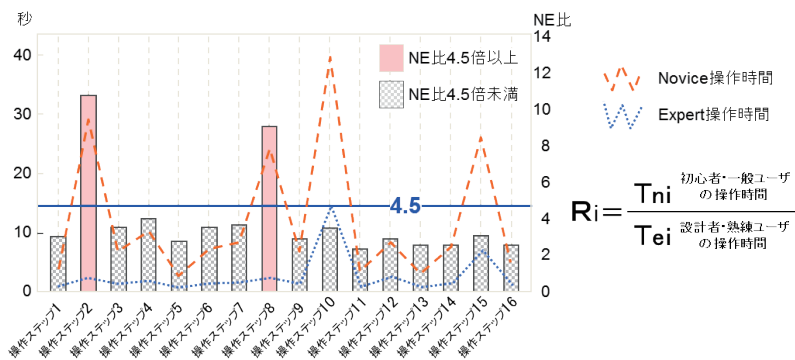


図 3.2-1 銀行 ATM 振込操作の場合(株式会社 U'eyes Design による)

<sup>11</sup> NE 比の考え方は電子政府ユーザビリティガイドラインにも紹介されている。

[http://www.kantei.go.jp/jp/singi/it2/guide/security/kaisai\\_h21/dai37/h210701gl\\_f.pdf](http://www.kantei.go.jp/jp/singi/it2/guide/security/kaisai_h21/dai37/h210701gl_f.pdf)

[http://www.kantei.go.jp/jp/singi/it2/guide/kaisai\\_h20/dai2/siryou4.pdf](http://www.kantei.go.jp/jp/singi/it2/guide/kaisai_h20/dai2/siryou4.pdf)

<sup>12</sup> <http://ueyesdesign.co.jp/case/paper/upa2002.pdf>

## ■「カーコックピット操作」における効率性の問題

図 3.2-2 は自動車コックピット内におけるカーエアコン操作に対して NEM を適用した例<sup>13</sup>である。不適切な用語(「MODE」が「風向き変更」の意味を伝えられない)や、曇りを除去する対象がフロントかリアかの違いを見分けにくいピクトグラム(サイン)デザインが要因となって NE 比が高くなっている。ピクトグラムやアイコンデザインの意味が的確に伝わるためには表 3.2-2 の「B1 視認性、C1 適切な表示・表現」を高める対策を施せば良い。

また、自動車コックピットにおける操作では安全運転が優先され、エアコンなどの従たる操作が運転操作を阻害してはならない。このような多重課題を伴う場合は NE 比が 4.5 倍未満であっても安全運転を確保できないので 2 倍未満にする必要がある。

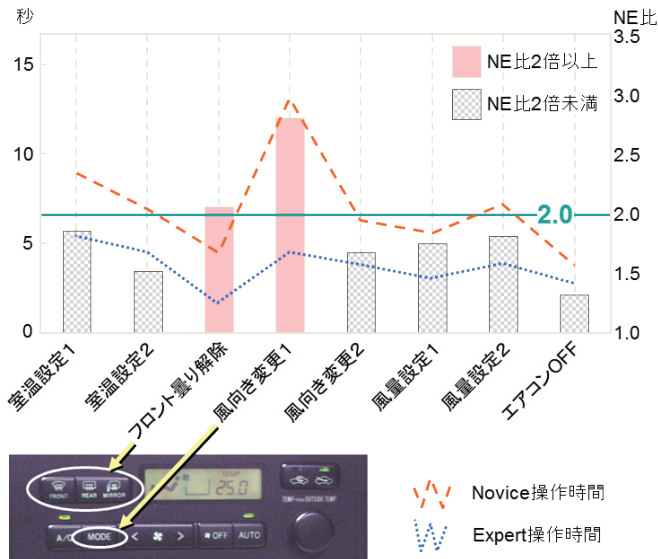


図 3.2-2 カーエアコン操作の場合(株式会社 U'eyes Design による)

<sup>13</sup> <http://dl.acm.org/citation.cfm?id=633394>、  
[http://ueyesdesign.co.jp/case/paper/nem\\_chi2000.pdf](http://ueyesdesign.co.jp/case/paper/nem_chi2000.pdf)

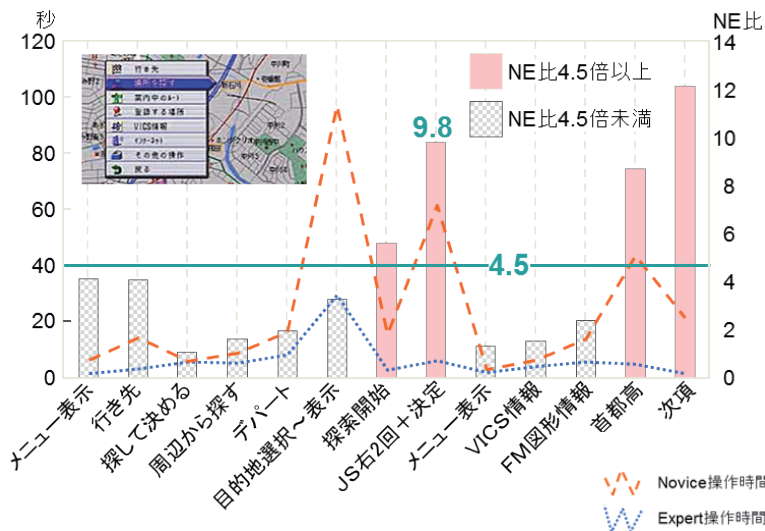


図 3.2-3 カーナビの設定操作の場合

図 3.2-3 は停車時に集中して作業できるカーナビの設定操作の例である。多重課題ではないので、ATM や後述の WEB サイトの事例と同様 NE 比 4.5 倍未満を目指して良い。棒グラフの中央付近にある NE 比が 9.8 倍と顕著に高い操作は「ジョイスティックを同じ方向に 2 回連続で倒し項目を選択してから決定ボタンを押す」という操作方法がユーザーに伝わらなかった例である。新たなサービスが追加され多機能になると操作が複雑になる。新機能を既存の操作デバイスで扱おうとする際に工夫を凝らして機能をアサインするケースは多いが、ユーザーの推測できない操作仕様を採用すると効率性が高まらないだけでなく、課題の達成そのものができなくなる。この例では表 3.2-2 の「B3 触覚、C5 一貫性、C7 誘導性」など複合的な対策を立てることで「効率性」の低下を防ぐことができる。

### ■「地方自治体 Web サイト」における効率性の問題

図 3.2-4 は HCD-Net<sup>14</sup>が「地方自治体 Web サイトのユーザビリティ評価

<sup>14</sup> 特定非営利活動法人人間中心設計推進機構

2009～子育て&引越タスク編～<sup>15</sup>」を実施した例である。この事例では、

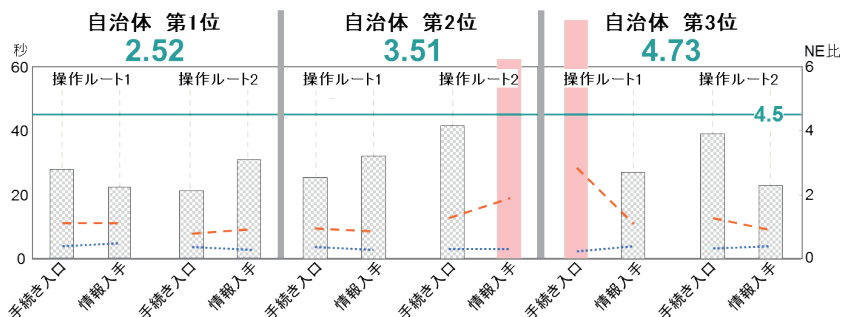


図 3.2.4 Web サイトの場合

課題作業として「小学生の転校(引越し時の転入)手続きに必要な情報を Web サイトから見つける。」を設定し、NE 比を測定している。自治体で第 1 位になったサイトの NE 比は 2.52 倍であり、これは前年に同測定を行った際よりも値が小さくなっており、Web サイトが改善され「効率性」が向上していく様子を値で確認することができた。NE 比は 4.5 倍未満であれば合格という基準ではなく、可能な限り NE 比の値を小さくする工夫を施すと良い。

上位 3 つのサイトの課題達成率は共に 92% と同じであったが、NE 比には違いが生じており効率性の差が明らかとなった。対策には「使用性」の検討項目を複合的に活用し、結果として「利用時の品質」の向上に繋げることが可能となる。

### ■「利用時の品質」向上のために

本節では「利用時の品質」を向上させるため、製品品質「使用性」副特性を総合的にマネジメントすべきことを示した。ユーザビリティの専門領域では、そのための多くの手技法が開発され、活用が進んでいるが、開発現場における認知度や普及度は十分ではない。古くは人間工学の知見がハードウェアの設計に有効に働いたように、ユーザビリティのノウハウや、

<sup>15</sup> <http://www.hcdnet.org/event/web2009.php>

多くの実践例から得られる知見が、ソフトウェア品質の向上に必ず役立つ。こうしたノウハウを効果的に利用するためには、ユーザビリティの専門家<sup>16</sup>を計画的に育成することに加え、外部の専門家を積極的に活用する方法もある。

---

<sup>16</sup> HCD-Net 認定 人間中心設計専門家資格 <http://www.hcdnet.org/certified/about/>

### 3.3 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、個人情報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<sup>17</sup>。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などを考慮する

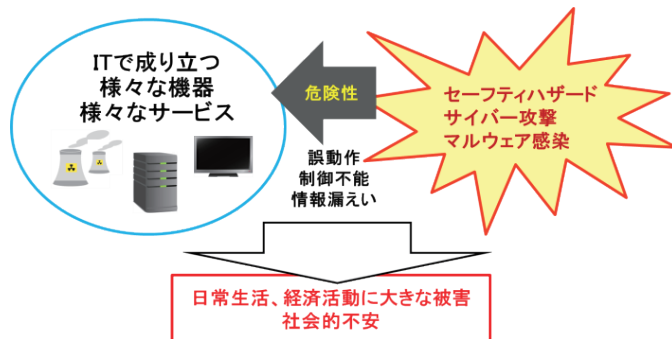


図 3.3-1 セーフティとセキュリティの重要性

<sup>17</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。

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 3.3-2 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、

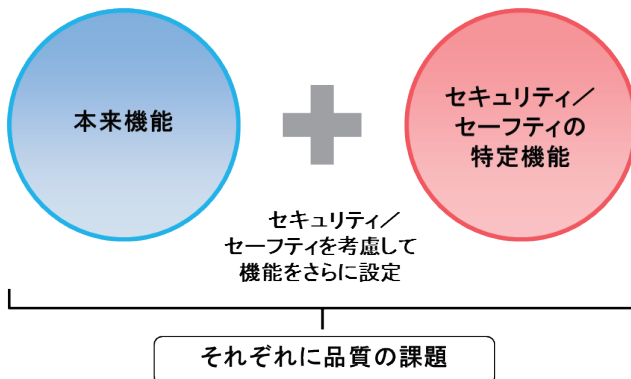


図 3.3-2 本来機能と特定機能



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>18</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>19</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、安全度水準(SIL)を用いることが多い。安全が重

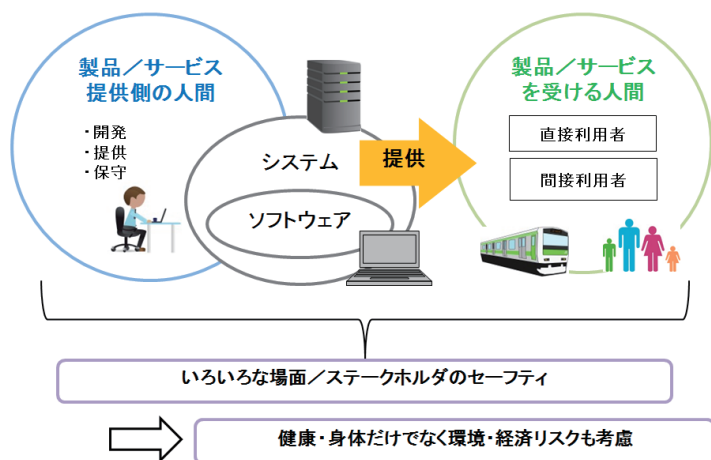


図 3.3-3 システムのセーフティとさまざまなステークホルダ

<sup>18</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>19</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。

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 3.3-3 参照)。

図 3.3-3 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>20</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>20</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>21</sup>のセキュリティ機能要件の中にも、SQuaRE での品質特性(副特性)にあたる、機密性、インテグリティ<sup>22</sup>、真正性といった項目の対応項目が含まれる(図 3.3-4 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>23</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 3.3-4 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>21</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>22</sup> 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>23</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 3.3-4 で示すような特性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>24</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 3.3-4 で示した真正性<sup>25</sup>やインテグリティに関して、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>26</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

<sup>24</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>25</sup> 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。

<sup>26</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。

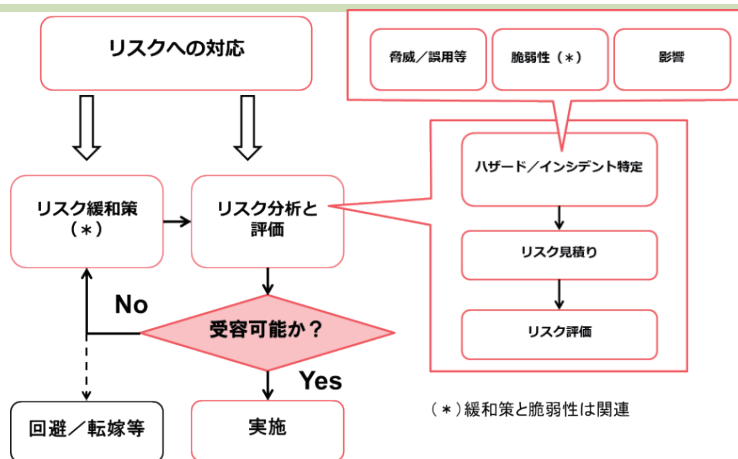


図 3.3-5 セキュリティ/セーフティのリスク評価とリスク緩和の反復プロセス

い(図 3.3-5 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>27</sup>。

<sup>27</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。

## 3.4 自動車分野【事例】

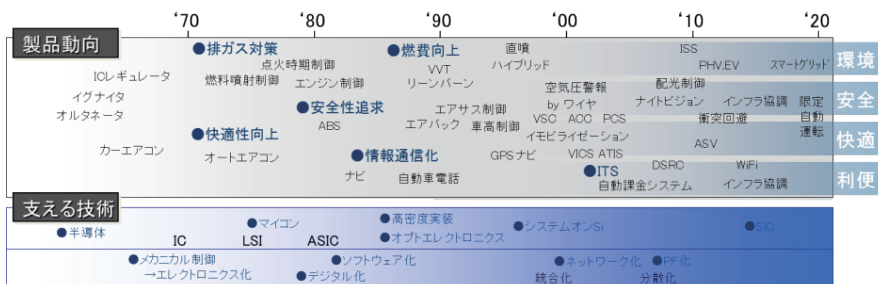
### ■車載電子システムの動向

社会と自動車の関わりを考える上で、「環境」「安全・安心」の視点は欠かせない。

自動車は「環境」と常に向き合いながら発展をとげてきた。1970年代から導入された排ガス規制をきっかけとしてエンジン制御の電子化が始まった。今日では温暖化防止に向けたCO<sub>2</sub>排出量削減のため、燃費改善に加えて環境負荷の低いエネルギーを利用することが求められている。このためハイブリッド車、電気自動車、燃料電池車といった新たな技術を導入した自動車が開発され製品化されている。

「安全・安心」も自動車にとっては重要な視点である。これまでも機構の改良や安全装置を工夫することで、交通事故による死亡者数を削減してきた。今日では、事故そのものを削減するためにドライバーの認知・判断・操作ミスを冒さないように支援する技術、衝突が避けられない場合に被害を最小限に抑える技術が実用化されている。

これからの環境技術や安全技術は、情報通信や電力といった社会インフラとつながりながら発展していく。高度な運転支援には、車車間・路車



"高信頼車載電子システムの安全性とソフトウェア", SECjournal No.38 より引用

図 3.4-1 車載電子システムの発展



間通信といった技術が使われ始めており、環境面では自動車の充電を電力システムの中で最適化するマイクログリッドも実用化が始まっている。今後は、スマートフォンやクラウドといった IT 技術との連携により、新たな自動車の利用方法やサービスの創出が加速的に進む(図 3.4-1)。

このような先進的なシステムはソフトウェアにより実現されている部分が多い。車載ソフトウェアは、年々指数関数的に増加してきており、2015年には総ソースコード行が1億行/1台を越える。これに対応する為に ISO 16949 や SQuaRE 等を参照しながら、機能性・信頼性・効率性・保守性・移植性といった特性の改善に取り組んできた。今後は、前述したように、人命に直接的に関わる高度運転支援・車内外の通信を活用した新たなサービスが普及するにつれ、セーフティ・機密性(セキュリティ)といった特性が、ますます重要になってくる。以下、車載ソフトウェアのセーフティ・セキュリティについて述べる。

### ■自動車向け機能安全規格(ISO 26262)

自動車業界では、機能安全規格ISO 26262の制定により、車載電子システムの安全に関する標準的な考え方が確立された。

ISO 26262では「モノが壊れること」(偶発的故障)と「人がミスをする事」(系統的故障)の2点を前提に安全対策を行う。偶発的故障はハードウェアの通常の意味での故障であるため、故障率を許容レベル以下に抑える(信頼性をあげる)か、安全装置や安全機構を追加することで危害の発生確率を抑制するといった設計上の対策を取る必要がある。一方、系統故障は設計上の考慮漏れや設計ミスなどに起因するために、開発プロセスの中で必要な対策を行う。

品質マネジメント体系(ISO TS 16949)に基づく車載電子システムの安全性は、安全性の作りこみを行うことを重視してきたが、ISO 26262では安全性を規格に照らし合わせて説明できることが必要となる。



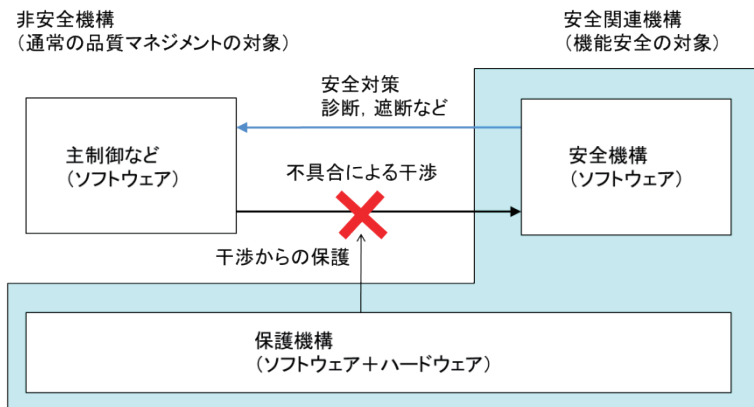
## ■車載ソフトウェアの安全性

ソフトウェアの不具合は系統的故障に分類される。しかし、開発プロセスを整備すれば十分であるわけではなく、ソフトウェアに対しても安全設計を行うこととその検証を行うことが要求されている。

昨今のような複雑なシステムに対しては、アーキテクチャレベルで安全構想を基本設計に落としこんでおくことが重要となる。アーキテクチャレベルで安全対策を行うためには、安全要件とその安全対策を明確にすることと、安全機能とそれ以外の非安全機能(主制御など)を明確に分離しておくことなどがポイントである。

ソフトウェア不具合のうちメモリ破壊や暴走につながるものは、その影響の解析が一般には難しいが、保護機構もしくは監視機構などを用いて非安全機構の不具合からの干渉を受けない構成を取ることで、安全機構を分離することができる。この場合には、干渉に関する安全性解析を安全機構に対して行うことで、非安全機構も含めた安全性の確認ができる。

このようにソフトウェアであっても、安全構想から基本設計に落とすレベルで対策を考慮しておくことで十分に安全性の説明が可能となる(図3.4-2)。

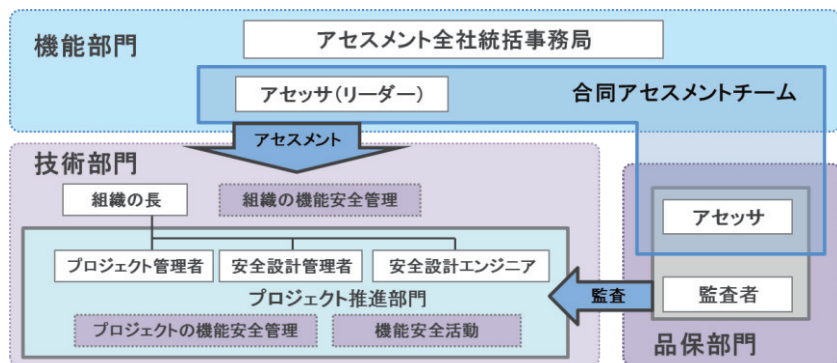


"高信頼車載電子システムの安全性とソフトウェア", SECjournal No.38 より引用

図 3.4-2 ソフトウェア安全性の設計への織り込み

## ■安全マネジメント

ISO 26262は安全設計、開発プロセスだけでなく、製品ライフサイクル全体の観点から、組織面やマネジメント面での要件も規定している。そのため機能安全に対応するためには企業活動の広範囲に亘った対応が必要となってくる。機能安全への組織的対応としての一例を図3.4-3に示す。



"高信頼車載電子システムの安全性とソフトウェア", SECjournal No.38 より引用

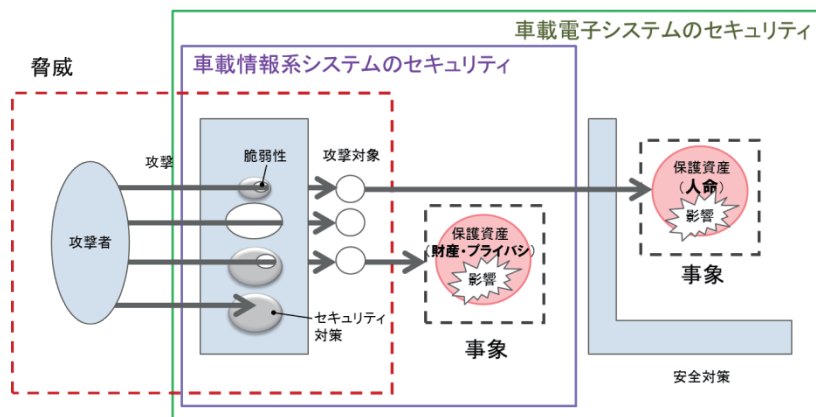
図 3.4-3 機能安全への組織的対応

## ■セキュリティ

自動車は情報系技術を取り込みながら社会とつながる存在に変わろうとしている。車載電子システムがネットワークに常時接続することにより、様々な利便性をサービスとして実現できるようになってきていると同時に、外からの脅威にも晒されることになる。セキュリティの観点からはこのような脅威に対する対策が必要となる。

脅威には大きく分けて、①財産・プライバシーに関わるものと②人命に関わるものがある。財産・プライバシーに関わる脅威については、情報システムを対象とした標準や技術が既に存在しているため、社会インフラとの調和を考慮してこれらの標準技術を取り入れる必要がある。一方で、人命に関わる脅威については、ISO 26262の観点から安全性を担保する必要がある。

車載電子システムにおけるセキュリティ対策は、安全設計を中核として情報システムの技術・手法を取り込むアプローチが現実的である。具体的には脅威分析の段階で財産・プライバシーなどの従来の情報セキュリティの観点に加えて人命を考慮した分析を行い、人命に関わる脅威が識別できた際には ISO 26262 の観点から安全性の検証を行って必要な品質を担保することである(図3.4-4)。



"高信頼車載電子システムの安全性とソフトウェア", SECjournal No.38 より引用

図 3.4-4 車載電子システムのセキュリティ

## ■まとめ

自動車は外とのつながる機能・性能の進展により、社会システムの一構成要素としての色合いが濃くなる。情報セキュリティ含め自動車が安全であることを、アーキテクチャ・プロセス面から担保することに加えて、社会に対してその証明・説明をすることが今後ますます重要となる。

## 3.5 医療・ヘルスケア分野【事例】

### ■つながる医療・ヘルスケア ～医療 ICT について～

患者及び様々な医療従事者が医療情報を共有し活用していく仕組み、「医療情報連携ネットワーク」が広がりつつある。将来、一人の患者に一つの電子カルテとなり、健康管理情報から、通院記録、診断記録、処方箋、診断画像など、全てのデータが一元的に管理され、日本中のどこの病院でもアクセスできる時代がくると考えられている。ただし、患者情報の取り扱いには極めて重要で、万が一の場合でも医療事故が起こってはならないし、データ連携の標準化もやさしいことではない。技術的にも、先進医療機器から、各種センサー、スマートフォンやタブレットなどのモバイル端末、医療クラウドといった幅広いデバイス・システムの連携が考えられる。私たちの QOL(Quality Of Life)にとって重要な役割を果たす医療

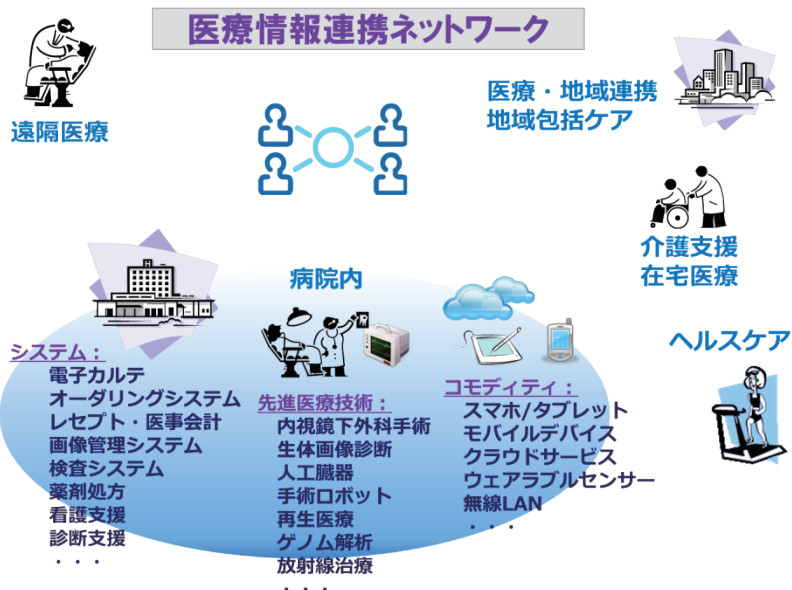


図 3.5-1 つながる医療・ヘルスケア

ICT だが、このように社会的にも技術的にも高度で複雑な関係を併せ持つために、その変化は一度に起こるものではなく、部分的にデザインされたものが徐々に進化していき、お互いにつながりあい、大きなシステムへと発展していくものと考えている。

### ■医療機器に求められることは ～リスク回避性～

生命にかかわる医療機器には、高い安全性と信頼性が求められている(図 3.5-2)。そのため、その開発においては、利用のされ方をよく理解するとともに、リスクにしっかり対処することが重要になっている。利用する人の意図した通りに動くことは当然として、トラブル時を含めたあらゆるユースケースを想定し、万が一正しく動かなくなってしまった場合でも、患者や医療従事者への損害を限りなく少なくすることが必要となっている。そのために医療機器の開発においては、きめの細かい仕様を設定し、それを確実に検証し、品質を高めていくことが求められている。

### ■医療機器に関連した国際規格・各国規制事例

安全性と品質を担保するために、医療機器に対しては、様々な国際標準規格や各国の規制があり、それらを遵守していくことが求められている(図 3.5-2)。医療法規制の考え方では、製品そのものの品質を保証するだけでなく、それを作り込むプロセスが維持できていることも証明する必要がある。客観的根拠に基づきながら設計仕様やトレーサビリティを保証していかななくてはならない。また、製品の安全性をソフトウェアの観点から規制する動きも世界中で高まっており、診断・治療等を目的としたプログラムは単体であって法規制への対応が求められている。ここでは、ソフトウェアそのものの品質確保も大切だが、そのソフトウェアを動作させるシステムへの配慮も必要で、特にコンピューターウイルスやハッキングの不正攻撃から守る仕組みも含めた総合的な品質確保が重要になっている。

## ■医療機器のクラス分類

医療法規制では、図 3.5-3 に示すように人体に与えるリスクの程度によって医療機器を分類し、そのクラス分類ごとの規制基準をもって管理を行っている。人の生命・健康に重大な影響を与えるおそれがあるクラスの医療機器にはより高度な管理が求められている。ここで注意しなくてはならない点は、品質管理基準に関し、クラスが高ければ高品質で、クラスが低ければ品質が低くても構わないということの意味するものではない。クラス I ではバグが 5 つまで許されるというのではなく、各クラス分類に応じて必要な品質管理の厳格さが要求事項として設定されており、その各々への対応が求められるということの意味している。

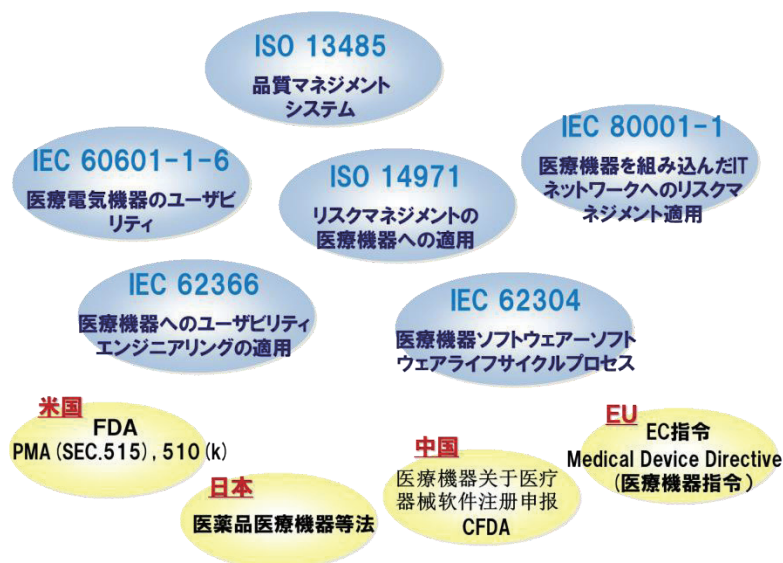


図 3.5-2 医療機器に関連した国際規格・各国規制事例

| 小 ← リスク → 大  |  |   |   |   |
|--------------|--|---|---|---|
| 国際分類<br>(注1) | クラスⅠ   | クラスⅡ  | クラスⅢ  | クラスⅣ  |
| 具<br>体<br>例  | 不具合が生じた場合でも、人体へのリスクが極めて低いと考えられるもの<br><br>(例) 体外診断用機器、鋼製小物(メス・ピンセット等)<br>X線フィルム、歯科技工用品<br> | 不具合が生じた場合でも、人体へのリスクが比較的低いと考えられるもの<br><br>(例) MRI装置、電子内視鏡、消化器用カテーテル、超音波診断装置、歯科用合金<br> | 不具合が生じた場合、人体へのリスクが比較的高いと考えられるもの<br><br>(例) 透析器、人工骨、人工呼吸器<br> | 患者への侵襲性が高く、不具合が生じた場合、生命の危険に直結する恐れがあるもの<br><br>(例) ヘルスメカ、人工心臓弁、ステントグラフト<br> |
|              | 薬事法の分類   | 一般医療機器  | 管理医療機器  | 高度管理医療機器  |
| 規制           | 届出   | 第三者認証(注2)   | 法改正で拡充  | 大臣承認(PMDAで審査)   |

(注1) 日米欧追加の5地域が参加する「医療機器規制国際整合化委員会(GHTF)において平成15年12月に合意された医療機器のリスクに応じた4つのクラス分類の考え方を薬事法に取り入れている。

(注2) 厚生労働大臣が基準を定めたものについて大臣の承認を不要とし、あらかじめ厚生労働大臣の登録を受けた民間の第三者認証機関(現在12機関)が基準への適合性を認証する制度。

※厚生労働省「薬事法等の一部を改正する法律について」改正法の参考資料より抜粋  
<http://www.mhlw.go.jp/file/06-Seisakujouhou-11120000-Iyakushokuhinkyoku/0000066816.pdf>

図 3.5-3 医薬品医療機器等法におけるクラス分類

## ■医療機器における品質要件事例

### (1)内視鏡システム —信頼性—

内視鏡を用いた検査では検査中に画像が消えてしまうと、挿入したスコープを一旦取り外して再度挿入することになり、患者に不快な思いをさせてしまう。また、内視鏡下外科手術の場合では、機器が正常に動作するまで手術を中断したり、開腹手術に切り替えたり、後日改めて手術を行うようなケースにもつながってしまう。このように、内視鏡を用いた検査や手術において、「画像消え」はあってはならない事象の筆頭に上げられている。これは”医師は一瞬足りとも術野から視線を外さない”という鉄則を犯すことになってしまうためである。そのため、内視鏡





システムにおいては、民生品では許容範囲とされるような「画像消え」に対しても、異常系を含むあらゆるケースを想定した設計・検証を行うなど、細心の注意を払った対応を行っている。

## (2) 外科用エネルギーデバイス —機能適合性—

外科手術で使われる電気メスや超音波メスといったエネルギーデバイスでは、スイッチを押していないにもかかわらず出力してしまう場合、設定した値より過大な出力になってしまう場合が危険なユースケースとなる。それによって正常部位に火傷を与える、或いは穿孔に至るようなケースが考えられる。一方、実際の超音波メスや、電気メスではその出力の増減で切開の深さ／広さ、凝固の範囲などが大幅に変わってしまう。そのためこのような外科用エネルギーデバイスでは、使用する医師の意図通りに確実に作動するとともに、使用する周辺の環境が変わっても、いつでも同じ結果を出すことが求められている。



## ■まとめ

医療情報連携ネットワークが広がりつつある中、医療・ヘルスケアの環境は様々なステークホルダーが様々なデバイスをつなぎ情報を共有していくものになっていく。そうした医療・ヘルスケア・システムが今後どのように品質を確保していくかが重要な課題になっていくであろう。本節では主に現状の医療機器の環境を述べたが、今後のつながる医療においては、SQuaRE 等も参考に、広範な視点に立った医療向け品質モデルの定義が大切になっていくのではないかと考えている。



## 3.6 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 3.6-1 ビッグデータの例

出典:総務省 <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 3.6-1 に示すデータ品質モデルを規定している。

表 3.6-1 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)  | ○     | ○      |                            |
| 可用性 (Availability)       |       | ○      | システム依存の視点からのデータ品質特性        |
| 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。

## コラム

### IT サービス品質

システム及びソフトウェアの品質に関連したもう1つのトレンドとして、システム及びソフトウェアを様々な活用したITサービスの品質が重要になってきている。ITサービスの品質では、サービスの提供に関わる人の能力、それらの人とサービス提供基盤としてのシステム及びソフトウェアとの相乗効果などが問題になる。そのようなサービスを開発し提供するベンダにとって、またサービス利用者にとって、ITサービスの品質に対する共通の認識、理解を促進するための品質モデルが必要となってきた。このような背景から、現在ISO/IEC 25011として、ITサービスの品質モデルの国際規格化の作業が進められている。

## 3.7 クラウドサービス分野【事例】

### ■クラウドサービスの現状

近年、多岐にわたる分野で M2M (Machine to Machine) や IoT (Internet of Things) などを用いて、社会や企業活動を通じて生成される多種多様なデータを活用し、新たな事業を創出していくことが期待されている。こうした中、システムの初期投資を抑えつつ、短期間でシステム構築や構成変更が容易に行え、企業のグローバル事業拡大や新事業の立ち上げに迅速に対応できるシステム基盤として、クラウドサービスを活用することが一般的になっている。

また、クラウドサービスの裾野の広がり、さまざまなクラウド基盤を相互運用するための技術標準化を踏まえ、企業ユーザーの関心は、「どのクラウドサービスを使うか」から、「どうクラウドサービスを組み合わせるか」に移ってきている。サービスを提供する事業者にとっては、多様化する利用者ニーズに対応した製品・サービスを迅速に提供するとともに、それらの品質確保の重要性がますます高まっている。

「つながる領域」として見たときに、センター設備、ネットワーク、ハードウェア、基本ソフトウェア、アプリケーション、ヘルプデスクなど、様々な資源を必要とするサービスを提供する場合、単一の事業者(企業や組織)でこれらすべてをまかなうことは困難であり、多くの場合、複数の事業者が関与してサービスを提供する形態がとられる(これを垂直方向のつながりと呼ぶ)。

一方、グローバルで高い競争力をもつパートナーとのアライアンスにより、異なる複数のクラウドサービスをシームレス、かつセキュアに接続するクラウド連携をとる形態がある(これを水平方向のつながりと呼ぶ)。

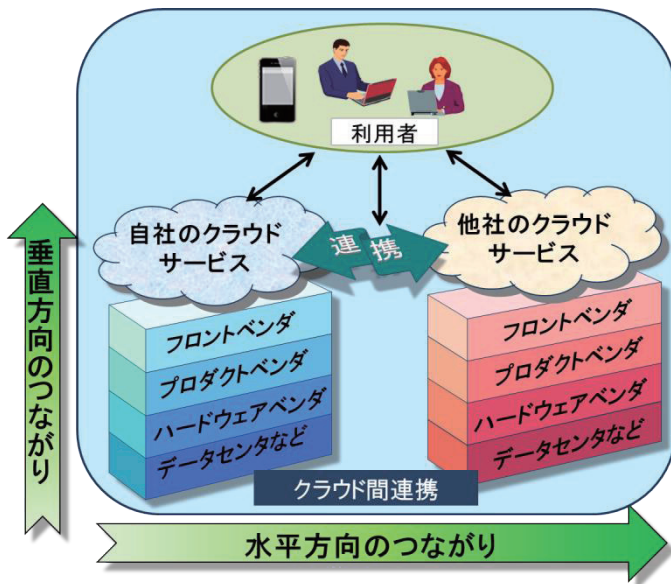


図 3.7-1 つながる領域のイメージ

## ■ サービス品質の向上に向けた取組み

本節では、利用者に対するサービス品質を維持・向上していくために、クラウドサービス提供に関わる事業者の内部で行われている施策のうち、垂直・水平方向の連携を前提とした取組み事例を中心に述べる。

### (1) 利用者とのサービス品質の合意

クラウドサービスのように、システムの運用まで含めた IT サービスを提供するビジネス形態においては、サービス内容について評価指標を設定し、達成すべき品質レベルを利用者に対して合意する SLA<sup>28</sup>の設定が一般的となっている。利用者にとっては、メニューなどで謳われている性能や機能がどの程度の品質で確保されるのか事前を知ることができ、また、求めるサービスの品質とコストを比較して事業者を選定できるようになる。

<sup>28</sup> SLA : Service Level Agreement

サービスレベルアグリーメント / サービスレベル合意書

SLA の普及は、サービスを提供する事業者にとっては、より高い品質レベルのメニューを準備し他事業者との差別化を図るための基準となるが、同時に、高品質なサービス提供にはリスクが伴うため、事業者内部の様々な改善努力が必要になる。

## (2) 垂直方向のつながりに対する取組み

IT サービスを提供する事業者が、組織内部で行うマネジメントプロセスが SLM<sup>29</sup>である。SLM は、要求されるサービスレベルを保証するために、提供事業者が継続的に取り組むべき基本的なマネジメントプロセスである。しかしながら、SLA への関心の高まりと比較して、SLM の重要性への認識は未だ十分とはいえず、体系的・組織的な取り組みは遅れている。

内部目標を設定し、それをマネジメントすることで、不要なコストの削減、安定したサービスの提供が可能となり、その結果として顧客満足度の向上につながる。

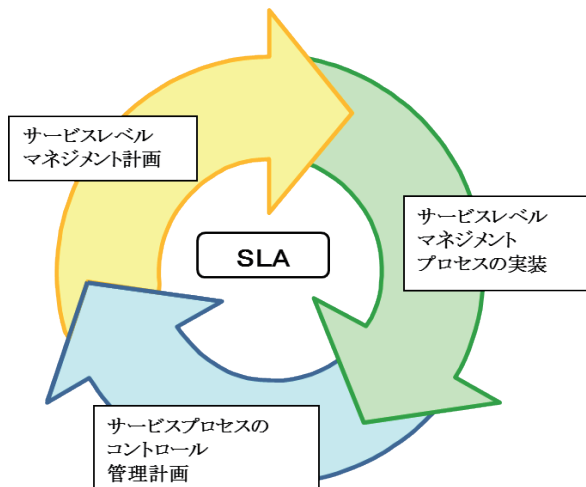


図 3.7-2 SLM プロセスの概要

SLM は継続的で予防的な PDCA サイクルをもった改善活動であり、

<sup>29</sup> SLM : Service Level Management

サービスレベル管理 / サービスレベルマネジメント

SLM 計画の策定時には、SLA を保証するために必要となる詳細な評価指標と定量的な達成目標(これを KPI<sup>30</sup>という)を設定する。

KPI の評価指標の範囲は多岐にわたるため、その選定や評価は難しいが、システムやソフトウェアの品質に関しては、SQuaRE の品質モデルが参考になる。

### (3) 水平方向のつながりに対する取組み

サービス事業を新規に立ち上げる際、全てを一事業者単独で開発するのではなく、コモディティ化が進んでいる領域については、他事業者とパートナー関係を築くことで、開発期間を短縮する取組みも増えている。パートナーが提供する各種サービスを事業の一部として活用することで、サービス事業の拡大を推進し、よりよい品質を提供していくことが可能となる。一方で、このような形態では、パートナーの品質レベルを正確に把握したうえで、たとえ不十分な箇所があってもそれを許容する必要があるなど、自組織での開発に比べ品質管理は大きく制限される。

このため、パートナーのサービスを利用したサービス事業の品質を確保するために、他社サービスの設計概念、標準プロセスなどに関する留意項目や評価に関するノウハウを体系化したガイドラインを整備し、サービスの開発に活用する取組みも行われている。

表 3.7-1 にガイドラインの一例を示す。

<sup>30</sup> KPI : Key Performance Indicators 重要業績評価指標

表 3.7-1 ガイドライン項目の一例

| No. | ガイドライン項目             |                         | 確認事項  |
|-----|----------------------|-------------------------|---|
| 1   | 企業背景                 | パートナー企業                 | パートナー企業の日本での体制・言語   |
|     |                      |                         | システムで取り扱える業務件数（業務量）   |
|     |                      | 可用性設計                   | 利用者に提供するサービスに要求される信頼性とパートナーのサービスの信頼性に乖離                         |
|     |                      | 第三者機関の認証                | ISMS認証、プライバシーマーク制度(Pマーク)などの認証（受けていることが望ましい）                     |
| 2   | 法律関係                 | 関連業法（事業を営む上で必要となる許免許関連） | 電気通信事業者としての届出・登録<br>（この届出がなされていない状況下では、その他の法遵守が蔑ろにされている懸念がある）   |
| 3   | 契約書・SLA              | 可用性                     | サービス時間、サービス稼働率、ディザスタリカバリの有無、システム二重化の有無                          |
| 4   | 体制・会議体               | 緊急体制                    | エスカレーション体制、災害時体制など<br>（通常体制では解決しない時の体制を明確にしておくべきである）            |
| 5   | 運用                   | 定常運用、<br>臨時運用           | スケジューリング、巡回、入在庫、レポート、消耗品管理、開始管理、障害管理、終了管理、セキュリティ対策、秘密情報等取り扱いなど。 |
| 6   | 監視・インシデント<br>対応・変更管理 | 生死監視・<br>障害予兆監視         | ノード監視、プロセス監視などの仕組み、ログ情報から障害予兆が検出できる仕組みの有無など。                    |
| 7   | テスト                  | 負荷試験・<br>レギュラー試験        | 性能限界試験、1:n試験、不正値試験、準不正値試験、手順不正試験の実施有無。                          |

このようなガイドラインを整備することで、採用するパートナーのサービス固有の仕様によらない普遍的な評価と、想定されるリスクの最小化が可能となり、顧客満足の向上につながる。



## ■サービス品質の見える化の取組み

IT サービスの利用者に対して、サービス品質の見える化を促進する取組みも進んでいる。一般財団法人マルチメディア振興センター<sup>31</sup>は、総務省から公表されている指針に対応した「ASP・SaaS の安全・信頼性に係る情報開示認定制度」を2008年から実施しており、さらに2012年から、同じく総務省の指針に対応した制度として、「データセンターの安全・信頼性に係る情報開示認定制度」及び「IaaS・PaaS の安全・信頼性に係る情報開示認定制度」を創設し、これら3つの情報開示認定制度を、「クラウドサービスの安全・信頼性に係る情報開示認定制度」と総称し運営している。これらの制度は、クラウドサービスの利用を考えている企業や地方公共団体などが、事業者やサービスを比較、評価、選択する際に必要な「安全・信頼性の情報開示基準」を満たしていることを第三者機関が認定し公開するもので、サービス品質の全体的なレベル向上につながると期待される。

## ■まとめ

クラウドサービスや、ITシステムの運用まで含めたITサービスを提供するビジネス形態が広く、加速的に普及している。

一方で、サービスを利用するユーザーの顧客満足を十分なレベルで提供するための、国際規格を含む標準化(3.6 内コラム参照)やガイドラインの整備は、まだまだ発展途上である。本節で述べた事例も参考に、顧客が満足する製品・サービスの品質の維持・向上に向けた、サービス事業者の継続的な取組みが必要といえる。

---

<sup>31</sup> <http://www.fmmc.or.jp/>



# 第4章

## 品質向上に向けた 改善のポイント

必要な品質を確保するために事業者はどのように取り組めばよいであろうか。この章では、SQuaRE で規定されたプロセスを中心に、開発現場における改善ポイントについて解説する。

- 
- 4.1 開発・運用プロセスへの組み込み
  - 4.2 品質のトレーサビリティとは
-

## 4.1 開発・運用プロセスへの組み込み

システム及びソフトウェアの開発・運用を進める中で品質向上に向けた改善を行うには、品質の仕様化、実装、制御の3つの側面から取り組むことが重要である。ここでいう仕様化は、顧客要求や組織目標に基づいて品質要求を定義することで、実装は、開発手順を整備し、技法やツールを活用するなどして品質を作り込むことで、制御は、品質の達成度を定量化・可視化して検証・改善することである。

図 4.1-1 に、品質の仕様化、実装、制御という一連の活動と、SQuaRE を構成する国際規格との関連を示す<sup>1</sup>。

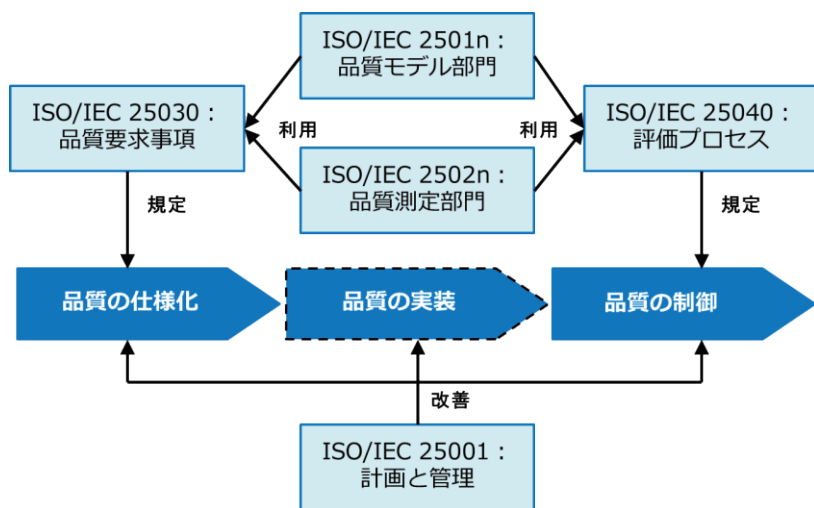


図 4.1-1 品質の仕様化、実装、制御における SQuaRE の活用

以下、SQuaRE の考え方を参考に、システム及びソフトウェアの品質向上に向けた改善を行うための、3つのステップを説明する。

<sup>1</sup> SQuaRE の規格番号と内容は第 2 章参照。

## ■品質の仕様化

システム及びソフトウェアの仕様化を行う際に、その機能を仕様書や設計書の形で明確化することは通常行われるが、品質に対する仕様を漏れなく記述することは必ずしも実践されていない。その結果、実運用のイメージが明確になってくる開発後期に品質に対する認識の齟齬が生じ、後戻りが発生することがままある。まずは開発初期に品質を仕様化することが重要である。

システムを構成するソフトウェアの品質の仕様化では、ソフトウェア自体に求められる内部(静的な)品質とソフトウェアをシステムとして動作した際に求められる外部(動的な)品質を、製品品質モデルで規定している品質(副)特性を参照して仕様化するのがよい。また、システム及びソフトウェアが利用者にも与える影響から見た品質は、利用時の品質モデルで規定している品質(副)特性を参照して仕様化するのがよい。さらに、システム及びソフトウェアの処理対象となるデータの品質は、表 3.6-1 のデータ品質モデルで規定している品質特性を参照して仕様化するのがよい。

この際、図 4.1-2 に示すように、異なる品質モデルの品質(副)特性の関

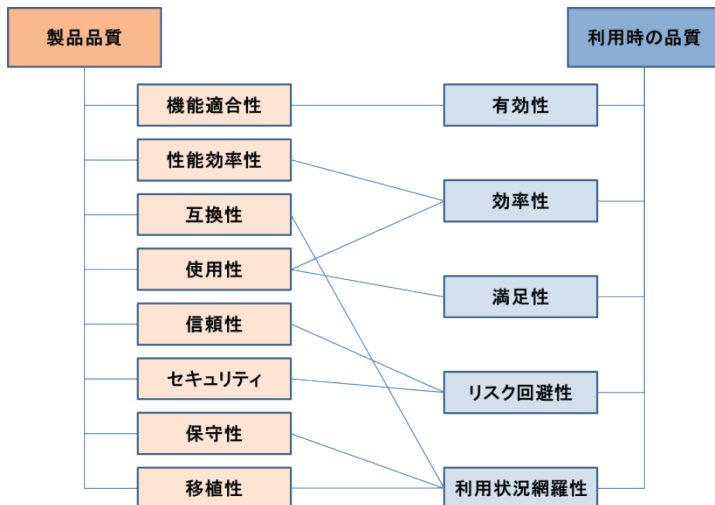


図 4.1-2 製品品質と利用時の品質の特性間の関係(例)

係性を考慮して整合を図ることが重要である。例えば、利用時の品質モデルで、システム及びソフトウェアの利用者がそれを用いることで作業の効率やどの程度向上するかを扱う「効率性」に対するニーズを持っているならば、製品品質モデルでは、システム及びソフトウェアの動作の効率を扱う「性能効率性」や操作のしやすさを扱う「使用性」の観点から要求事項を定義することとなる。

品質の仕様化では、まずはシステム及びソフトウェアの利用に対するニーズ、またはその機能や品質に対するニーズを持っているステークホルダを識別する。ステークホルダには、システム及びソフトウェアの開発に関与する開発者と発注者だけでなく、直接、間接の利用者が含まれる。それらを共通の属性を有するグループに分類する。これらステークホルダの個々のグループが、システム及びソフトウェアに求める品質特性とそれらの重要度を品質モデルに照らして特定し、重視する特性に対する具体的な要望を品質要求として仕様化する。この際、相反する品質要求が識別された場合、あるいは費用、期間などの制約からすべての品質要求に応えることが困難な場合には、ステークホルダ間で協議を行い、調整を行

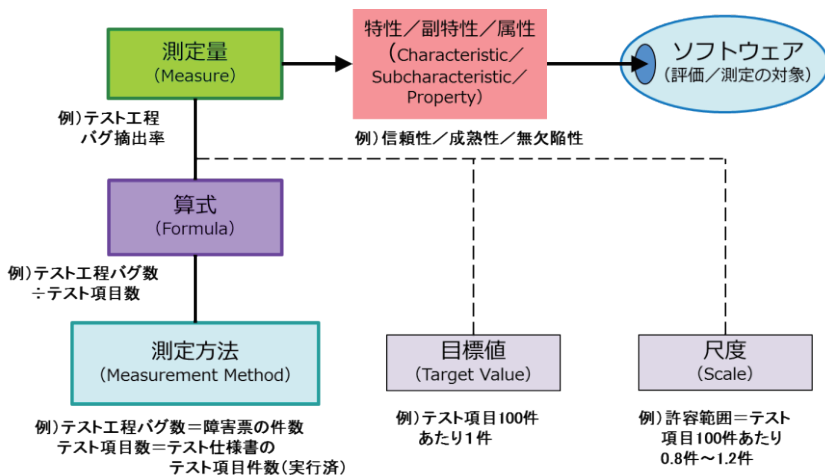


図 4.1-3 品質要求事項定量化の考え方

う。

仕様化された個々の品質要求について、その充足の度合いをどのように測定し、評価するのかを明確にする。測定と評価は、図 4.1-3 に示すように、算式と測定方法によって定義された測定量(これを品質測定量と呼ぶ)、及び算出された測定値の良否判定を行うための目標値や尺度を設定する。目標値や尺度の設定は、過去の実績データの収集・分析によるのが望ましいが、利用可能なデータが欠如している場合には、公開されているデータ白書や事例を参考に設定し、データの蓄積に伴って見直しを行う。

## ■品質の実装

品質の仕様化で、重視する品質特性と具体的な品質要求を定義したなら、それらをいかに実現するか、プロセスデザインの視点から検討することが必要になる。

重視する品質特性と具体的な品質要求が定義されたなら、システム及びソフトウェアのライフサイクルプロセス<sup>2</sup>に照らして、重視すべきプロセス、アクティビティ、タスクを特定する。例えば、「機能完全性」(求められる機能の充足)については、構成管理のプロセスを確実に実施し、要求仕様に定義された機能が、開発プロセスを通して漏れなく実装できていることを追跡できるようにする必要がある。また、「成熟性」(成果物に欠陥を含まない)については、レビュー、検証、妥当性確認、適格性確認などのプロセスを確実に実施し、それらを通じての瑕疵やバグの収束度合いなどを管理する必要がある。

個々の品質特性のうち、利用時の品質やセキュリティ/セーフティに関しての実装に関する留意事項については、本書の第 3 章を参照するとよ

---

<sup>2</sup> システム及びソフトウェアを開発するプロセスは、ISO/IEC 12207:ソフトウェアライフサイクルプロセス(主に、テクニカルプロセスとソフトウェア固有プロセス)、ISO/IEC 15528:システムライフサイクルプロセス(主に、テクニカルプロセス)に規定されている。これらの国際規格では、要求分析、設計、実装、結合、検証、妥当性確認などのプロセスごとに、プロセスの目的、成果、アクティビティ及びタスクを規定している。

い。

## ■品質の制御

システム及びソフトウェアの品質の制御は、品質要求定義として仕様化され、開発・運用された成果物及び製品に実装された品質を評価し、評価結果に基づいて改善することで行われる(図 4.1-4 を参照)。ここでは、SQuaRE の品質評価プロセスにそって、その活動を説明する。

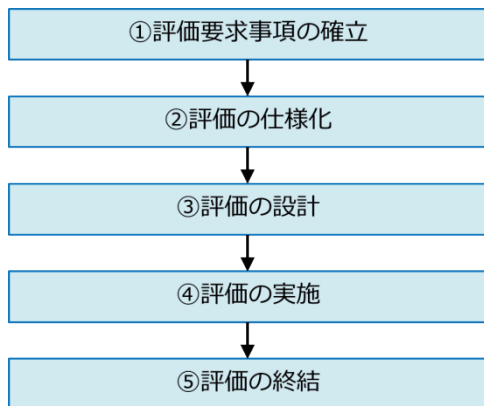


図 4.1-4 SQuaRE の品質評価プロセス(JIS X 25040)

### ① 評価要求事項の確立

評価の目的には、取得者による受入れ等の際の評価や、独立の立場からの第三者評価もあるが、ここでは開発者による評価活動を説明する。その際の基本的な目的は、“開発した成果物が要求を満足しているかを確認すること”である。

評価の目的に照らして、個々の品質要求の充足を評価する対象を特定する。最終的な製品に加え、開発の過程で作成される中間成果物(設計書、ソースコードなど)やシステム及びソフトウェアを構成するコンポーネント(サブシステム、モジュールなど)も評価対象になる。システム及びソフトウェアの構成要素の評価では、成熟性(欠陥がないこと)など共通の



に考慮すべき品質要求もあれば、セキュリティの実装など特定の構成要素で考慮すべき品質要求もある。このような点を加味して、ライフサイクルの中で開発・検証・運用される評価対象ごとに、「品質の仕様化」で定義された品質要求が、どの構成要素に関連するかを特定する。また、品質の重要度と工数・コストの制約に応じて、どの程度の厳密さで評価を実施するかを検討する。

## ② 評価の仕様化

仕様化された品質要求を定量的に評価するための品質測定量を選定し、個々の品質測定量に対する評価基準を設定する。評価目的、評価対象、評価の厳密さに照らして、「品質の仕様化」の際に設定した品質測定量と評価基準が各評価対象に適用できるかを検討する。

また、個々の品質測定量による良否判定や、重要度による重み付けを加えたうえで、工程終了、出荷などの可否判定に用いる総合的な評価基準を設定する。

## ③ 評価の設計

品質評価の計画を立案する<sup>3</sup>。

図 4.1-5 のようなライフサイクルを参考にデータの収集と分析、分析結果に基づく評価と判定などのスケジュールを策定する。具体的な評価方法の確立や、評価作業の効率化などに役立つ技法やツールの利用に関しては、SQuBOK<sup>4</sup>、SWEBOK<sup>5</sup>などの知識体系が参考になる。

<sup>3</sup> ISO/IEC 25001 に品質評価計画テンプレートが定義されているので参考にするとよい。

<sup>4</sup> SQuBOK: Software Quality Body Of Knowledge ソフトウェア品質知識体系

<sup>5</sup> SWEBOK: SoftWare Engineering Body Of Knowledge ソフトウェアエンジニアリング基礎知識体系

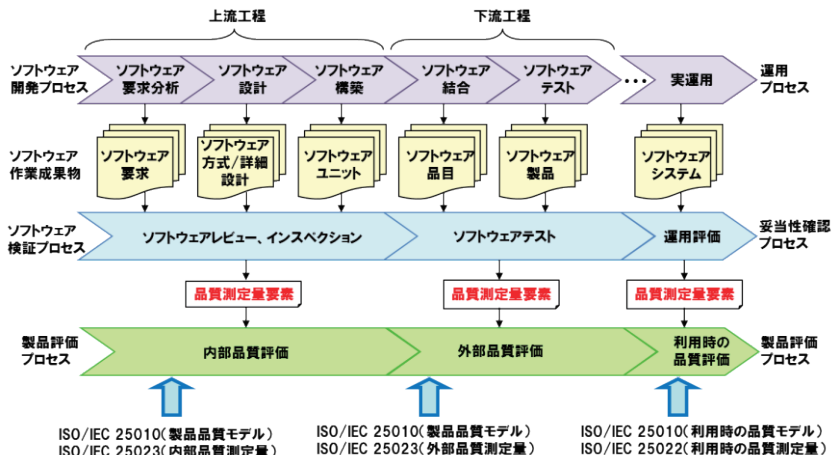


図 4.1-5 ソフトウェアのライフサイクルと品質評価

#### ④ 評価の実施

品質評価計画に基づいて評価を実施する。ライフサイクルの各段階で、品質評価に用いる品質測定量の算出に必要なデータを収集し、測定値を算出する。しかし、測定値単独では、その値の良否判定はできない。図 4.1-6 に示すように、「評価の仕様化」で設定した評価基準を用いて、目標

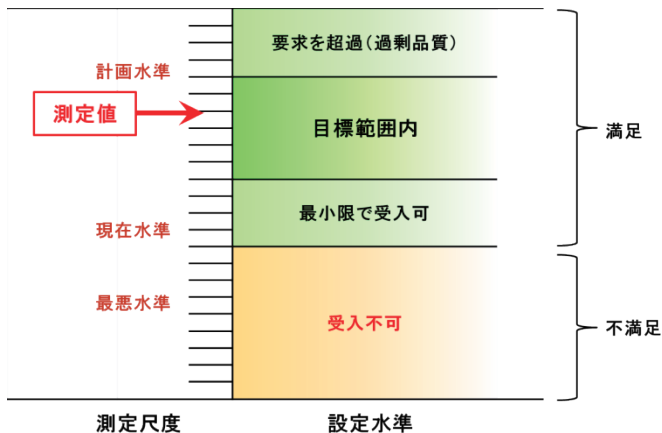


図 4.1-6 品質測定量の評価イメージ(JIS X 0133-1)

とする値との乖離の大きさを見たり、優／良／可などの水準に合わせて設定された値の幅（尺度：Scale）に測定値を照らし合わせたりすることで良否判定が可能になる。更に、品質（副）特性別の評価、品質全体の評価、品質及びその他管理ファクタを考慮して総合的に評価を行うことが重要である。

#### ⑤ 評価の終結

一連の計画や結果は評価報告書に記録し第三者の確認や、今後の改善活動に利用するとよい。

各種評価において、測定値が評価基準に満たない場合には、開発されたシステム及びソフトウェアの修正などのフィードバック処置と再評価を計画し、実施する。また、評価の方法に見直しの余地がある場合には、報告書に課題や改善方策を記載して、測定方法や評価基準の見直しを行う。

## コラム

### アジャイル開発や小規模開発への応用

SQuaRE で述べられている品質プロセスの考え方は、基本となる筋道を整理したものなので、特定の開発スタイルを前提としていない。したがってアジャイル開発や小規模開発等にも応用できる。その際には、システムの再利用性、他のシステムとの「つながり」の度合、システムの利用期間等を考慮して、メリハリのある重点がどこにあるかに着目した取組みを行うとよい。その際、品質プロセスの管理的な側面は軽い取扱い(メモ書きによる確認など)で済ませられる場合がある。

## 4.2 品質のトレーサビリティとは

一般に、システム及びソフトウェアの開発におけるトレーサビリティ(追跡可能性)とは、開発の過程で、設計書、ソースコード、テスト結果、等の成果物に、要求項目がどのように、展開されているか追跡できること、あるいはその逆方向に辿れるようになっていいることをいう<sup>6</sup>。これを品質の観点で管理することを、本書では、「品質のトレーサビリティ管理」と呼ぶ。

不具合が生じたときの生命、財産、環境などへの影響が大きいシステム及びソフトウェアでは、開発及び運用の過程で品質のトレーサビリティ管理を実施し、要求された品質が求められた水準で漏れなく実現されているかを可視化して、リスクを制御することが極めて重要になる。

システム及びソフトウェアの品質のトレーサビリティは、4.1 に述べた品質の仕様化、実装、制御を、開発及び運用の中で適時かつ反復的に実施していくことがベースになる。それによって、開発及び運用の過程で、要求された品質が品質モデルを用いて多角的観点から仕様化され、品質測定量と設定した基準を用いてどの程度の水準で実装されているかが定量化され、適時に対処することが可能になる。

また、開発者の説明責任の観点からも、品質の可視化、トレーサビリティの確保を実現する必要がある。継続的な品質評価並びに評価結果の保管に加え、第三者視点からの客観的な品質評価を可能とする仕組み作りが求められており、開発部門から独立した品質保証部門によるレビューやテストの実施とその結果に基づく工程審査や出荷審査の実施、独立したテスト機関の活用、SQuaRE に基づくシステム及びソフトウェア品質の認証機関による審査の受診など、その実施形態、あるいはシステム及びソフトウェアのクリティカリティのレベルに応じて実施する。

この際、開発者側の自助努力だけに頼るのには限界がある。発注者、

<sup>6</sup> ISO 9000 では、「考慮の対象となっているものの履歴、適用または所在を追跡できること」と定義されている。

購入者側でも、開発を委託又は購入するシステム及びソフトウェアの品質が及ぼす影響を理解し、然るべき水準の品質を実装し、維持するのに要する費用を受容することが求められる。

開発、運用の各段階で実施した品質評価の結果、システム及びソフトウェアの品質上の欠陥が判明した場合には、評価対象となったシステム及びソフトウェアの修正を実施するのに加え、品質向上の仕組み自体の改善を図ることも重要である。

SQuaRE では、品質要求定義及び品質評価のプロセスを実践する上での原理原則を規定しているが、そこに示された原理原則を効果的に実践するには、それらに関連する技術、ノウハウ、データなどの知識や経験を組織的に蓄積して標準化し、プロジェクトへの適用と移転を行い、プロジェクトでの適用結果に基づいて改善することが不可欠である。この技術管理の活動は、ISO/IEC 25001 に規定されている。組織とプロジェクトが連携して遂行する技術管理活動の流れを図 4.2-1 に示す。

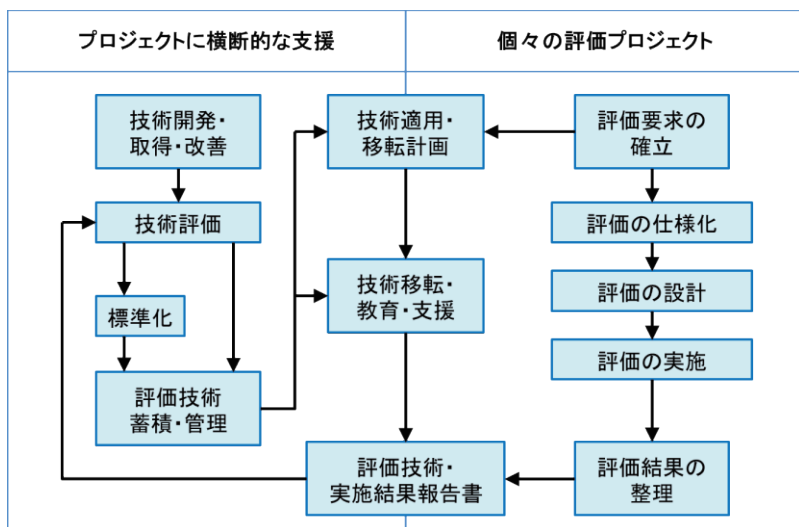


図 4.2-1 SQuaRE の技術管理のフロー (JIS X 0133-1 の解説)

## コラム

### 品質を戦略的に考える

「品質を確保すること」はコストがかかるというのは、決して間違っていない。一般的に、品質の目標レベルを高く設定すれば、必要な開発コストは増加する。しかし、品質が不十分な製品が、出荷後に問題を出してしまうと、その対策コストや、企業イメージの低下によるマイナス要因などを含めると、開発時の何倍ものコストが発生する、という研究結果もある。この点も踏まえて、適切に品質コストをコントロールする必要がある。

本書では、利用者の要望が多様化し、関連するステークホルダが増えている昨今の状況を踏まえ、事業者に対して品質要求を多様な観点で見直すことを推奨している。しかし、これは、必ずしもコストの増加につながることを意味していない。製品の特徴、会社での位置づけ、方針などに合わせて、あえて品質を低いレベルに設定するという選択もある。つまり、事業戦略に基づいて、対象とする利用者や、利用場面を想定したうえで、品質の種類や目標レベルとの間に不一致がないように適切に設定することが重要である。





# 第5章

## 利用者の安全・安心につながる品質説明

製品利用者の安全・安心のためには、製品の品質が十分であることや、製品が想定している環境や使い方を、利用者にはっきりと説明することが重要である。また、第三者によってソフトウェアの品質を確認する認証制度について事例を交えて紹介する。

- 
- 5.1 品質説明と第三者評価
  - 5.2 PSQ 認証制度 **事例**
-

## 5.1 品質説明と第三者評価

近年、システム及びソフトウェアの利用の形態が増々多様化する中で、その欠陥に起因した不具合が、利用者の不利益や社会的な混乱を引き起こす事例が後を絶たず、訴訟に発展するケースも見受けられる。このような背景のもと、事業者の品質に対する説明責任(アカウントビリティ)への関心が高まっている。事業者には、製品の品質に関する取組みの正当性や十分性を、利用者を含む幅広いステークホルダに対して説明する努力が求められている(品質説明)。

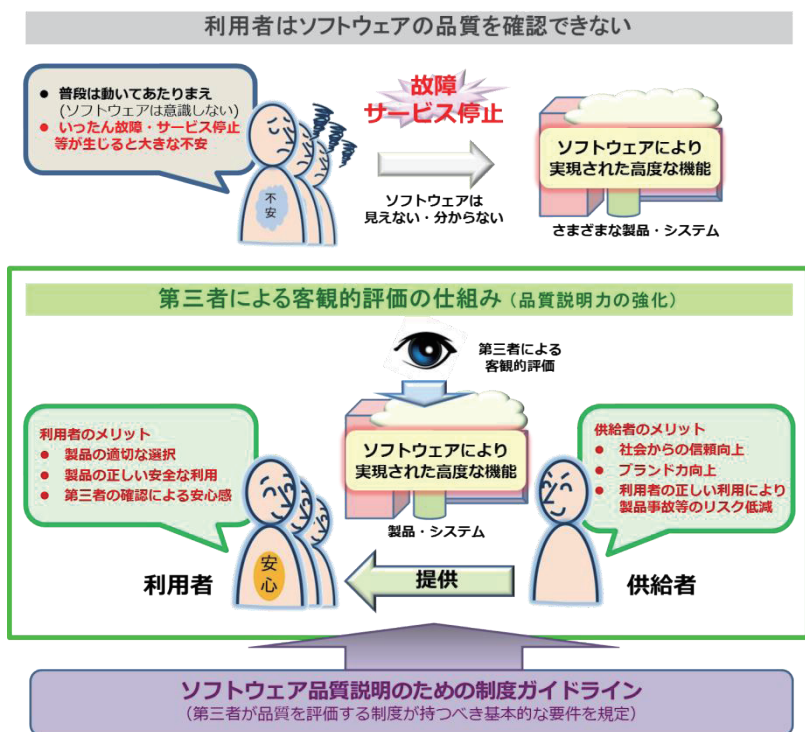


図 5.1-1 ソフトウェア品質説明

出荷後や運用中に何らかの不具合が発生した場合の品質説明だけでなく、製品の購入やサービスの利用に先だって、利用者に安心して使ってもらうための事前の情報提供が重視される傾向がある。

一方で、利用者の側にも、自分の利用目的や利用場面を考えたうえで、どんな機能やどのレベルの品質を必要としているかを整理し、事業者が提供する情報を収集・理解し、適切なものを選択し、正しく利用する姿勢が求められる。

IPA/SEC では、このような品質説明の取組みを促進するための環境整備の一環として、事業者の品質説明が適切であることを、第三者の専門家が客観的に評価し、利用者に向けて分かりやすく公開する仕組みを構築するための指針をまとめた「製品・システムにおけるソフトウェアの信頼性・安全性等に関する品質説明力強化のための制度構築ガイドライン」（以降、制度ガイドライン）を作成し公開している。

「制度ガイドライン」では、事業者と利用者から独立した第三者が評価することによる公正性の確保や、制度の運用体制（制度フレームワーク）のあり方、さらには、制度設計において順守すべき43項目を規定している。

### IPA「ソフトウェア品質説明のための制度ガイドライン」

#### ■ 目的

- 技術の進化が速く、国際標準等の整備が十分でない分野において、客観性のある品質基準や**第三者評価の仕組みを整備**することで、**高品質な製品・サービスの普及を促進し、利用者の安全・安心につなげる**
- 何らかの障害発生時に、**事業者が品質に対する説明責任**を果たせる環境を整備することで、事業者の参入障壁を緩和する

#### ■ 対象利用者

- 新たな認証の仕組みを構築したいというニーズを持ち、制度の企画、構築及び運用に責任を持つ組織・団体

#### ■ 内容

- **制度が公正性を持ちかつ継続的に運用されるための制度設計のポイントを解説**
- **制度責任主体の順守事項や、制度規定文書に記載すべき事項を規定（分野に依存しない43項目）**



平成25年6月公開

また、「制度ガイドライン」では、品質を評価するための審査基準の策定方法として、「国際標準化機構(ISO)、国際電気標準会議(IEC)等の定めた国際標準や、日本工業規格(JIS)等の国家標準に依拠して策定されることが望ましい」としている。

## 制度ガイドラインにおける制度フレームワーク

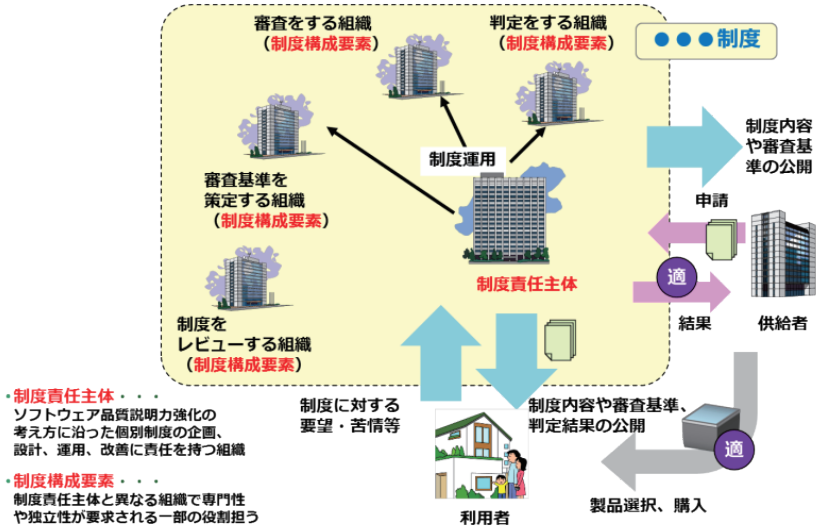


図 5.1-2 制度ガイドラインにおける制度フレームワーク

## 5.2 PSQ 認証制度【事例】

ソフトウェア製品の品質を、国際規格 SQuaRE に基づいて第三者が評価して認証する「PSQ 認証」という制度がある。

この PSQ 認証は、国際規格 ISO/IEC 25051:2006 (JIS X 25051:2011) に準拠した第三者適合性評価に基づく認証制度として、一般社団法人コンピュータソフトウェア協会 (CSAJ)<sup>1</sup> が 2013 年 6 月より運営を開始した。この規格は SQuaRE シリーズの中で ISO/IEC 25010 が規定している品質モデルを基本とした拡張規格である (第 2 章 2.2 参照)。



ISO/IEC 25051:2014

PSQ 認証は、公正性を担保するため、先に説明した「制度ガイドライン」に則り制度化されている。さらに、2015 年 4 月から、ISO/IEC 25051:2014 に準拠した新評価基準により、対象とするソフトウェア製品を、従来のパッケージソフトウェアに加え、クラウド環境で提供される会計システム等のアプリケーションサービスも含むよう拡大している。

### PSQ 認証制度の目的

- ・ソフトウェア製品に関する利用者や市場への品質説明力強化
- ・国際市場における日本製品の品質に関する正当な評価の確立
- ・利用者の潜在的リスクの軽減
- ・ソフトウェア製品の本質的な品質向上

ソフトウェア製品はさまざまな分野での使用が増加しており、その役割は非常に大きくなっている。しかし、それを購入しようとする利用者は事業者が提供する製品情報(カタログやWeb等)でしか性能判断を行うことが

<sup>1</sup> CSAJ: Computer Software Association of Japan

できない。この規格は利用者が安心してソフトウェア製品を選択できることを求めている。すなわち、製品説明と利用者用文書(マニュアル等)、ソフトウェアの試験文書が一致していること、整合性がとれていることを求めている。

具体的には、ISO/IEC 25051 に基づき、「製品説明」「利用者用文書」「試験文書」とソフトウェアが実装する機能で評価する(図 5.2-1)。評価のために評価機関が実際の環境下でソフトウェアを稼働して検証したり、開発マネジメントの観点から間接的に品質を評価したりするものではない。この点が本制度の特徴である。言い換えれば、提供側における品質基準や開発プロセスではなく、できあがった製品に対する直接的な品質を見るものであり、あくまでソフトウェアの購入予定者、利用予定者にとって、要求した通りの機能が正しく提供されているか、また、選定に必要な情報が十分に提供されているかを評価する。

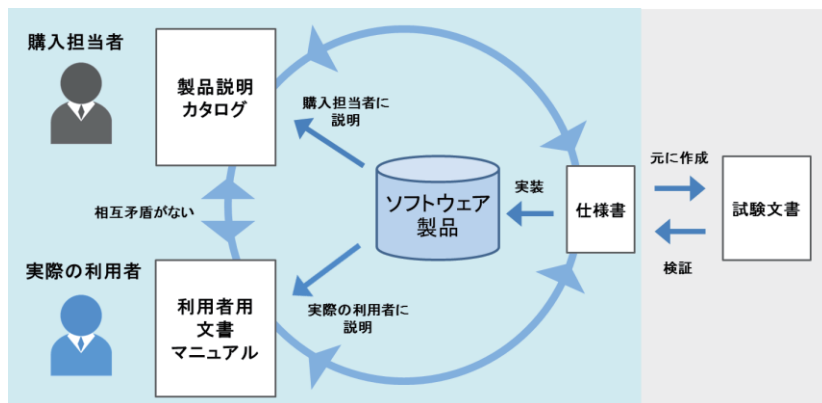


図 5.2-1 PSQ 認証での文書間の相関関係(PSQ 認証制度資料より引用)

#### ■製品説明の評価基準

- ・入手／参照のしやすさ、利用のしやすさ
- ・利用者の要求に対するソフトウェアの整合性を判断できる情報を含んでいるか

- ・製品の特徴、機能、性能、制約などの情報が明確かつ正確に記載されているか

#### ■利用者用文書の評価基準

- ・利用する上で必要な情報が正しく記載されているか
- ・製品説明と矛盾や不一致がないか
- ・想定している利用者にとって理解しやすいか

#### ■試験文書の評価基準

- ・製品説明や利用者用文書に記載されている製品の機能が正しく実装されていることが、試験によって確認されているか
- ・製品の性能や使いやすさなどが試験によって確認されているか
- ・試験を実施した結果や合否判定が具体的かつ明確に記載されているか

これまでに認証取得に取り組んだ企業からは、品質評価の基準を国際標準に合わせて具体化・整理することによって、自社開発プロセスの標準化につながるメリットもあった、との意見が寄せられているとのことである。

## 海外の品質認証制度

「PSQ 認証」と同様に、フランスや韓国でも国際規格 SQuaRE に基づくソフトウェア品質認証制度が運用されている(いずれの制度も 10 年以上の実績がある)。

どちらの制度も、SQuaRE の品質モデルに照らして、製品説明、利用者用文書(マニュアルなど)、試験文書などに、求められる品質をどのように記述し、それをどのように試験し確認しているかを見ている。

各々の認証制度が対象とするシステム及びソフトウェアには違いがあり、韓国の Good Software 認証制度がより広範である。韓国では中央省庁が関与して制度を運用していて、官公系の調達案件では、認証取得製品が優遇されている。また、ソフトウェアの欠陥に起因するトラブルによって発生する損害に対する保険制度があり、認証を取得することで保険料率が軽減される仕組みもある。

「PSQ 認証制度」を運用する CSAJ では、これらの制度を運営する国々と、国際相互承認を目指した情報交換を進めている。



---

# S Q u a R E品質モデル活用 リファレンス編

---

# ◆目次

## 第1章 SQuaRE とは

---

..... P104

## 第2章 SQuaRE での品質特性、測定の考え方

---

..... P112

## 第3章 SQuaRE の構成

---

..... P122

## 第4章 特性項目表の凡例

---

..... P126

## 第5章 利用時の品質 品質特性表

---

..... P130

## 第6章 製品品質 品質特性表

---

..... P150

## 第7章 データ品質 品質特性表

---

..... P198

## 參考資料

---

.....P209



# 第1章

## SQuaRE とは

SQuaRE とは、Systems and software Quality Requirements and Evaluation の略称で、システムとソフトウェアに関する品質定義と品質評価視点の枠組みを定める国際規格 ISO/IEC 25000 シリーズの呼称である<sup>1</sup>。本書は、SQuaRE 品質モデル活用のための入門的なリファレンスを提供する<sup>2</sup>。

この規格のシリーズは、1999 年の国際会議 (ISO/IEC<sup>3</sup> の下部組織である JTC 1/SC 7<sup>4</sup> の WG6 の金沢での会議) で提案されたプロジェクトにより開発された<sup>5</sup>。それ以前からあった古い規格 ISO/IEC 9126、ISO/IEC 14598 のシリーズを統合・再編・強化する形で作られた。

さて、システム／ソフトウェアの品質を課題として取り上げるときには、製品の開発に着目すると、図 1-1 に示すような流れとなる。

---

<sup>1</sup> 本書では、SQuaRE の規格番号として、国際規格としての ISO/IEC で始まる番号表示を主として用いる。JIS (日本工業規格) としての番号も数字の部分は基本的には同じで、詳しくは「第 3 章 SQuaRE の構成」に示した。

<sup>2</sup> ここでは、分かりやすさ、親しみやすさを主眼として紹介するので、厳密な表現とは異なることがある。規格の詳細や応用については、章末に示した「主な役に立つ参考文献」や巻末の文献リストにある文献等を参照。

<sup>3</sup> ISO: International Organization for Standardization、国際標準化機構

IEC: International Electrotechnical Commission、国際電気標準会議

<sup>4</sup> JTC 1: Joint Technical Committee 1、第 1 合同技術委員会

SC 7: Standardization Subcommittee 7、第 7 専門委員会 (ソフトウェア及びシステム技術)

<sup>5</sup> 正式には、2000 年に行われたマドリッド会議でプロジェクトが発足承認された。

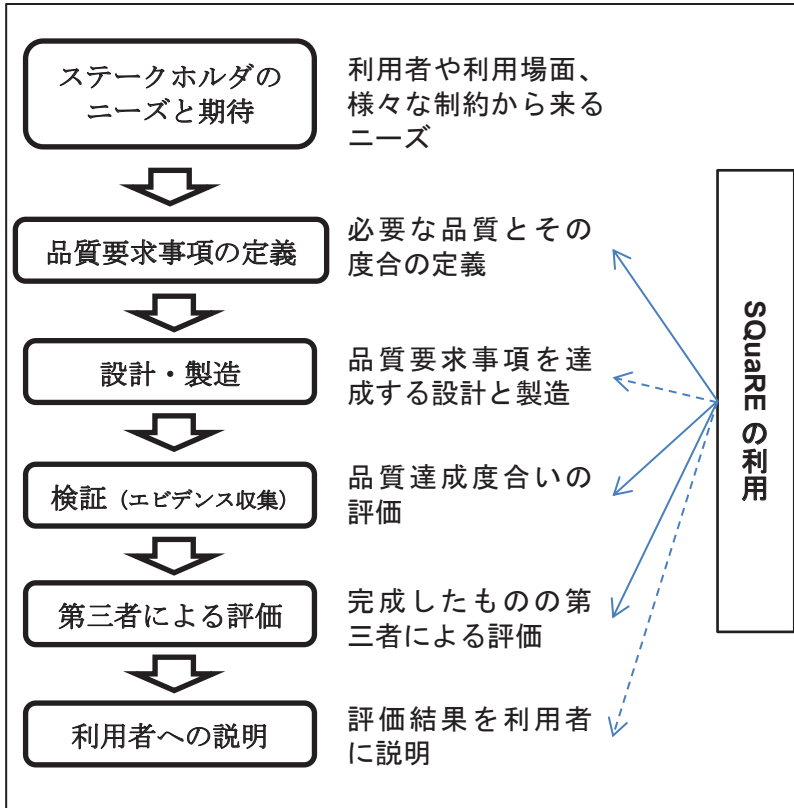


図 1-1 品質取扱いの大きな流れ  
(図は、第三者評価を想定する場合の流れ)

本書は、品質の取扱いで必要となる種々の課題を広範囲に対象としたものではなく、品質評価の基本となる品質モデルと品質測定にフォーカスした解説を行う。より広く見ると、組織的な取組みとしては、より大きな視点での品質管理や品質保証の仕組みなども考える必要がある。また実務面では、品質要求の獲得技術や、認証取得のための客観的な評価方法、品質を高めるためのツール活用といった観点も重要となる。これらの他の関連する技術や手法などについては、プロジェクトの状況に応じて工夫したり、他の文献を参照したりして深掘りする必要がある。

ところで、一般にはシステム／ソフトウェアの品質とは、どのようなものだと感じられているだろう。コンピュータプログラムの開発者やテストの従事

者の中には、品質とはバグのないこと、機能仕様のテストを無事通過して出荷できることだと思っている人々もいる。他方では、利用者の立場から、ソフトウェアの品質とは、そのシステムがいかに素晴らしい価値を自分たちに届けてくれるかだと思っている人々もいる。SQaRE では、そうした観点の拡がりカバーした偏りない視点を提供している。

規格では、ソフトウェア品質を次のように定義している。

「明示された状況下で使用されたとき、  
明示的ニーズおよび暗黙のニーズを  
ソフトウェア製品が満足させる度合い」

また、品質の良し悪しは最終的には利用者のニーズに照らして判断されると考えている。

そして、品質を表す特徴には、システム／ソフトウェアの品質要求定義や評価を行う対象に応じて、「製品品質」と「利用時の品質」、また幅広く考えると「データ品質」もあると考えている。この全体の様子を図 1-2 に示し、次のパラグラフでそれぞれの考え方を説明する。



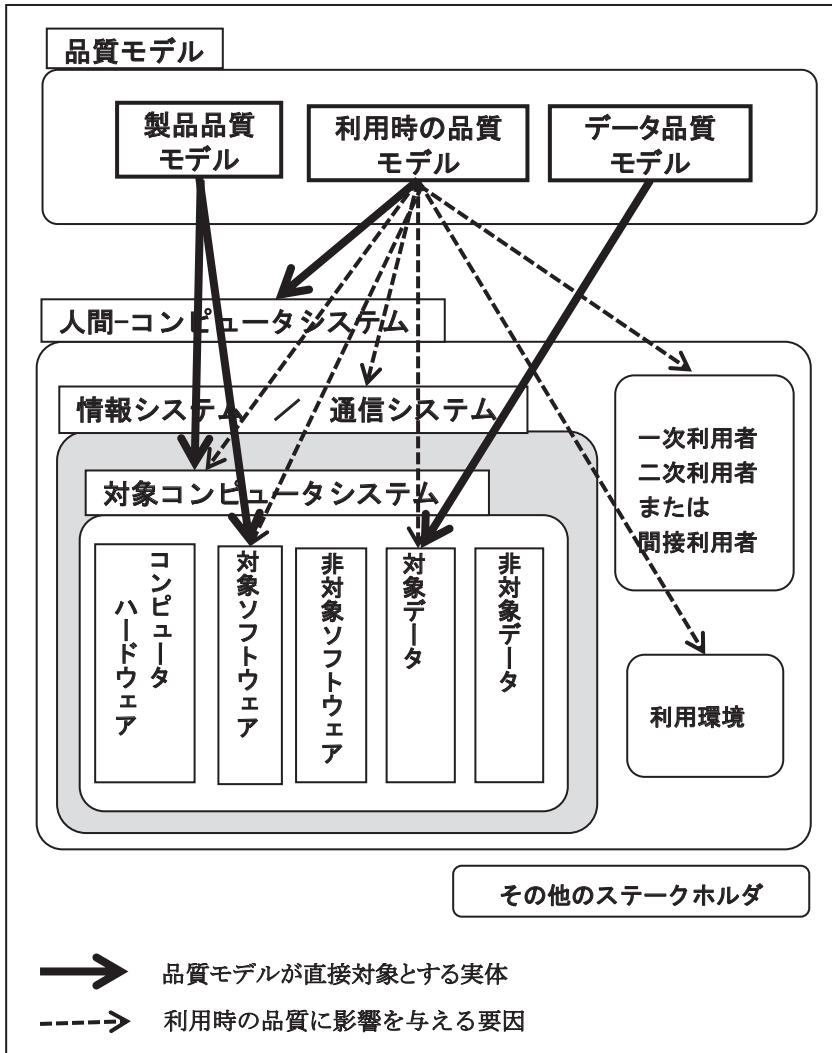


図 1-2 SQaRE 品質モデルの対象  
(JIS X 25010:2013 から簡略化して作成)

製品品質とは、文字通りソフトウェア製品（計算機プログラム、手続き、関連文書とデータを含めた一つのまとまり）／システム製品の品質である。システム／ソフトウェアは、ニーズにもとづくシステム要求・仕様を反映して開発されるが、開発された製品がしっかりした機能や性能を持つものであり、信頼して使えるものであり、また多くの場面で使えるものであることが求められる。そうした要求・仕様のあり方と開発のあり方とから見て、ニーズ

に対応できる品質をどのような面から考えるべきかという観点を示している。明らかな欠陥がないといった消極的な観点だけでなく、より良い品質を提供できているかどうかも問題としている。

**利用時の品質**とは、広い意味での利用者に対するシステムの働きの影響から品質をみる視点である。利用者から見た品質の視点であり、ビジネス上・生活上の自分のニーズに照らして、IT システムが効果的なシステムであり、安全に安心して利用でき、総合的にみて満足できるもの、場合によっては快適に享受できるものであることを評価する品質の観点である。製品品質が製品の側から品質課題をみているのに対して、ここでは利用者への影響、利用者の満足の側から課題をとらえている。また、「広い意味での利用者」という言い方をするのは、例えば自動改札機のシステムトラブルがあった場合、改札の現場では、担当者が保守作業をしなければならないが、めったにない事故への対応、非常に多くの設置場所への対応などで、迅速かつ有効なトラブル対処が可能なシステムとなっているかどうか問われる。その際の保守者は、「広い意味での利用者」に含まれる。また、もっと広く見る時は、自動改札機の故障によって不便を受ける改札口利用者も場合により利用者を含めて考えるということである。

**データ品質**とは、主としてコンピュータシステムが取り扱うデータや情報の品質を取り上げたものである。システムを利用者が利用する際には、プログラムや装置のほか、データや情報が一体となって利用者に価値を提供する。そのデータ／情報の品質を問題としている。

それぞれの品質モデルでは、品質についてのより詳しく区分した観点として品質特性という考え方を導入する。そして、こうした品質特性の状況の評価するには、まず品質要求と関連した評価基準を設定し、それに沿った測定が必要だというように考える。それぞれの品質特性は特定の分野に依存しない品質の観点を定義したものであるため、その良し悪しを直接測れるとは限らない。むしろ、測定できる品質に関する特徴が複数あることが多くある。それらの複数の特徴の測定結果を用い、設定してある評価基準に基づいて、本来システム／ソフトウェアはどうあるべきかの総合的な評価(アセスメントともいう)を行って初めて元の品質特性の評価に結び付く。こうした評価活動は多少複雑な観点を含むため、整理された評価視点や準備活動、データ蓄積・整理活動などが求められるケースも多々ある。

本書では、このような品質特性、そのもう一段の詳細化である副特性、そして品質の測定・評価について、SQuaRE がどのように標準化をしているか、それらをどのように応用すればよいかを見ていく。

### <主な役に立つ参考文献>

- ・情報処理学会誌「情報処理」Vol.55 No.1 (2014年1月号)p.15
- ・東基衛編「ソフトウェア品質評価ガイドブック」日本規格協会(1994刊)
- ・経済産業省 ソフトウェアメトリクス高度化プロジェクト プロダクト品質メトリクスWG編「システム／ソフトウェア製品の品質要求定義と品質評価のためのメトリクスに関する調査報告書」(2011年3月)



## 第2章

# SQuaRE での品質特性、 測定のお考え方

SQuaRE では、品質上の特徴を見る際に、品質要求定義や評価を行う対象に応じて「製品品質」、「利用時の品質」という、2 つの品質モデルを用いる<sup>1</sup>。またこれらとは別にデータの品質にも注目するという考え方（データ品質モデル）が導入されている<sup>2</sup>。これらの品質モデルと品質要求事項の関係を図 2-1 に示す。

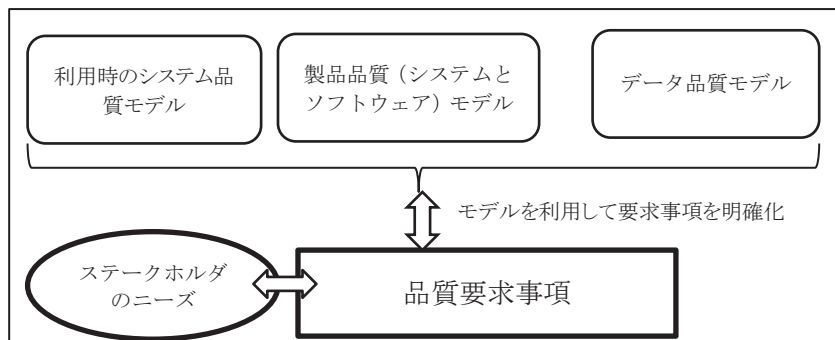


図 2-1 品質モデルの区分と関係

各品質モデルには、品質を表す特徴を取り上げる枠組みとして、品質特性が定義されている。この品質特性は複数の副特性を含み、さらに副特性は、場合によって、より下位の品質特徴を含むとされる（図 2-2）。最下位の品質特徴は属性とも呼ばれる。

図 2-3 から図 2-5 に、それぞれ、「利用時の品質」、「製品品質」、「データ品質」の特性・副特性の一覧を示す。これらの品質特性は、歴史的には、いろいろな研究者や実務者が整理したり、新しい視点を導入したりした結果を受けて、現時点での国際的な知見をまとめたものである。現在のまとめ方の特色としては、利用時の品質が大きく取り上げられたこと、またインターネットなどの下での互換性の視点や、セキュリティの視点が強化されたことなどが挙げられる。

<sup>1</sup> 旧規格の ISO/IEC 9126 では、品質特性を外部特性と内部特性に大きく分類していた。現在の SQuaRE ではそれに類似した考え方として、製品品質に関わる測定において、内部測定量、外部測定量という視点をを用いる。内部測定量はソフトウェアを稼働させないでソフトウェアの記述などから測定できる量、外部測定量はソフトウェアを稼働させてシステムの運用状態で測定する量である。

<sup>2</sup> さらにシステム/ソフトウェアが提供するサービス品質についても SQuaRE の対象とする方向で審議中である。

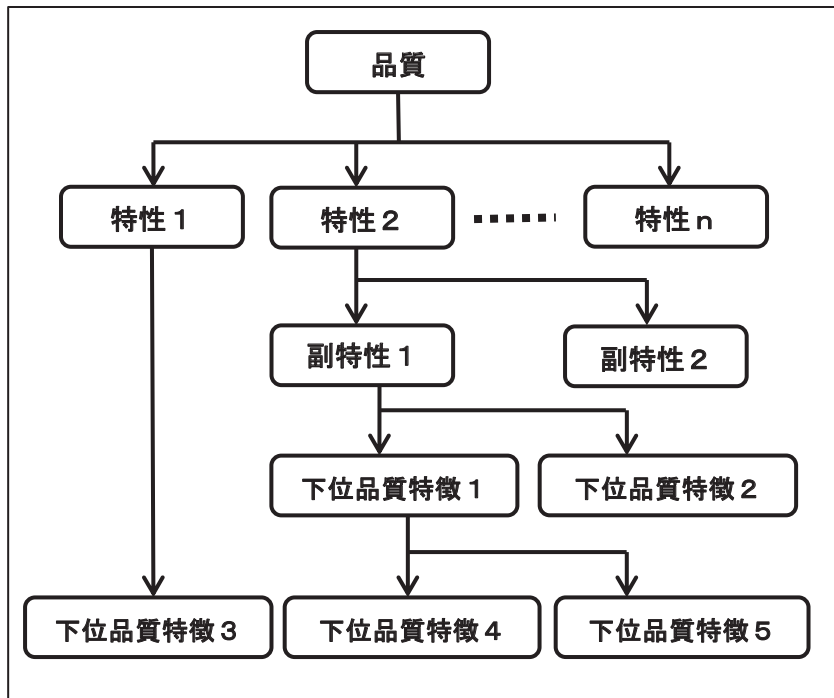


図 2-2 品質モデルに使われる構造  
(JIS X 25010:2013 から簡略化して作成)

利用時の品質とは、「利用者がある利用状況において、利用者のニーズに照らして、製品／システムを利用できる度合い」である。そのニーズとは、品質特性である有効性、効率性、満足性、リスク回避性、及び利用状況網羅性に関する特定の目標の達成を求めることである。利用時の品質には、5つの特性と11の副特性が定義されている。図 2-3 にその全体を示す。

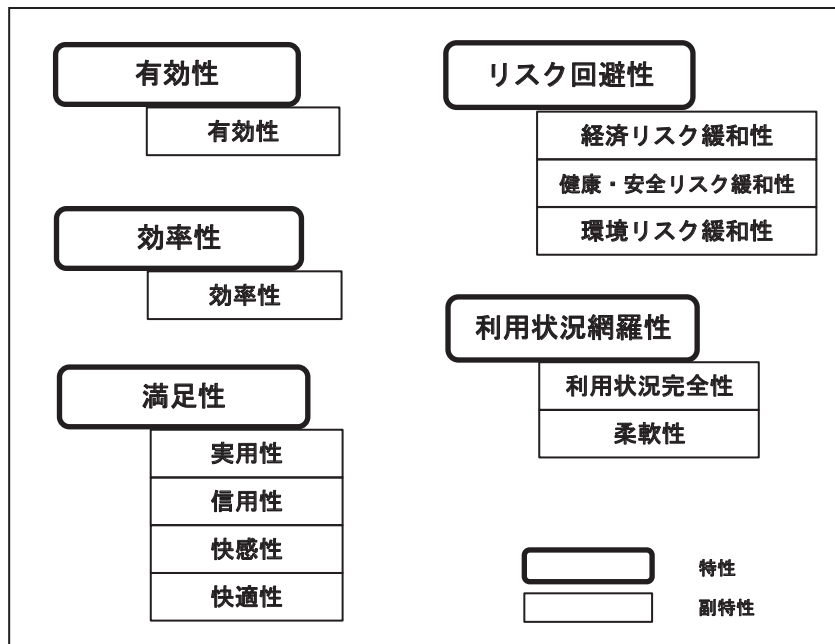


図 2-3 SQuaRE 品質モデル(利用時の品質)

**製品品質**とは、文字通りソフトウェア製品(コンピュータプログラム、手続き、関連文書と、パラメータ等のデータを含めた一つのまとまり)、またシステム製品の品質であり、利用者のニーズに照らして判断される。製品品質には、8つの特性(機能適合性、性能効率性、互換性、使用性、信頼性、セキュリティ、保守性、移植性)と31の副特性が定義されている。図2-4にその全体を示す。



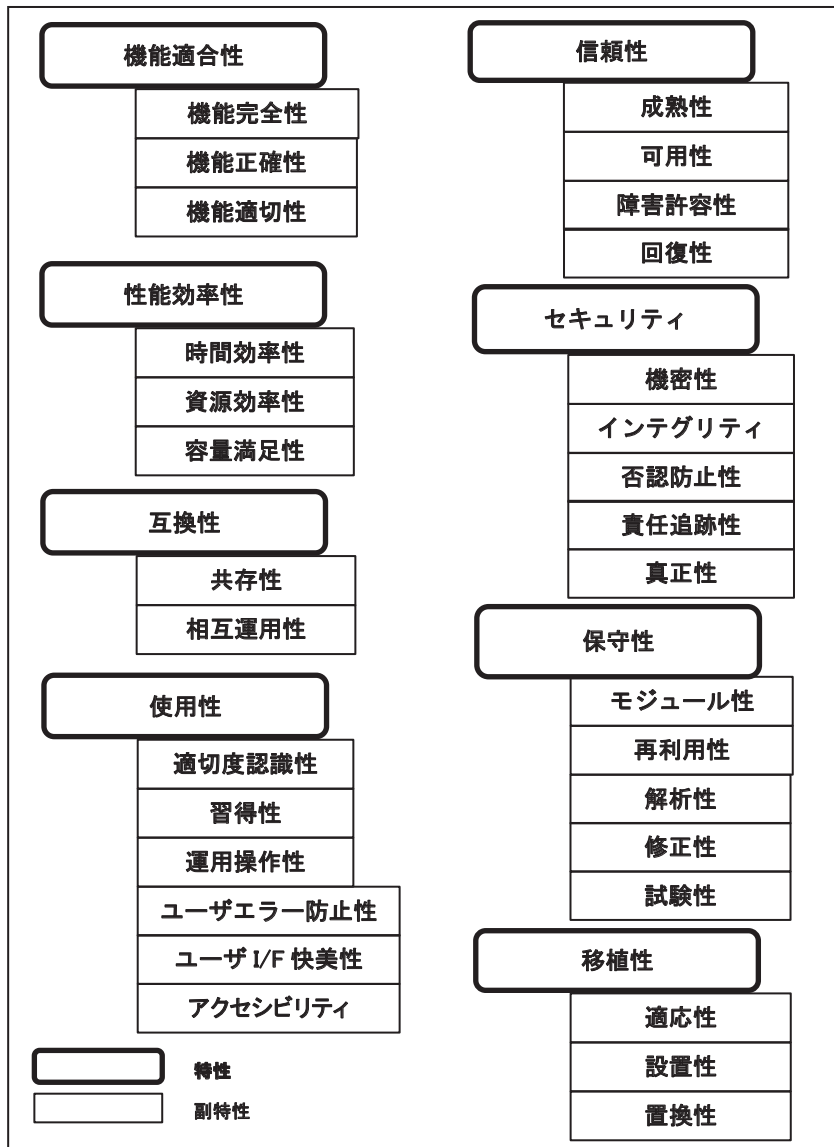


図 2-4 SQuaRE 品質モデル(製品品質)

データ品質とは、主としてコンピュータシステムが取り扱うデータや情報の品質を取り上げたものである。データ品質の品質特性は、「固有のデータ品質」と「システム依存のデータ品質」に分類されている。後者は、「ある条件下のコンピュータシステム中でデータ品質が到達・維持される度合い」

と定義される。それに対して、前者「固有のデータ品質」は、「データの品質特性が本来備えている潜在力の度合い」と定義される。すなわち、データだけを取り出してみた場合に問題となる品質が「(データ)固有のデータ品質」であり、システムの動作との関係でデータの品質が利用者に見えるのが「システム依存のデータ品質」である。図 2-5 にそれらの全体を示す。

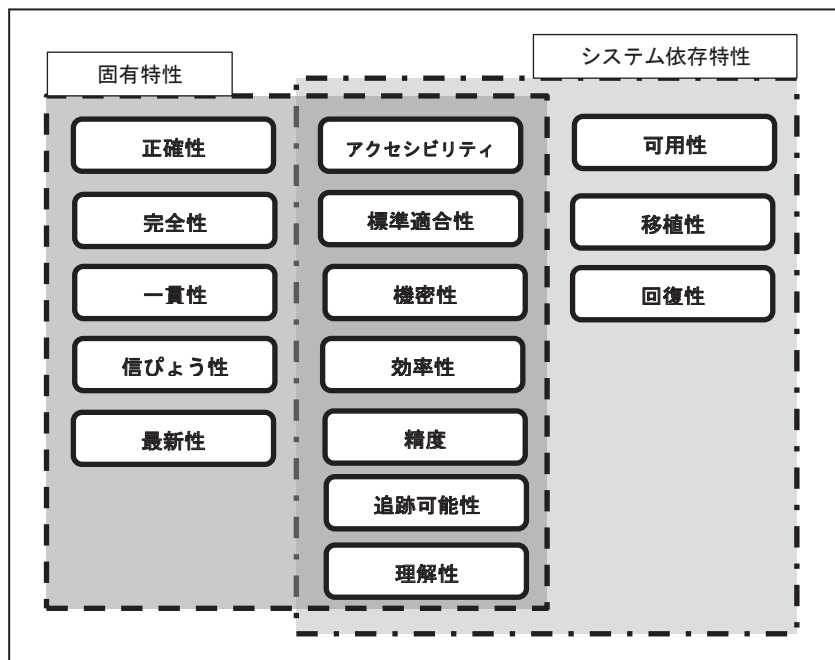


図 2-5 SQaRE 品質モデル(データ品質)

品質モデルは、システム/ソフトウェアの開発において、品質要求事項の明確化に利用することができる。システム/ソフトウェアへの品質要求事項のあり方を定めた ISO/IEC 25030 では、システム/ソフトウェア開発者に対して、品質要求事項を策定する上で、概ね次のような事柄を求めている。

- 品質要求事項の対象となる**システムの境界**を明確化する。
- すべての明示した利害関係者の**ニーズ**と期待が品質要求事項に文

書化されるようにする。

- c) **利害関係者の品質要求事項の妥当性を確認する。**
- d) システムの品質要求事項から対応する**ソフトウェア機能**や**ソフトウェアの品質要求事項**を割り当てる。
- e) 品質要求事項を、利用する品質特性モデルと関連付ける。製品品質／利用時の品質などの**品質モデル**のどれに該当するかを明確化する。
- f) どのような**測定量**を用いるか、またその**目標値**(値の範囲など)を明確化し、仕様として文書化する。
- g) ソフトウェア品質要求事項とソフトウェアの**機能仕様との関係性**を明示する。
- h) **測定量を選択する基準**を明示する。

「はじめに」で、「品質特性の状況を評価するには、まず品質要求と関連した評価基準を設定し、それに沿った測定が必要だというように考える。」と述べたが、品質測定の参照モデルを定めた ISO/IEC 25020 では、測定量の選択や新規の考案に関する基本的な要求事項を、概ね次のように規定している。

- a) 規格で与えられている**測定量の選択基準**を明示するか、新しい測定量と品質モデルとを関連づける。
- b) **測定量の定義と測定量要素の定義**を明確にする。
- c) **測定**を計画し、実施する。

品質要求事項に基づいて製品品質／利用時の品質やデータの品質目標が定められ、それに対する評価基準が設定されると、それらは、開発プロセスを進めるうえで、各プロセスにおける品質の目標値として具体化・詳細化されることが望ましい。それにより、レビューや試験等で検証可能となる。

品質特徴の要素を個別に測定した結果を品質測定量要素という。品質測定量要素はしばしば他の品質測定量要素と組み合わせて(測定の関数による計算の結果として)、品質測定量になる。品質測定量をさらに総合的視点から評価することにより、いろいろな品質(副)特性についての品質評価結果(目標が達成されたかどうか)が得られる(図 2-6 参照)。

品質測定量の例として、MTBF(平均故障間隔)を取り上げると、その構成要素となる品質測定量要素は、故障数と時間である。故障数は、故障の回数を何等かの基準を設けて(この基準設定が測定方法の定義にあた

る)カウントすることで測定できる。測定対象とする時間は稼働時間について同様に基準を設けて測定する。これらを用いて、測定関数[(時間) / (故障数)]を適用することにより、品質測定量 MTBF が計算される。(ここで詳しくいえば、故障数や時間も多くの測定の平均値である可能性がある。)そして、MTBF は、品質特性の一つである「成熟性」を「指し示している」ことになる。

### 測定量について

SQuaRE (ISO/IEC 25010)は、測定量について次のような説明をしている。

「試験時に発見された**故障の回数**は、コンピュータシステムに存在する**障害の数**に関するソフトウェア品質の外部測定量である。試験は全ての障害を発見できるわけではないので、**二つの測定量**は、必ずしも同じではない。そして、障害は、異なる環境において明らかに異なる故障を引き起こす。」

規格では、このように、実測された(される)測定量とその背後にある測定される対象となる特徴の違いを明確にするために、測定量と品質特徴という言葉を使い分けている。また、測定量は、個々の測定値が入る入れ物である変数を意味している。

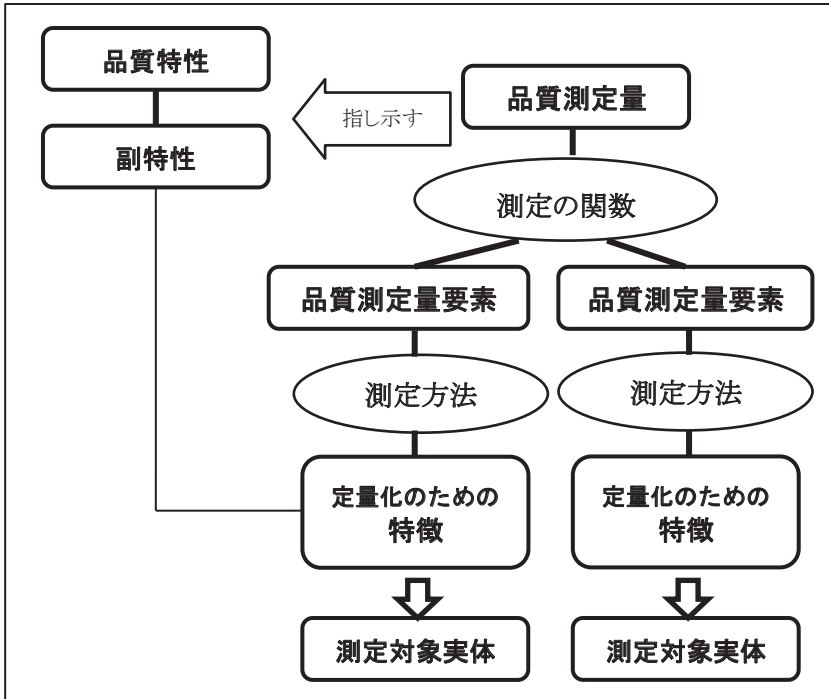


図 2-6 測定量とその要素  
(JIS X 25021:2014 から簡略化して作成)



# 第3章

## S Q u a R Eの構成

SQuaRE は、ISO/IEC 25000 シリーズ規格の呼称だが、そのシリーズ（規格類）は、全体として図 3-1 のような構成を持っている。

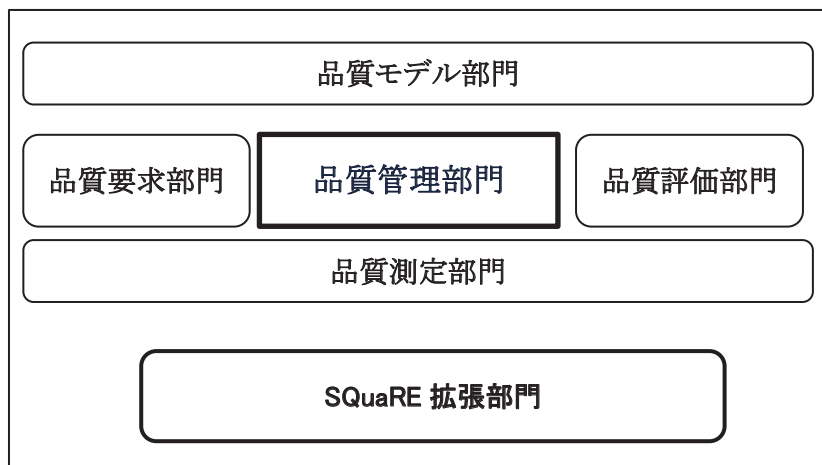


図 3-1 SQuaRE シリーズの構成

**品質管理部門(規格番号 2500n)**は、SQuaRE 全体に共通の考え方、用語などを定めるとともに、どのように SQuaRE 全体を利用したらよいかのガイドも提示している。

**品質モデル部門(規格番号 2501n)**は、製品品質・利用時の品質・データ品質の各観点から、詳細な品質特性・副特性を定義する。あわせて、品質モデル利用のための手引きも提供する。

**品質要求部門(規格番号 2503n)**は、品質要求事項の仕様化に役立つ作業指針を示している。どのように品質要求事項を定めればよいか、検証すればよいかも述べている。

**品質測定部門(規格番号 2502n)**は、品質測定の基本視点である「品質測定の参照モデル」を定めるほか、品質測定量の定義、具体的な品質測定の要素及び測定法も例示している。

**品質評価部門(規格番号 2504n)**は、品質モデルと品質測定の結果などに基づき、品質についての判定・評価を行うプロセスを定めている。評価基準には、仕様化された品質要求事項などが利用される。評価は、評価専門家、利用者、または開発者がそれぞれの立場から必要なやり方で



う。

**SQuaRE 拡張部門(規格番号 2505n~2509n)**は、システム/ソフトウェア製品一般ではなく、特定の応用範囲や特定のソフトウェアの種別についての製品品質を扱う規格類を扱う計画となっている。現在、既製ソフトウェア製品(RUSP<sup>1</sup>)ソフトウェア向けが開発済み、また、ユーザビリティに関わるものが開発中となっている。

これらの規格のうち、2014 年末時点で JIS 規格となっているものは表 3-1 の通りである<sup>2</sup>。

表 3-1 JIS 化されている SQuaRE の規格

| ISO/IEC<br>及び JIS<br>番号 | JIS<br>発行年   | JIS のタイトル<br>(2015 年時点のタイトル共通部分: システム及びソフトウェア製品の品質要求及び評価(SQuaRE)) |
|-------------------------|--------------|---|
| 25000                   | 2010         | SQuaRE の指針(改訂予定)  |
| 25001                   | 2012         | 計画及び管理(改訂予定)  |
| 25010                   | 2013         | システム及びソフトウェア品質モデル   |
| 25012                   | 2013         | データ品質モデル  |
| 25021                   | 2014         | 品質測定量要素   |
| 25030                   | 2012         | 品質要求事項  |
| 25040                   | 2014         | 評価プロセス  |
| 25041                   | 2015<br>(予定) | 開発者、取得者及び独立した評価者のための評価手引  |
| 25051                   | 2011         | 商用既成(COTS)ソフトウェア製品に対する品質要求事項及び試験に対する指示<br>(JIS 改訂中であり、タイトルも変更される) |

<sup>1</sup> RUSP は、Ready to Use Software Product の略。

<sup>2</sup> JIS 規格名称等のタイトルの最初の部分は、2015 年時点、「システム及びソフトウェア製品の品質要求及び評価(SQuaRE)」であるが、規格制定の始めのころは、「システム及び」がないタイトルも使われていた。



# 第4章

## 特性項目表の凡例

本書では、表 4-1 と表 4-2 に示す形式を用いて特性項目、副特性項目と、それらに付随する品質の測定方法例などを紹介する。

表 4-1 と表 4-2 の説明のうちで「品質(副)特性名／定義」の項目以外は、SQuaRE の文章内容ではなく、その趣旨を分かりやすくするために参考資料などをもとにして執筆した説明である。

表 4-1 品質特性表の形式

| 表の項目名    | 項目の説明   |
|----------|---|
| 品質特性名／定義 | 規格が定めた品質特性名を示す。<br>( ) 内に英語名を示し、その後に規格 (ISO/IEC 25010 または 25012) の対応する箇条番号を示す。<br>また規格における定義文を付記した (注記などは省略)。 |
| 説明       | 品質特性の簡潔な説明を示す。<br>【詳細】で、より踏み込んだ説明と解釈上の留意点を示す。   |
| ニーズ      | 利用 <sup>1</sup> 及び開発においてこの品質特性が必要となる状況の例を示す。  |
| リスク      | この品質特性の考慮漏れや不完全な実装が及ぼす影響の例を示す。  |
| 参考文献     | この品質特性をより深く理解するのに役立つ参考文献を示す。  |

表 4-2 品質副特性表の形式

| 表の項目名     | 項目の説明  |
|-----------|--|
| 品質副特性名／定義 | 規格が定めた品質副特性名を示す。<br>( ) 内に英語名を示し、その後に規格 (ISO/IEC 25010 または 25012) 内の対応する箇条番号を示す。<br>また、規格における定義文を付記した (注記などは省略)。 |

<sup>1</sup>利用者には、通常の IT システムの一次的利用者のほか、コンテンツプロバイダや保守作業者などの二次的利用者 (副次的利用者)、さらに、間接利用者 (システムの出力情報を業務等に利用する人、影響を受ける人) などが含まれる。これら多様な利用者ニーズをどのように汲み上げ評価するかが、どの品質特性でも問題となる。

|              |   |
|--------------|---|
| <b>説明</b>    | 品質副特性の簡潔な説明を示す。<br>【詳細】で、より踏み込んだ説明と解釈上の留意点を示す。  |
| <b>ニーズ</b>   | 利用及び開発においてこの品質副特性が必要となる状況の例を示す。   |
| <b>品質測定量</b> | 品質副特性の定量化に用いる測定量の例を示す。<br>必要に応じて、その測定方法、評価方法、品質目標などを補足する。<br>測定量の例示には、本書編集時点で審議中の国際規格案などに記載された例を参考としている。<br>品質目標の例示では、一部、経済産業省「ソフトウェアメトリクス高度化プロジェクト、プロダクト品質メトリクス WG:システム/ソフトウェア製品の品質要求定義と品質評価のためのメトリクスに関する調査報告書(2011)」などを参考としている。 |
| <b>参考文献</b>  | 必要に応じてこの品質副特性の参考文献を示す。  |

注1: 副特性が対応する品質特性と同一の場合、表は一部簡略化している。また、類似の副特性は場合により、合併して一つの表で示している。

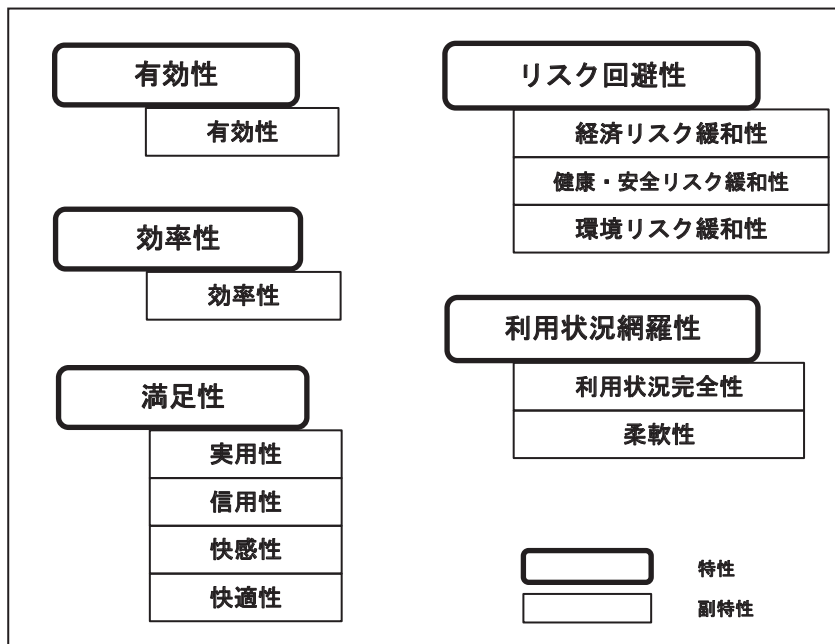
注2: データ品質については、副特性表は作成していない。



# 第5章

## 利用時の品質 品質特性表

以下に利用時の品質の各特性／副特性の内容をそれぞれ表の形で示す。





|              |  |
|--------------|--|
| 品質特性名<br>／定義 | <b>有効性 (effectiveness)</b><br><p style="text-align: right;">4.1.1</p> 明示された目標を利用者が達成する上での正確さ及び完全さの度合い。   |
| 説明           | 該当システムを利用すると、どの程度うまく、実施したい事柄のニーズの内容を達成できるか。<br><br><b>【詳細】</b><br>システムを利用して様々な業務を実施する場合に、その利用目的の一つは実施したい事柄(企業向けシステムでは「業務内容」)の品質(正確さや、完成度の高さ)の向上である。その向上への寄与度に関する特性である。<br>システムの利用において何らかの制限事項・前提条件がある場合には、そのことが明示されていないで導入されると、「実際には使えない」といった事態になることもある。そうした場合は有効性が低い(ない)といえる。 |
| ニーズ          | <ul style="list-style-type: none"> <li>・システムの導入効果、投資効果を業務品質の面から評価したい。</li> <li>・システムを結合して利用するとき、全体的な有効性がどのようになっているか評価したい。</li> </ul>   |
| リスク          | <ul style="list-style-type: none"> <li>・有効性の評価をしないため、システムを導入しても導入した目新しさだけで満足し、導入効果を具体的に把握できなかった。</li> <li>・システムを導入した結果、かえって業務品質が低下しているのに気が付かなかった。</li> <li>・会計ソフトが、今年度の税制までしか対応しておらず、翌年使えないことが後から判明した。</li> </ul>  |

|                  |   |
|------------------|---|
| <b>品質副特性名／定義</b> | <b>副特性 有効性 (effectiveness)</b><br>(品質特性「有効性」と同じ)  |
| <b>説明</b>        | (品質特性「有効性」と同じ)  |
| <b>品質測定量</b>     | <p>・『あるワープロソフトには数式入力機能があるが、機能や表示性能が不十分であるため、執筆論文 10 本のうち、3 本しかそのワープロでは完成せず、残りは他の文書ソフトを利用せざるをえなかった』。この場合、業務が完了したことの判定基準を明確にする必要がある。また、業務に重み付けをして集計することもある。</p> <div style="border: 1px solid black; padding: 10px; text-align: center; margin: 10px 0;"> <math display="block">\text{業務完了率} = \frac{\text{完了した業務項目数}}{\text{着手した業務項目数}}</math> </div> <p>・有効性に関するデータの処理の仕方として、有効性を測る手法、「課題達成率」があり、次の手順で算出する。<br/>         事前に課題の成功の条件と達成の度合の指針を決めておき、課題ごとの被験者の成功率を算出する。成功＝1、部分的成功＝0.5、失敗＝0など任意のスコアを用意する。</p> <p><b>【品質目標設定例】</b><br/>         「新バージョンでの業務のエラー率は旧バージョンより〇〇%以上改善」とすることにより新旧システムの有効性の改善目標を明示する。</p> |

|              |   |
|--------------|---|
| 品質特性名<br>／定義 | <b>効率性 (efficiency)</b> <p style="text-align: right;">4.1.2</p> 利用者が特定の目標を達成するための正確さ及び完全さに関連して、使用した資源の度合い。  |
| 説明           | 該当システムを利用すると、業務資源(作業時間、作業要員、エネルギー、材料等を含む)の使い方がどの程度効率的なのか。<br><br><b>【詳細】</b><br>様々な業務にシステムを利用する目的の一つは業務における使用資源量の上での効率の向上である。   |
| ニーズ          | <ul style="list-style-type: none"> <li>・システムの導入効果、投資効果を業務資源効率の面から評価したい。</li> <li>・システムを結合して利用するとき、全体の効率性がどのようになっているか評価したい。</li> <li>・業務効率や業務の拡張性の面で、同時に並行作業して効率を上げるため、同時アクセス可能な端末数が多いのがよい。</li> </ul>                               |
| リスク          | <ul style="list-style-type: none"> <li>・システムを導入してもその目新しさだけで満足し、業務効率上の導入効果を具体的に把握できなかった。</li> <li>・システムを導入した結果、かえって業務効率が低下しているのに気が付かなかった。</li> <li>・パッケージソフトウェア等のシステムを導入した時に、付加的に導入すべきシステム、周辺機器、消耗品等が結構多く、コストが予定より多大となった。</li> </ul> |

|           |   |
|-----------|---|
| 品質副特性名／定義 | <b>副特性 効率性 (efficiency)</b><br>(品質特性「効率性」と同じ)   |
| 説明        | (品質特性「効率性」と同じ)  |
| 品質測定量     | <p>・ (副) 特性の直接的な測定量の例。有効性が同じでも、業務完了に必要な時間が短くなれば効率性は向上する。</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <math display="block">\text{利用効率} = \frac{\text{達成した業務目標数}}{\text{業務に要した時間}}</math> </div> <p>NEM 評価では、熟練者(設計者)と一般利用者の操作時間を測定し、NE 比(熟練者の操作時間を1とした場合に一般利用者の操作時間がその何倍かの比)を計算する。操作ステップを横軸とし、操作時間と NE 比の値をプロットしたグラフから操作ステップの問題点を抽出する。</p> <p><b>【品質目標設定例】</b><br/>         「NEM 評価の NE 比が4.5倍以上にならないようにする」という目標を設定する。</p> |

|              |  |
|--------------|--|
| 品質特性名<br>／定義 | <h2>満足性 (satisfaction)</h2> <p style="text-align: right;"><b>4.1.3</b></p> <p>製品又はシステムが明示された利用状況において使用されるとき、利用者ニーズが満足される度合い。</p>   |
| 説明           | <p>システム利用によって利用者がどの程度満足するか。</p> <p><b>【詳細】</b></p> <p>満足性は、製品品質特性の使用性と強く関係しているが、使用性と比べて、利用時の品質としてシステムの品質を見ている。満足性は、利用者がシステムをどのように理解・認識できているかといった基本点にも関係し、有効性、効率性、リスク回避性といった他の利用時の品質にも関係している。</p> <p>開発者が使用性を試験するときに、利用者の実際の満足性の評価ができると、使用性設計の改善に貢献することがある。</p> <p>システムのリリース後の(専門的な)アンケート評価等で、ニーズに合致していたかどうかの満足性の程度を効果的に評価できることがある。また、利用者に対するヘルプデスクなどの支援サービスの提供が満足性を向上させることがある。</p> |
| ニーズ          | <ul style="list-style-type: none"> <li>・他システムとつながった状態での満足性を十分検討したい。</li> </ul>   |
| リスク          | <ul style="list-style-type: none"> <li>・開発者が自信を持って開発するあまり、実際の利用者のニーズに合致していなかった。</li> <li>・習熟していない利用者が誤った使い方をしたり、使用方法や指示するメニューの所在が分からず困惑したりした。</li> <li>・システムの利用手順や操作の組合せがあるか理解できず、また面倒な操作に嫌気がさすことがあった。</li> <li>・利用者が操作法、不具合等について相談したいが、連絡する方法が明示されておらず、不満が高まった。</li> </ul>  |
| 参考文献         | <ul style="list-style-type: none"> <li>・ユーザビリティテストング 共立出版 (2003) 黒須 正明編著</li> <li>・ユーザビリティハンドブック 共立出版(2007) 『ユーザビリティハンドブック』編集委員会</li> </ul>  |

|           |  |
|-----------|--|
| 品質副特性名／定義 | <p><b>副特性 実用性 (usefulness)</b></p> <p style="text-align: right;"><b>4.1.3.1</b></p> <p>利用の結果及び利用の影響を含め、利用者が把握した目標の達成状況によって得られる利用者の満足の度合い。</p>  |
| 説明        | <p>利用者が行いたいことが達成され、満足するか。</p> <p><b>【詳細】</b><br/>         利用者が実際のビジネスの場面や、生活の場面で、「これは使える」と評価できるかどうか。利用者にとっての目標の達成状況がキーポイントである。なお、利用に際して見出されるシステム不安定等の不具合点は「信用性」の副特性で扱う。</p>  |
| ニーズ       | <ul style="list-style-type: none"> <li>・ マーケットでソフトウェア製品が訴えたい点が、利用者の具体的な目標達成の程度や、それを利用者が納得できる程度、にある場合には、この副特性が基本的な品質の指標となるので十分検討したい。</li> </ul>   |
| 品質測定量     | <ul style="list-style-type: none"> <li>・ 利用者が、実用性を認めて利用した機能やシステム(製品)がどれかを相対評価する測定量の例として次のものがある。</li> </ul> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <math display="block">\text{選択利用率} = \frac{\text{特定の機能やシステムの利用回数}}{\text{評価対象全体の利用回数}}</math> </div> |

|                  |  |
|------------------|--|
| <b>品質副特性名／定義</b> | <b>副特性 信用性 (trust)</b> <p style="text-align: right;"><b>4.1.3.2</b></p> 利用者又は他の利害関係者がもつ、製品又はシステムが意図した通りに動作するという確信の度合い。   |
| <b>説明</b>        | 利用者がシステムの動作として提示され、期待して、行いたいことが達成されるという確信をどの程度持てるか。<br><br><b>【詳細】</b> システムが示す情報や動作が利用者からみて疑わしい場合には、システムの満足性は低下し、利用されない可能性もある。システムが活用されるためには、信用性・信頼感の醸成も重要な要因となる。         ネットバンキングやネットショッピングでは、信頼感ある取引が常に行われ、実施でき、利用者の不慣れな各種操作に対するシステムの対応も適切である必要がある。         災害警報システムは、その通知内容が常に正しく、タイミングも適切だという信用を持つ必要がある。         官庁の広報 Web システムは、データ内容の品質とともに、システム動作として常に正しい最新情報を提供しているという信用性が求められる。 |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・ 利用者のシステムへのアクセスにおける信用性を重視したい。セキュリティが保たれているか、個人情報の取扱いは大丈夫か、システム利用上で難しい点や動作欠陥はないかなど、利用者からみて不安になる点は多数ある。システムが利用者の不慣れな操作をカバーしてくれるか等にも配慮したい。</li> <li>・ ヘルプデスクのサービス品質を高め、利用者からの意見を反映して、製品改良のサイクルを適切に実施することにより信用性を高めたい。</li> <li>・ システムの保守性や保守サービス品質でも信用性を高めたい。</li> </ul>   |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・ 副特性の直接的な測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; text-align: center; margin: 10px 0;"> <p><b>計量心理学的信用性</b> =<br/>信用性に関する計量心理学的測定量</p> </div> <p>「計量心理学的測定量」とは、5段階尺度等を用いたアンケートの設計や結果集計に関して、心理学的な観点からのひずみ等を考慮しつつ数値的に取り扱っていく手法にもとづいた測定量の総称</p>  |

|             |   |
|-------------|---|
|             | <p>である。ここでは、単なる定量的調査結果ではなく、専門的な見地からの分析が必要であるという立場から、この用語を用いている。</p>   |
| <b>参考文献</b> | <ul style="list-style-type: none"><li>・ Jian, J-Y., Bisantz, A. M., Drury, C. G: Foundations for an empirically determined scale of trust in automated systems. International Journal of cognitive Ergonomics, 4, 1, 53-71 (2000)</li></ul> |



|                         |   |
|-------------------------|---|
| <p><b>品質副特性名／定義</b></p> | <p><b>副特性 快感性 (pleasure)</b></p> <p style="text-align: right;">4.1.3.3</p> <p>個人的なニーズを満たすことから利用者が感じる喜びの度合い。</p> <p><b>副特性 快適性 (comfort)</b></p> <p style="text-align: right;">4.1.3.4</p> <p>利用者が(システム又はソフトウェアを利用する時の)快適さに満足する度合い。</p>  |
| <p><b>説明</b></p>        | <p>知識欲や個性の発揮といった利用者の欲求を満たせるか。システム操作上の快適性を提供できるか。</p> <p><b>【詳細】</b></p> <p>快感性と快適性は似ているが、快感性の方は、使ってうれしかったという感覚に対応している。また快適性の方は、操作感がスムーズであったなど、素早いシステム動作に快適さを覚えたといった状況を表している。快適さがない状況では利用者のイライラ感が募ることになる。これらの品質特性は、製品品質モデルのユーザインタフェース快美性と関係がある。ユーザインタフェース快美性では、製品としての品質への注力の仕方や見栄えに焦点を当てているのに対して、ここでは利用者の主観的な受け取り方を問題としている。</p> <p>(注)快美性という言葉は一般にはなじみが薄いだが、辞書でも「快く美しいこと。また、非常に快いこと。また、そのさま」と説明されている(大辞林 第3版から)。英語の aesthetics はしばしば美学、美的センスなどと訳される。</p> |
| <p><b>ニーズ</b></p>       | <ul style="list-style-type: none"> <li>・システムが愛されて使い込まれるように、そのシステムにほれ込むといった好感の度合いを獲得することを目指したい。</li> <li>・マニュアルレスで使えるシステムなど、直観的に活用できるシステムを目指したい。それにより効率性が上がるとともに、利用者の快適性向上にも貢献できる。</li> </ul>  |
| <p><b>品質測定量</b></p>     | <ul style="list-style-type: none"> <li>・副特性の直接的な測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; text-align: center; margin: 10px 0;"> <p><b>計量心理学的快感／快適性 =</b><br/>快感／快適性に関する計量心理学的測定量</p> </div> <p>「計量心理学的測定量」については、「信用性」の副特性の項を参照。</p>   |

|      |   |
|------|---|
|      | <ul style="list-style-type: none"><li>・一般に品質特性「満足性」の測定量として、SUS(The System Usability Scale)がある。ユーザビリティ評価で広く使われる質問紙調査手法である。10 項目の尺度(それぞれ、「私はこのシステムを頻繁に利用したい」というような文章からなる)について、5 段階ないし 7 段階の賛成・反対の程度を選択させる手法(リッカート法)を用いる。最終的な総合得点は、0~100 の間に位置したものになる。</li></ul>   |
| 参考文献 | <ul style="list-style-type: none"><li>・Hassenzahl, M. &amp; Monk, A. (2010). The inference of perceived usability from beauty. <i>Human-Computer Interaction</i>, 25(3), 235-260.</li><li>・Hernandez, L. (2002). Evaluation of different scales for measurement of perceived physical strain during performance of manual tasks. <i>Int. J. Occup. Saf. Ergon.</i> 2002; 8(4):413-32.</li><li>・Watson, D., Clark, L. A. &amp; Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: the PANAS scales. <i>Journal of Personality and Social Psychology</i>, 54(6), 1063-1070.</li><li>・電子政府ユーザビリティガイドライン 附属文書 (<a href="http://www.kantei.go.jp/jp/singi/it2/guide/guide_line/siryou2.pdf">http://www.kantei.go.jp/jp/singi/it2/guide/guide_line/siryou2.pdf</a>)</li></ul> |

|                      |   |
|----------------------|---|
| <b>品質特性名<br/>／定義</b> | <b>リスク回避性 (freedom from risk)</b><br><p style="text-align: right;">4.1.4</p> 製品又はシステムが、経済状況、人間の生活又は環境に対する潜在的なリスクを緩和する度合い。   |
| <b>説明</b>            | 利用によって出会うかもしれない様々なリスクが低く抑えられているか。<br><br><b>【詳細】</b><br>システムを利用することにおいて何らかのリスクが生じることがある。そのようなリスクで悪影響を受けるものには経済的なもの、健康や身体の安全上のもの、また人間の生活環境全般に関するものが考えられる。リスク回避性の品質特性では、システムがどのように有効にそうしたリスクを生まないようにしているか、緩和してくれるかを評価する。<br>システムのステークホルダのうち、保守者の安全リスクの配慮が重要となることがしばしばある。また、間接利用者の様々なリスクに対する幅広い配慮が重要となることがある。  |
| <b>ニーズ</b>           | <ul style="list-style-type: none"> <li>・ システムの動作目的が直接に経済活動、健康や安全上の措置（医療機器の場合や、機械の機能安全装置の場合など）、環境保護（危険物質の漏えい防止装置など）に寄与している場合に、特にシステムの失敗によるリスクの分析と、リスクに対応する品質達成を重視したい。</li> <li>・ インテグリティレベルといったレベルを表す評価を用いて、リスク回避性の評価をしたい。</li> <li>・ 運用時のリスクを回避するために、運用時インシデント管理や運用保守サービスの品質を評価したい。</li> <li>・ 複合的なシステムでは、リスクの分析と回避等の対応はこれまでに比べて一層高度なものが求められる。</li> <li>・ システムの生むリスクの評価に際しては、製品の機能適合性、互換性、使用性などといった特性の評価が役立つことがあるため、組み合わせて評価したい。</li> </ul> |
| <b>リスク</b>           | <ul style="list-style-type: none"> <li>・ 利用すると高いリスクがある機能を起動した時に、その起動後の状態が表示されず、また、アクセス権限を逸脱しても適切に通知・拒否されないことがあった。</li> <li>・ リスクがある機能について、過去のログが残されていないため、インシデント発生時に実施履歴の点検ができなかった。</li> </ul>  |

|                  |  |
|------------------|--|
| <b>品質副特性名／定義</b> | <b>副特性 経済リスク緩和性</b><br><b>(economic risk mitigation)</b><br><br><b>4.1.4.1</b><br>意図した利用状況において、財政状況、効率の運用操作、商業資産、評判又は他の資源に対する潜在的なリスクを、製品又はシステムが緩和する度合い。  |
| <b>説明</b>        | システム利用によって出会うかもしれない経済的なリスク、ビジネスリスクが低く抑えられているか。<br><br><b>【詳細】</b><br>システム事故事例で、システムのデータチェック機能や入力データ修正機能が甘かったために多額のビジネス損失を出した例がある。また、セキュリティに関連して、個人情報の漏えい等が直接的な損害賠償や企業ブランド価値の損傷につながった例がある。安全性に関連して、製品のリコール等が広範に生じた例がある。そのような経済的な事故を防ぐシステムの機能や品質が幅広く求められる。   |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・上記の説明にあるような事故を幅広く防止するために、適切な要求事項を幅広い情報に基づいて設定したい。</li> <li>・消費者等からの訴訟リスクにも対応した堅実な開発を実施したい。</li> </ul>  |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・経済的な影響のあるエラーの発生確率に関わる基本的な測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; text-align: center; margin: 10px 0;"> <b>経済影響エラー率</b> =<br/> <math display="block">\frac{\text{経済影響エラー発生件数}}{\text{システム利用状況数}}</math> </div> 結果として生じる経済的影響を測定するための基本測定量はビジネスの特性に応じて多数考えられるため、品質評価の実情に応じて適切な基本測定量の組合せを考慮する必要がある。また、個々の品質特性がどのように経済的な影響を持つかを評価するのは難しく、事例経験に頼る場合も多くある。 |

- ・ 経済的な影響を事前に評価する立場の測定量の例。

$$\text{仕様外シナリオ想定度} = \frac{\text{仕様外のリスクシナリオを考慮している件数}}{\text{全件数}}$$

品質副特性  
名／定義副特性 **健康・安全リスク緩和性**  
(health and safety risk mitigation)

4.1.4.2

意図した利用状況において、製品又はシステムが人々に対する潜在的なリスクを緩和する度合い。

副特性 **環境リスク緩和性**  
(environmental risk mitigation)

4.1.4.3

意図した利用状況において、環境に対する潜在的なリスクを製品又はシステムが軽減する度合い。

## 説明

システム利用によって出会うかもしれない健康・安全上のリスクと環境破壊のリスクが低く抑えられている。

## 【詳細】

システム事故事例で、プログラムの不具合や、システムのデータチェック機能の甘さのために、システム誤動作、誤表示を起こした例がある(航空管制上のトラブルの誘発、自動車やロボットの不具合・危険動作、医療機器の誤診誘発など)。

日常の IT 端末操作でも、プログラムの不具合とは言えないが、操作者に頸肩腕症候群や眼精疲労への影響等が報告されたことがある。

制御装置や警報装置のソフトウェアの不具合により、機械の危険動作、災害への対応の遅れ、環境影響物質の漏えい等が懸念される事例もある。

そのような健康・安全上の、または環境上の事故を防ぐシステムの機能や品質が幅広く求められる。また、品質の状況について評価する必要がある。

## ニーズ

- ・上記の説明にあるような悪影響の発生を防止するために、リスク分析等にもとづいて適切な要求事項を幅広く設定したい。
- ・リスク発生の可能性が既知である機能がある場合は、そのことを漏れなく明示したい。
- ・熟練者向けのシステムで、利用者に特定の訓練や取扱い上の注意等が必要であることを、説明書等に明記するとともに、利用者の訓練環境を整備したい。
- ・医療機器の分野で、健康・安全面での法的な規制、業界標準等に対応したい。
- ・システムの保守時の作業者の安全性や、保守作業のミス誘発によるセーフティハザードに十分に留意したい。

|                     |  |
|---------------------|--|
| <p><b>品質測定量</b></p> | <p>・ 次のような測定量は、例えば医療機器使用現場において、患者の中で正しくない治療を受けた患者の割合といった測定量として利用できる。</p> <div style="border: 1px solid black; padding: 10px; text-align: center; margin: 10px 0;"> <math display="block">\text{システム利用で影響を受ける対人安全性} = \frac{\text{ハザードにさらされる人数}}{\text{全利用人数}}</math> </div> <p>【品質目標設定例】<br/>医療機器ソフトウェアの開発にあたって、医療機器の安全性に関しても規定している IEC 62366:2007 を指針として適用し、安全性確保方策、品質目標とすることがある。</p> |
| <p><b>参考文献</b></p>  | <ul style="list-style-type: none"> <li>・ IEC 62366:2007 (医療機器へのユーザビリティ工学の適用)</li> <li>・ JIS T 2304:2012(IEC 62304:2005) 医療機器ソフトウェア – ソフトウェアライフサイクルプロセス</li> <li>・ IEC 65 委員会(工業用プロセス計測制御;IEC 61508を開発)や、ISO/IEC JTC1/SC27 (IT セキュリティ技術) 委員会が提供している諸規格及びその分野別規格</li> </ul>  |

|                      |  |
|----------------------|--|
| <b>品質特性名<br/>／定義</b> | <b>利用状況網羅性 (context coverage)</b><br><br><b>4.1.5</b><br>明示された利用状況及び当初明確に識別されていた状況を超越した状況の両方の状況において、有効性、効率性、リスク回避性及び満足性を伴って製品又はシステムが使用できる度合い。  |
| <b>説明</b>            | 提供している利用時の品質は、いろいろな利用状況の違いに関わらず提供できるか。<br><br><b>【詳細】</b><br>たとえば、特定のコンピュータや OS、周辺機器のみを対象とするシステムもあるが、より広い、多様な条件で利用できるとよい場合もある。そのような対応可能条件の広さや柔軟性を評価する品質特性である。<br>利用状況網羅性では、IT 機器を用いる作業環境(温度・湿度等を含む)や、多様な人的特性(熟練度や好み、利き手といったものを含む)を持つ利用者の存在が考慮されているかどうか、という観点も重要である。そのため、利用者ごとのカスタマイズ機能といったものも必要となる可能性がある。またこの観点は、製品品質モデルのアクセシビリティ副特性とも関連する。  |
| <b>ニーズ</b>           | <ul style="list-style-type: none"> <li>・ 市場に存在する多様なコンピューティングプラットフォームに対応して、システムがどの程度の多様な範囲で正常に稼働可能か、製品企画・設計・実装・試験に対応して明確にしたい。</li> <li>・ プラットフォームやその他の利用条件も時間とともに変化する。そうした変化に対してどの程度対応できるか明確にしたい。</li> <li>・ 多様な利用者の範囲や、過酷な稼働(作業)環境をどこまで見込むかの判断を正確に行いたい。また、操作のやり直しといった、利用状況の多様なあり方をできるだけ網羅したい。</li> <li>・ 稼働可能なプラットフォームの範囲がどの程度かの認識について、ベンダーとして稼働を確認・保障する範囲を定義するほか、利用者が試みた稼働可能プラットフォームの情報も収集したい。</li> </ul> |
| <b>リスク</b>           | <ul style="list-style-type: none"> <li>・ 実装や試験で用いた環境とは異なるプラットフォームに展開することが困難となった。</li> <li>・ 利用する情報機器の画面サイズ、メモリサイズ、利用者の能力等に制約があるにも関わらず、そのことが認識・説明されず、利用者からクレームが寄せられた。</li> <li>・ システムがいろいろな利用条件に合わせた設定パラメータを持っているにもかかわらず、現在の設定や変更の方法が分かりやすく提示されていないため、利用者が利用環境に合わせた使い方ができなかった。</li> </ul>   |



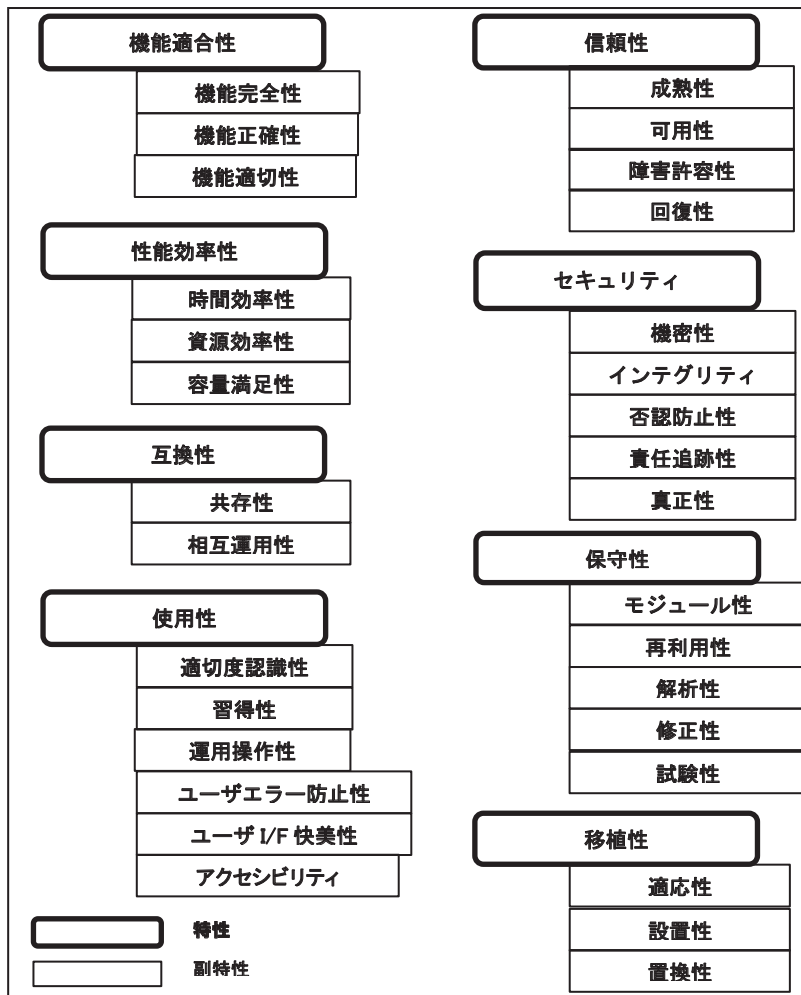
|                  |   |
|------------------|---|
| <b>品質副特性名／定義</b> | <b>副特性 利用状況完全性</b><br><b>(context completeness)</b><br><br><div style="text-align: right;"><b>4.1.5.1</b></div> 明示された全ての利用状況において、有効性、効率性、リスク回避性及び満足性を伴って製品又はシステムが使用できる度合い。  |
| <b>説明</b>        | 最初から意図した範囲では完全に、いろいろな利用状況の違いに関わらず利用時の品質を提供できるか。<br><br><b>【詳細】</b><br>利用状況完全性のイメージは、「携帯機器等の小さい画面を使用したり、低いネットワーク帯域幅しか使えなかったり、熟練していない利用者が使ったり、また、オフラインモードで使ったりした場合でも、ソフトウェアが使用可能である程度」と説明されている。<br>この副特性では、利用状況網羅性のうち、開発者があらかじめ利用を想定している範囲の広さと、その対応策の実現の程度を問題としている。                     |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・ 複合的なシステムでは、どのようなシステムの要素の組合せで、ニーズを実現できるのかの見極めが、ベンダー側からも利用者側からも難しくなるため、特に注意して利用状況網羅性を明確にした。</li> <li>・ 利用者特性ごとに網羅的にシナリオベースのデザインを実施したい。シナリオの網羅性は、タスクの特徴を列挙する場合や、ヒューマンエラー／想定外誤操作事象を列挙する場合などがある。</li> </ul>                                       |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・ 副特性の直接的な測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; text-align: center; margin: 10px 0;"> <p><b>利用状況完全性</b> =</p> <math display="block">\frac{\text{適切な使用性とリスク水準が維持可能な利用状況数}}{\text{要求にあるすべての利用状況数}}</math> </div> |

|                  |   |
|------------------|---|
| <b>品質副特性名／定義</b> | <b>副特性 柔軟性 (flexibility)</b><br><p style="text-align: right;"><b>4.1.5.2</b></p> 要求事項の中で初めに明示された状況を逸脱した状況において、有効性、効率性、リスク回避性及び満足性を伴って製品又はシステムが使用できる度合い。   |
| <b>説明</b>        | 最初から意図したのとは違う状況でも、ある程度の状況の違いに関わらず利用時の品質を提供できるか。<br><br><b>【詳細】</b><br>利用状況網羅性のうち、あらかじめ意図しなかった状況でも対応可能な程度。柔軟性が不足している場合、意図していない状況で製品を使用すると安全でなくなる可能性があり、そのような、「意図していないが、想定としてはありうる利用」での安全性等の確保が重要となることもある。また、柔軟性が高いことで、利用者の創造性を刺激することもある。   |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・システム開発時に想定しなかった対象システム／人との連携や、環境が変化する中でも稼働させたい。</li> <li>・当初は特定のスキルを持った利用者を前提としていたが、より広範囲の利用者に提供するようになった場合でもシステムを少ない改良で対応させたい。</li> </ul>  |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・企業向けのシステム開発等で、要求事項が増えていくことが予想されるときに、副特性の直接的な測定量の例として次のものがある。</li> </ul> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <math display="block">\text{製品柔軟性} = \frac{\text{追加要求に対して既存機能で対応可能な件数}}{\text{追加要求の総件数}}</math> </div> <p>「顧客に利用条件として示した PC スペックを満たさなくとも、国内で 2 年以内に購入されていると推定されるすべての PC 機種で稼働するようにする」といった品質目標を立てることがある。</p> |

# 第6章

## 製品品質 品質特性表

以下に製品品質の各特性／副特性の内容をそれぞれ表の形で示した。



|                      |  |
|----------------------|--|
| <b>品質特性名<br/>／定義</b> | <b>機能適合性 (functional suitability)</b><br><p style="text-align: right;">4.2.1</p> 明示された状況下で使用するとき、明示的ニーズ及び暗黙のニーズを満足させる機能を、製品又はシステムが提供する度合い。   |
| <b>説明</b>            | 機能がステークホルダのニーズにどの程度うまく適合しているか。<br><br><b>【詳細】</b><br>製品やサービスは、入力／表示／計算／動作など、多くの機能を持っているが、それらがステークホルダの本来のニーズにどの程度適合しているかの度合いである。<br>これは、機能項目の内容を列挙することではなく、機能が総合的な見地から「適合」していて過不足がないかどうかを問題としている。<br>実際の開発においては、開発途中で何らかの理由により、予定していた機能を狭める、あるいは利用に制限をかけたりすることがあるが、その場合、適合の度合いは低くなる。<br>機能には、利用者の目的に沿った、「目に見える(説明書に明記された)」ものだけでなく、例えば利用者の間違った使い方を制限したり、データ訂正作業を支援したり、万が一障害が発生した場合に利用者やそのデータの安全を確保したりするような、文章の上では見えにくいものもある。 |
| <b>ニーズ</b>           | <ul style="list-style-type: none"> <li>・当初の機能仕様が抽象的であったり、複雑な利用条件があったりする場合に、実装の結果がニーズに合っているかどうか点検・評価したい。</li> <li>・開発プロセスにおける「妥当性確認」といった事柄と関連づけて機能適合性を検討したい。</li> <li>・他システムとつながった状態での機能適合性を幅広く検討したい。</li> </ul>  |
| <b>リスク</b>           | <ul style="list-style-type: none"> <li>・製品説明に利用者が呼び出せる機能がすべて書かれていない、また逆に、実装機能が説明書記載通りに網羅されていなかった。</li> <li>・安全性やビジネスリスクに関係する機能の内容が十分説明されず、あるいは実装されていないため、利用者がリスクにさらされた。</li> <li>・ソフトウェア機能が異なるバージョンや製品導入時の選択肢がある場合に、その説明が不十分であったため、利用現場での導入時の必要機能オプションの選択漏れがあった。</li> <li>・データ容量やデータの精度、範囲等に制約があるにも関わらず、そのことが適切に説明されず、または不適切に実装された。</li> <li>・ソフトウェアやその機能に対する不適切なアクセス(うっかりミスなどを含む)とその誤動作が適切に防護されていなかった。またそ</li> </ul>      |

- の防護機能の有効範囲などが適切に説明されていなかった。
- ・操作手順書やインストールガイドの内容が、実際の作業順序を反映しておらず、省略があった。

|                  |   |
|------------------|---|
| <b>品質副特性名／定義</b> | <b>副特性 機能完全性</b><br><b>(functional completeness)</b><br><p style="text-align: right;">4.2.1.1</p> 機能の集合が明示された作業及び利用者の目的の全てを網羅する度合い。   |
| <b>説明</b>        | 機能が、必要な作業や利用目的達成の範囲をどの程度網羅しているか。<br><br><b>【詳細】</b><br>各種ステークホルダの本来のニーズに対して機能が十分に網羅されているか、また利用者の実際の利用時の満足に足る機能が十分に網羅されているかどうか、さらに本当に実装されているかどうかを問題としている。  |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・ 機能要求事項はレビューされていて良好だが、仕様上の要求機能件数が非常に多く、比較的短いプロジェクト期間にそのすべてが実装及び試験されたのかどうか確認したい。</li> <li>・ 旧システムで、機能が不足し、業務で必要となる作業の一部が面倒な手作業を必要とすることがあり、また、機能が不足し、データや操作の後日での訂正等が適切にできないことがあったので新システムでは対処したい。</li> <li>・ 複合的なシステムでは、システムの要素間で相互に期待する機能の完全性を十分検討したい。</li> <li>・ パッケージソフトウェアの各種バージョン間に機能上や法的規制への対応上などでの違いがあり、それらが適切にマニュアル等に反映されているか確認したい。</li> </ul> |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・ 副特性の直接的な測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <math display="block">\text{機能実装率} = \frac{\text{実際に実装された機能件数}}{\text{機能要求事項中の機能件数}}</math> </div> <p>実務上で入力データの訂正など特定の操作が必要であるにも関わらずそれが実装されていない、といった機能不足をカウントすることも測定量として考えられる。</p>   |

|                  |  |
|------------------|--|
| <b>品質副特性名／定義</b> | <b>副特性 機能正確性</b><br><b>(functional correctness)</b><br><p style="text-align: right;"><b>4.2.1.2</b></p> 正確さの必要な程度での正しい結果を、製品又はシステムが提供する度合い。  |
| <b>説明</b>        | 機能がどの程度に正確な処理機能や出力データを提供しているか。<br><br><b>【詳細】</b><br>もともとの各種ステークホルダのニーズや、利用者の実際の利用に際しての満足から見て、製品の機能が、きちんとした正確なデータや機能を提供しているかどうかを問題としている。正確さのうちには、精度や桁あふれ処理、端数処理の正確さといった課題もある。  |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・工学的な設計の計算で、データの精度や誤差の程度を評価したい。</li> <li>・会計計算で、端数処理が正しく行われているかどうか評価したい。</li> <li>・システムの要素間で授受されるデータ等の正確性の程度が異ならないか評価したい。</li> </ul>  |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・副特性の直接的な測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <math display="block">\text{機能正確率} = \frac{\text{正しく実現されている機能件数}}{\text{機能要求事項中の機能件数}}</math> </div> <p>工学的な計算では精度見積り・比較やシミュレーション計算を用いた精度試験の結果等が測定量となる。</p> |



|                  |   |
|------------------|---|
| <b>品質副特性名／定義</b> | <b>副特性 機能適切性</b><br><b>(functional appropriateness)</b><br><p style="text-align: right;">4.2.1.3</p> 明示された作業及び目的の達成を、機能が促進する度合い。  |
| <b>説明</b>        | 機能とそれに対応する操作が不必要に煩雑なものとなっていないかどうか。業務の的確な実施を促進しているかどうか。<br><br><b>【詳細】</b><br>同じ機能を提供するとしても、煩雑な手順や、手戻りを起こしやすい機能の組合せが提供されているは、利用者の満足が得られない。<br>この副特性は使用性の品質特性とも関係するが、ここでは機能がニーズに適合しているかどうかという観点から問題としている。<br>要求事項を確立する場合に即して言えば、機能要求の確立では、「特定のある機能ができること」を中心に述べているのに対して、この機能適切性の品質要求では、「その機能あるいは関連する機能群によって業務が適切に促進できること」を要求する。 |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・ 同じようなデータを入力する画面で毎回すべてのデータの入力が必要となるような場合に、複写機能やデフォルト入力機能などがどの画面にもあることが望ましいことがある。</li> <li>・ コンテキストに応じて適切なメニュー選択肢を提供するといった機能性を十分考慮したい。</li> <li>・ 業務の効率化のためのシステムでは、システムが業務の流れを適切に反映したものとなっているか評価したい。</li> <li>・ システムの要素の機能の組合せが不必要に複雑になっていないか評価したい。</li> </ul>                         |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・ 副特性の直接的な測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <math display="block">\text{機能適切率} = \frac{\text{適切に実現されている機能件数}}{\text{機能要求事項中の機能件数}}</math> </div>  |

|             |   |
|-------------|---|
|             | <p>・機能適切性の測定量として次のような測定量もある。</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"><math display="block">\text{機能仕様安定率} = 1 - \frac{\text{ライフサイクル中に改訂された機能件数}}{\text{機能要求事項と利用者マニュアル中の機能件数}}</math></div> |
| <b>参考文献</b> | <p>・JIS Z 8520:2008 人間工学—人とシステムとのインタラクション—対話の原則</p>   |

|                      |   |
|----------------------|---|
| <b>品質特性名<br/>／定義</b> | <b>性能効率性</b> (performance efficiency)<br><div style="text-align: right;">4.2.2</div> 明記された状態(条件)で使用する資源の量に関する性能の度合い。  |
| <b>説明</b>            | システムの実行性能が効率的か、効率的に資源を使っているか。<br><br><b>【詳細】</b><br>利用時の品質にも「効率性」があるが、そちらは業務としての効率性を問題としているのに対して、この性能効率性は、システム自体の動作や直接使用するコンピュータ周辺の資源(計算時間、使用メモリ量、使用媒体量、通信量、印刷量など)を問題としている。どれくらいの労力を必要とするかという「人の稼働量」については、利用時の品質として扱われる。<br>個々のシステム要素の性能から期待される性能効率が全体としての様々な状況に対応できているかどうかの問題となることがある。 |
| <b>ニーズ</b>           | <ul style="list-style-type: none"> <li>・ 技術計算等で、実際の技術上の解に到達できるまでに要する時間を評価し短縮したい。</li> <li>・ 利用者との対話型システムで、一つ一つの操作の応答時間を評価したい。</li> <li>・ メモリ量が低く抑えられた組込みシステムで、特殊な状況でも稼働できるようメモリ使用量を低く抑えたい。</li> <li>・ 結合された他のシステムからの応答を待つ時間など、統合されたシステムとしての性能効率性を評価し、実用性を高くしたい。</li> </ul>              |
| <b>リスク</b>           | <ul style="list-style-type: none"> <li>・ 対話型システムで、処理の応答時間が長く、またそのことへの警告表示・途中経過表示もないため、利用者はシステムが不具合に陥ったのか、正常に稼働しているのか判断できなかった。</li> <li>・ システムとしての性能効率の低下により、利用時の品質としての利用者の業務効率も大きく低下させてしまった。</li> </ul>   |

|                  |  |
|------------------|--|
| <b>品質副特性名／定義</b> | <p><b>副特性 時間効率性 (time behaviour)</b><br/>4.2.2.1<br/>製品又はシステムの機能を実行するとき、製品又はシステムの応答時間及び処理時間、並びにスループット速度が要求事項を満足する度合い。</p> <p><b>副特性 資源効率性 (resource utilization)</b><br/>4.2.2.2<br/>製品又はシステムの機能を実行するとき、製品又はシステムで使用される資源の量及び種類が要求事項を満足する度合い。</p>   |
| <b>説明</b>        | <p>システムの実際の動作速度や、システムが利用する実際の資源の種類と量が、システムへの要求を満たすか。</p> <p><b>【詳細】</b><br/>この二つの効率性は、それぞれ時間と、資源(時間を除く)とを対象としている。この副特性項目ではシステムを動かすための技術的な効率性を問題としているが、利用時の品質の効率性では、広く業務上の資源が問題となる。</p> <p><b>時間効率性</b>は主に次の3点が課題となる。</p> <ul style="list-style-type: none"> <li>— 応答速度</li> <li>— 一定の処理を完了するのに要する時間</li> <li>— 一定時間に何件の処理ができるかというスループット</li> </ul> <p><b>資源効率性</b>は主に次のような資源を考慮する。</p> <ul style="list-style-type: none"> <li>— 使用 CPU 時間</li> <li>— 使用メモリ</li> <li>— 使用媒体(ディスク、外部媒体など)</li> <li>— 通信回線占有率</li> <li>— その他の外部資源(プリンタなど)占有率</li> </ul> |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・業務システムにおいて、システム稼働上の時間や資源が少なくて済むスリムな運用を実現したい。</li> <li>・メモリ使用量等が大きな問題となる組込みシステムで、そうした制約に対応できているかどうかを評価したい。</li> </ul>   |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・副特性の説明で言及している要素の時間・資源効率性を測定するものとして次のものがある。</li> </ul>  |

$$\text{ターンアラウンドタイム充足率} = \frac{\text{ジョブ完了時刻} - \text{ジョブ開始時刻}}{\text{目標ターンアラウンド時間}}$$

|                  |  |
|------------------|--|
| <b>品質副特性名／定義</b> | <b>副特性 容量満足性 (capacity)</b> <p style="text-align: right;">4.2.2.3</p> 製品又はシステムのパラメータの最大限度が要求事項を満足させる度合い。   |
| <b>説明</b>        | 取り扱えるデータのサイズや量、投入される速度、また利用者数などが、システムへの要求を満たすか。<br><br><b>【詳細】</b><br>この容量満足性とは、システムが取り扱えるデータ量、データ投入速度や、同時アクセス可能数などの能力を扱うものである。  |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・ データベースや通信回線の容量など非機能要求を正確に評価・対応したい。</li> <li>・ データベースや Web サイトなどで、運用時の同時アクセス数を見積もり、対応できるシステム容量を設計・構築したい。</li> </ul>  |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・ オンラインシステムでの典型的な測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; text-align: center; margin: 10px auto; width: fit-content;"> <p><b>オンラインリクエスト処理率</b> =</p> <math display="block">\frac{\text{処理されたリクエスト件数} / \text{運用時間}}{\text{最大処理容量}}</math> </div> |

|                      |   |
|----------------------|---|
| <b>品質特性名<br/>／定義</b> | <b>互換性 (compatibility)</b><br><br><p style="text-align: right;"><b>4.2.3</b></p> <p>同じハードウェア環境又はソフトウェア環境を共有する間、製品、システム又は構成要素が他の製品、システム又は構成要素の情報を交換することができる度合い、及び／又はその要求された機能を実行することができる度合い。</p>   |
| <b>説明</b>            | <p>他のシステムなどと適切につながるかどうか、共存できるかどうか。</p> <p><b>【詳細】</b></p> <p>あるシステム／ソフトウェアが、他のシステム／ソフトウェアと相互に情報をやりとりする場合に適切に／円滑に情報交換が行われること、あるいは、相互に独立して動くべき場合に、相互に相手方の動作の障害とならないこと。</p> <p>他のシステムと情報交換をするシステムでは、特にこの互換性に対する注意が必要となる。動作環境と、想定する他システムをできるだけ明確にする必要がある。自システムと他のシステムという2者間の観点だけでなく、同時に稼働するシステム全体としての要素間の互換性を考慮しなければならない場合もある。情報交換する他のシステムの範囲には、周辺機器や通信回線の利用可能な種類なども含まれる。</p> |
| <b>ニーズ</b>           | <ul style="list-style-type: none"> <li>・ 互換性は利用時の品質の利用状況網羅性を支えるものだと位置づけて、評価・向上させたい。</li> <li>・ 他のシステムとのインタフェースとなるデータの品質を、システムのアウトプットの品質として評価・向上させたい。</li> </ul>   |
| <b>リスク</b>           | <ul style="list-style-type: none"> <li>・ システム／ソフトウェアが特定のバージョンのプラットフォームや特定の周辺機器に依存しているにも関わらず、そのことが明示されなかった。</li> <li>・ インストール時に、その時の環境に応じた適切な設定の選択ができなかった。</li> <li>・ システム／ソフトウェアの特定の設定が他のシステムの動作を妨げた。</li> <li>・ ソフトウェアの機能の一部しか他のシステムとの互換性がないにも関わらず、その制限が明示されなかった。</li> <li>・ 他のシステムとの互換性がないためにインストールができない、あるいはプラットフォーム等の設定変更が必要になることが、事前に利用者に明示されなかった。</li> </ul>        |

|                  |   |
|------------------|---|
| <b>品質副特性名／定義</b> | <b>副特性 共存性 (co-existence)</b><br><p style="text-align: right;"><b>4.2.3.1</b></p> その他の製品に有害な影響を与えずに、他の製品と共通の環境及び資源を共有する間、製品が要求された機能を効率的に実行することができる度合い。  |
| <b>説明</b>        | 他のシステムと、相互に妨害することなく、共存できるかどうか。<br><br><b>【詳細】</b><br>一つのハードウェア環境や一つのオペレーティングシステム環境のもとで、複数のシステム／ソフトウェアが共存して稼働でき、コンピュータ資源の共用にも支障がないこと。  |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・他のソフトウェアと共通の動的部品を利用する場合に、どのような動的部品とそのバージョンを利用するか明確にし、他のソフトウェアの動作と両立するかを幅広く点検したい。</li> </ul>   |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・副特性の直接的な測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; text-align: center; margin: 10px 0;"> <math display="block">\text{利用可能な共存率} = \frac{\text{共存可能なソフトウェア件数}}{\text{運用環境で共存を求められるソフトウェア件数}}</math> </div> <b>【品質目標設定例】</b><br>品質目標として、「新規開発ソフトウェアと、市場で広く利用されているソフトウェア及び自社ソフトウェアの各バージョンとの共存性は、製品試験時に試験項目に取り入れる」といった方針をとり、共存性の評価(共存が確認できる件数の評価)をすることが考えられる。 |



|                  |  |
|------------------|--|
| <b>品質副特性名／定義</b> | <b>副特性 相互運用性 (interoperability)</b><br><b>4.2.3.2</b><br>二つ以上のシステム, 製品又は構成要素が情報を交換し, 既に交換された情報を使用することができる度合い。  |
| <b>説明</b>        | 他のシステムなどと適切に情報交換し, つながるかどうか。<br><br><b>【詳細】</b><br>情報交換をするシステム同士が, 適切に情報交換できるかどうか, またそのように適切に情報交換できる対象の範囲はどの程度かを問題にする。<br>情報交換の手段・経路には, メモリ上でのやりとり, 通信回線を通じたやりとり, ファイルを通じたやりとりなどがある。   |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・ ネットワークを通じたデータ交換や入出力ファイルを取り扱うシステムにおいて, 交換するデータの形式の対応可能範囲や形式の規格のバージョン等を厳密にしたい。</li> <li>・ 情報交換をする相手システムや稼働する環境にさまざまなバージョンやバリエーションがある場合に, 対応の可否を明確にしたい。</li> </ul>   |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・ ファイルを通じて情報交換するシステムについての測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <math display="block">\text{データ交換率} = \frac{\text{情報交換可能なファイルフォーマット数}}{\text{情報交換したいファイルフォーマット数}}</math> </div> |

|                      |  |
|----------------------|--|
| <b>品質特性名<br/>／定義</b> | <b>使用性 (usability)</b> <p style="text-align: right;"><b>4.2.4</b></p> <p>明示された利用状況において、有効性、効率性及び満足性をもって明示された目標を達成するために、明示された利用者が製品又はシステムを利用することができる度合い。</p>  |
| <b>説明</b>            | <p>「利用時の品質」を支えるために、製品自体が技術的に見てどの程度使いやすいか。</p> <p><b>【詳細】</b><br/>         使用性は、操作や外観に関して使用のしやすさを問題とする品質特性である。<br/>         使用性がカバーする範囲は広く、六つの副特性によって具体的な内容が定められている。<br/>         「利用時の品質」は、全体として、この製品品質モデルの「使用性」に対応する事柄をそれぞれの品質特性の内容として持っているということもできる。<br/>         使用性が対象とする製品の範囲には、利用者マニュアルや保守マニュアルなども含まれる。マニュアルの使い勝手は、使用性の一部として重要なことがある。</p> <ul style="list-style-type: none"> <li>・ ネットショッピング等で、クレジットカード支払い画面が他システムである時に、その画面を含めた使用性を考えたい場合がある。</li> </ul>  |
| <b>ニーズ</b>           | <ul style="list-style-type: none"> <li>・ 会話的なシステムでは、人とシステムとの迅速で確実なコミュニケーションの確保を全体的に目指したい。</li> <li>次のような選択肢から適切なユーザインタフェースを選びたい。</li> <li>— コマンドラインインタフェース(キーボードからの命令文字列入力により、システムへの指示を与える。)</li> <li>— キャラクターベースのユーザインタフェース(CUI; 矢印キー等を含むキーボード入力と、文字ベースの表示を行う。)</li> <li>— グラフィックベースのユーザインタフェース(GUI;ビットマップ表示と、ポインティングデバイス等の使用により、グラフィカルな表示と操作を提供する。)</li> <li>— 音声や動作によるインタフェース</li> <li>— センサー等による自動データ読み取り</li> <li>・ オペレーティングシステム(OS)や、Web ブラウザが標準的に提供する操作仕様を徹底活用したい。</li> <li>・ 例えばEコマース(ネットショッピング)のサイトで、次のような幅広い「利用者」の「使用性」を十分に考慮したい。</li> <li>— 購入利用者</li> <li>— 出店者</li> <li>— ショッピングカート運営者</li> </ul> |

## リスク

- ・ビジネス推進の核となる Web サイトが見栄えがしなかったり、操作性が悪かったりして、再度利用することが敬遠された。
- ・ユーザインタフェースの種別や方針が明確にされないため、一つのシステム／ソフトウェア内で統一がとれていなかったり、利用者の期待に反する理解しにくいインタフェースが提供されたりした。
- ・アプリケーションの使用やシステムとしての運用操作に必要な知識が明確にされないため、知識の少ない利用者の利用が困難となり、エラーや利用できない例が多発した。
- ・個人の好みや利用の事情に合わせるためのカスタマイズの「設定」の方法やそうした設定の支援ツールの利用方法が分かりやすく提示されていないだったので、多くの機能が実際には使われなかった。
- ・ソフトウェアの知的財産権を保護するために行われる各種の使用制限措置が明示されず、制限措置によって利用者が利用場面を狭められた。
- ・アクセシビリティの対応範囲が狭く、かなり多くの人利用から締め出された。または、説明書等にアクセシビリティの範囲が明示されておらず、障がいを持つ人にとっては利用困難であるにも関わらず誤って購入される事例が出た。
- ・システムが表示するメッセージ類が分かりにくく、利用者が理解しないまま、次のステップに進み、誤操作のようになってしまった。また、エラーメッセージについて、そのエラーにどう対処したらよいか不明な例が多発した。

品質副特性  
名／定義副特性 適切度認識性  
(appropriateness recognizability)

4.2.4.1

製品又はシステムが利用者のニーズに適切であるかどうかを利用者が認識できる度合い。

## 副特性 習得性 (learnability)

4.2.4.2

明示された利用状況において、有効性、効率性、リスク回避性及び満足性をもって製品又はシステムを使用するために明示された学習目標を達成するために、明示された利用者が製品又はシステムを利用できる度合い。

## 説明

システムが利用者のニーズに合っているかどうか、製品やマニュアルから容易に判断できるか、また、システム活用法が製品やマニュアルから容易に習得できるか。

## 【詳細】

この二つの副特性は、一般利用者がシステム／ソフトウェアを比較・検討したり、あるいは使い始めたりするときに問題となる。

**適切度認識性**は、購入・導入したりする際に、実際に購入・導入し使いこむより前に、利用者のニーズに合っているかの判断が容易であるかどうかを問題としている。パッケージソフトの場合、製品の無償トライアル利用や機能制限版の無償提供が行われることもあるが、この時、初回起動後に短時間で良し悪しを判断できることが求められる。

**習得性**は、次のようなことがあるかどうかを問題としている。

<特別な学習や訓練なしで利用される場合>

- 時間をかけて習得したりマニュアルを詳しく見たりしなくても自然な流れで操作できる。
- ヘルプ機能や適切なワーニング、エラー告知等を手掛かりに容易に操作できる。

<何らかの程度の学習や訓練がある場合>

- 簡潔な説明を受けたり、マニュアルを読んだりすることによって利用できる。
- 自習支援機能や、効果的な習得コースが提供されている。

## ニーズ

- ・利用者としてどのようなニーズと知識・スキルの背景がある人が利用するかを的確に判断したい。
- ・どのような利用者にも利用してほしいというシステムの場合に、直観的に馴染みややすく、親しみのわくユーザインタフェースにより、習得性の難易度を特に低くしたい。

## 品質測定量

- ・適切認識性の副特性に関する測定量の例。

$$\text{デモ表示装備率} = \frac{\text{デモ表示能力がある機能数}}{\text{製品記述にある全機能件数}}$$

- ・初回起動後の画面表示に着目した測定量の例。

$$\text{起動直後に表示される画面内の主たる作業項目表示率} = \frac{\text{起動直後に表示される画面内の主たる作業項目数}}{\text{起動直後に表示される画面内のメニュー項目数}}$$

- ・習得性の副特性に関する測定量の例。

$$\text{利用者マニュアル・ヘルプ完全率} = \frac{\text{利用者マニュアル・ヘルプで適切に説明されている機能件数}}{\text{実装されていて文書化の必要がある機能件数}}$$

製品又はシステムの基本機能、応用機能、及び優位さを述べている特徴(サービス、タスク、もしくは機能)については、文書化の必要があり、特に、利用者にとって重要度が高く、利用頻度の高い機能については、利用状況を想定した説明が有効となる。

品質副特性  
名／定義

## 副特性 運用操作性 (operability)

4.2.4.3

製品又はシステムが、それらを運用操作しやすく、制御しやすくする属性をもっている度合い。

副特性 ユーザエラー防止性  
(user error protection)

4.2.4.4

利用者が間違いを起こすことをシステムが防止する度合い。

## 説明

システムがどの程度使いやすく、また利用者の間違いを防止してくれるか。

## 【詳細】

**運用操作性**は、操作や表示に一貫性があり使いやすいこととともに、現在状態表示やパラメータ設定機能等によって操作しやすさの向上や操作の中断といった制御が可能であることを表す。ネットショッピングのサイトで、購入のための様々な操作を途中でやめたり、内容変更したりできること、経過や最終結果(購入完了など)の確認ができることも、この運用操作性の課題に含まれる。運用操作性とは、operabilityの訳語だが、日本語では、運用性と訳すべき場合と、操作性と訳すべき場合とがある。SQuaRE では、まとめて運用操作性と訳している。

**ユーザエラー防止性**は、利用者がメニュー選択やデータ入力等の際に間違わないようにガードすることとともに、利用者が間違えたり、試し入力をしたりしたときに、訂正入力できたり、元の状態に復帰できたりする能力に対応している。

## ニーズ

- ・ 利用者の煩雑なデータ入力負担を軽減するために、コンテキストに依存したメニュー表示をしたり、入力欄によく使われるデフォルト値を表示したりする方針を徹底したい。
- ・ 重大な結果を引き起こす可能性がある利用者入力に対しては事前や実行時に警告を出したい。また、入力データの範囲や組合せが不正な場合は、それを識別し、警告・入力拒否したい。
- ・ システムに習熟していない利用者は、入力の途中経過では多数の誤りを犯すことがある。利用者がエラー入力をした場合に、その訂正作業が容易なユーザインタフェースを提供したい。
- ・ エラー表示では、例えば、複数ある入力欄のうちでエラー箇所はどの欄であるかを明示したり、複数のエラーがある場合にそのすべてを一挙に表示したりするなどの対応を品質方策としたい。
- ・ 画面遷移をした場合に、すでに入力した正しいデータを再度入

|       |  |
|-------|--|
|       | <p>力させることがないように配慮したい。</p> <ul style="list-style-type: none"> <li>・利用者の基本データや作業途中データは、保存でき、再入力によるエラー防止をしたい。</li> <li>・不正防止のためのログアウト、サインアウトの方法は分かりやすく提示し、ログアウトが実施されたことはその場で確認できるようにしたい。</li> </ul>   |
| 品質測定量 | <ul style="list-style-type: none"> <li>・運用操作性の副特性の測定量の例。</li> </ul> <div data-bbox="297 376 967 517" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <math display="block">\text{操作遷移(戻る動作)の滑らかさの度合い} = \frac{\text{戻る動作が適切に設計されている数}}{\text{戻る操作の全数}}</math> </div> <ul style="list-style-type: none"> <li>・画面操作のしやすさの測定量の例。</li> </ul> <div data-bbox="297 603 967 743" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <math display="block">\text{画面操作のしやすさの度合い} = 1 - \frac{\text{間接操作のための画面遷移数}}{\text{タスク遂行のための全画面遷移数}}</math> </div> <ul style="list-style-type: none"> <li>・ユーザエラー防止性の副特性の測定量の例。</li> </ul> <div data-bbox="297 823 967 1015" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <math display="block">\text{ユーザエラー検査率} = \frac{\text{入力エラーが識別され明確に修正されるユーザ入力項目数}}{\text{エラー検出可能なユーザ入力項目数}}</math> </div> |

|                  |  |
|------------------|--|
| <b>品質副特性名／定義</b> | <b>副特性 ユーザインタフェース快美性</b><br><b>(user interface aesthetics)</b><br><br><b>4.2.4.5</b><br>ユーザインタフェースが、利用者にとって楽しく、満足のいく対話を可能にする度合い。   |
| <b>説明</b>        | ユーザインタフェースの見栄えや、対話のリズム感などの考慮をしているか。<br><br><b>【詳細】</b><br>特に一般利用者を対象とするシステムでは、利用者の好みにも対応でき、また優れた鑑賞物、対話相手としての性質も持っていることが望ましい。<br>利用時の品質の満足性でも同様のテーマを取り扱っているが、ここでは、製品開発時にエンジニアリングとして追求する観点からユーザインタフェースの快美性というものを定義している。<br>快美性は、受け取り手の性別や年齢その他に起因する好みに依存するものであることに留意する必要がある。不要なマンガや動画の提示にも不快を感じる利用者もいる。このようなことに対応するには、多くの利用者に対応できるようにするアプローチもあり、対象とする利用者の範囲を明確化するというアプローチもある。  |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・色の使用やグラフィックデザインなどの見栄えの要素を重視した方針としたい。</li> <li>・視覚的な要素だけでなく、聴覚的な要素、その他の感覚要素も取り入れたい。</li> </ul>  |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・この副特性のうち、利用者の好みに適応できるという側面を表す測定量の例。(I/F はインタフェースの略)</li> </ul> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <math display="block">\text{ユーザ I/F 要素のカスタマイズ可能率} = \frac{\text{カスタマイズ可能要素数}}{\text{ユーザ I/F 要素総数}}</math> </div> <ul style="list-style-type: none"> <li>・この副特性のうち、一定の寸法リズムを提供することで利用者が心地よい印象を得られる側面を表す測定量の例として次のものがある。</li> </ul> <ol style="list-style-type: none"> <li>(1) 表示要素のプロポーシオン(設計寸法比)が一定の等比級数の組み合わせで構成されている自然界の法則(重力や生物の成長)にもとづく寸法体系を用いている程度を評価する。</li> <li>(2) 表示要素のプロポーシオン(設計寸法比)が一定の等差級数</li> </ol> |



の組み合わせで構成されている建築分野で活用され、人体寸法との整合性を持つ寸法体系を用いている程度を評価する。

**【品質目標設定例】**

この副特性のうち、利用者が心地よい印象を得られる色彩構成（色の組み合わせの問題）の側面を表す測定量の例としてコントラストの評価がある。コントラスト品質目標設定例として、次の場合がある。

- (1) 白地に黄色の文字や、黒字に茶色の文字のように、コントラストがほとんどないと問題となり、カラーモニターによっても色は違って見えるので、背景色とテキストの色の組み合わせについて注意深いテストが必要との方針をたてる。
- (2) Web コンテンツの前景色と背景色のコントラストが十分かどうかを判定する基準に、W3C の『WCAG 2.0』で定義されている「コントラスト比」の計算式を用いることがある。

|                  |  |
|------------------|--|
| <b>品質副特性名／定義</b> | <b>副特性 アクセシビリティ (accessibility)</b><br><b>4.2.4.6</b><br>製品又はシステムが、明示された利用状況において、明示された目標を達成するために、幅広い範囲の心身特性及び能力の人々によって使用できる度合い。  |
| <b>説明</b>        | 幅広い範囲の心身特性及び能力の人々がシステムを利用できるよう製品が配慮されているか。<br><br><b>【詳細】</b><br>アクセシビリティの副特性は、システム／ソフトウェアを、障害のある人を含むすべての人が快適・有効に使用できる程度を問題としている。<br>高齢者や、場合によっては子供も、アクセシビリティの考慮の対象となる。多くの人々にとって、流行語、略語、専門用語(業界用語)が理解できないことも多々ある。開発者の好みに合わせるのではなく、対象利用者をよく見極める必要がある。<br>アクセシビリティの範囲には、多様な言語、特に少数民族の言語をカバーしているかどうかを含めることがある。  |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・ Web のページ上などで広報をする場合、その情報をできるだけ多くの人にまんべんなく伝えたい。</li> <li>・ 社会的な要請により、アクセシビリティの規格を満たしたい。</li> </ul>   |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・ 視覚障害や聴覚障害にしばってアクセシビリティの測定を課題とした場合の、測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; margin: 10px 0; text-align: center;"> <math display="block">\text{視聴覚障害者のアクセシビリティ} = \frac{\text{視聴覚障害のある利用者が利用可能な機能数}}{\text{実装されている総機能件数}}</math> </div> <ul style="list-style-type: none"> <li>・ 視覚障害にしばってアクセシビリティの測定を課題とした場合、測定する品質特徴の例として次のものがある。</li> </ul> Web サイト利用タスク準拠 Web アクセシビリティチェックリスト <ol style="list-style-type: none"> <li>a) 表示速度</li> <li>b) 概要の理解</li> <li>c) 完全なアクセス</li> <li>d) 利用者による操作</li> <li>e) 本文の読みやすさ</li> <li>f) 適切な音声読み上げ</li> <li>g) コンテンツの視認性</li> </ol> |

|             |   |
|-------------|---|
|             | <p><b>【品質目標設定例】</b></p> <p>品質目標として、JIS X 8341 「高齢者・障害者等配慮設計指針—情報通信における機器, ソフトウェア及びサービス」での要求事項をできるだけ満たすようにすることがある。</p>   |
| <b>参考文献</b> | <ul style="list-style-type: none"><li>• JIS X 8341 シリーズ (ISO 9241 シリーズ) 「高齢者・障害者等配慮設計指針—情報通信における機器, ソフトウェア及びサービス」</li><li>• <a href="http://www.ueyesdesign.co.jp/guideline/waiv_ucd/content.html">http://www.ueyesdesign.co.jp/guideline/waiv_ucd/content.html</a></li></ul> |

|                      |   |
|----------------------|---|
| <b>品質特性名<br/>／定義</b> | <b>信頼性 (reliability)</b><br><br><div style="text-align: right;"><b>4.2.5</b></div> 明示された時間帯で、明示された条件下に、システム、製品又は構成要素が明示された機能を実行する度合い。   |
| <b>説明</b>            | <p>システムが、期待どおり動かないといった不具合から脱却できているのはどの程度（頻度、時間など）か。</p> <p><b>【詳細】</b><br/>         信頼性という用語は、元来は、故障等により製品が使用できなくなる状況に対応して、そうした事態が軽減される程度として考えられた。ソフトウェアでは、故障の原因は、機械類のように摩耗等ではなく、主として、開発過程での不具合、利用環境の変化、セキュリティ上の攻撃などから生じるので、特有の軽減策が必要となる。<br/>         システムが停止したり誤動作したりする場合、そこから早期に復帰する必要がある。また、システムの停止や復帰に際して、取り扱っていたデータが壊れることがあり、そのデータの回復も課題となる。<br/>         なお、信頼性と類似の言葉に、「ディペンダビリティ」という考え方がある。</p> |
| <b>ニーズ</b>           | <ul style="list-style-type: none"> <li>・人命に関わるシステムや、ノンストップ運用のシステムにおいて、システム運用上の基本的な問題として、信頼性を確保するための分析と方策を重視したい。</li> </ul>  |
| <b>リスク</b>           | <ul style="list-style-type: none"> <li>・システム利用上のエラー発生時に、連動する物理システムが制御不能になり、ソフトウェアや取扱いデータが使用不能になった。</li> <li>・システムの制御不能・使用不能状態からの回復手段がなかった。</li> <li>・利用者データや設定したパラメータ等のデータの退避手段がなかった。</li> <li>・入力データが制限違反であるにもかかわらず、それを検知できなかった。</li> <li>・データの自動バックアップ機能や、ソフトウェアの致命的な障害時の自動再起動、多重系による担保といった措置が施されていなかった。</li> </ul>   |

|           |  |
|-----------|--|
| 品質副特性名／定義 | <p><b>副特性 成熟性 (maturity)</b></p> <p style="text-align: right;"><b>4.2.5.1</b></p> <p>通常の運用操作の下で、システム、製品又は構成要素が信頼性に対するニーズに合致している度合い。</p>  |
| 説明        | <p>システムが十分にテストされ、または、実運用で使い込まれ、どの程度長く正常に稼働するといえるか。</p> <p><b>【詳細】</b><br/>この副特性は、システムはしっかりテストされ、または、使い込まれ、なかなか故障しないといった品質への信頼感と関係している。また、開発中のシステムでは、残存欠陥密度などの低減度合を判断する。運用時の測定量の見地からは、故障の頻度はどうか、故障なくどの程度の長時間の稼働を続けているかという面で評価する。<br/>(注)成熟性の定義で「通常の運用操作」と言っているのは、例外的でない運用操作が行われている(少なくとも停止していない)状態またはその時間帯のことである。</p> |
| ニーズ       | <ul style="list-style-type: none"> <li>・バグの収束等を厳密に評価し、テストの十分性を評価したい。</li> <li>・新規に稼働を始めたシステムは、初期不良発生の可能性があるものとして運用上の措置をとる方がよい場合がある。初期不良を防ぐために、成熟性の基準を設けて、対応するシステム運用試験を厳重に行うといった方策を考えたい。</li> <li>・使い込まれ、実務上で多くのケースで評価されている再利用部品は、新規の開発で成熟性の観点から信頼して利用できる可能性があるため、再利用を促進したい。</li> </ul>                               |
| 品質測定量     | <p>・出荷前の代表的な測定量の例。式の「信頼性に関わる」とは、例えばエラー処理(方針・実装・試験)が不適切である等。</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <math display="block">\text{欠陥修正率} = \frac{\text{設計／コーディング／試験で各修正完了した欠陥数}}{\text{指摘(発見)された信頼性に関わる欠陥総数}}</math> </div>                                   |

- ・運用に入ってからからの代表的な測定量の例。

$$\text{平均故障間隔(MTBF)} = \frac{\text{総稼働時間}}{\text{システム/ソフトウェア故障発生件数}}$$

「ノンストップ稼働のシステムであるため、MTBF は初期不良なしに〇年程度を目標とする」といった品質を目標とする。

|                  |   |
|------------------|---|
| <b>品質副特性名／定義</b> | <b>副特性 可用性 (availability)</b><br><p style="text-align: right;"><b>4.2.5.2</b></p> 使用することを要求されたとき、システム、製品又は構成要素が運用操作可能及びアクセス可能な度合い。  |
| <b>説明</b>        | システムが利用可能状態にあるのはどの程度(の時間割合)か。<br><br><b>【詳細】</b><br>システムがまれに短時間停止しても、すみやかに復旧し、トータルで見ると、ほぼ常に利用できていれば、可用性は確保されていると言える。<br>可用性を保つには、システムのサービスが不本意にダウンしない、また情報にアクセス不能にならないということとともに、回復性や障害許容性といった特性をシステムが持つことが必要であり、総合的な品質特性であるといえる。                                |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・ 計画的な保守期間を除き、24H 休止することなくサービスを続けるシステムとして可用性を確保したい。</li> <li>・ 最低限の可用性のレベルを高く確保するために、縮退運転(一部の機能を一部の構成要素だけからなるシステムで提供する)機能を持つことにし、可用性への寄与の程度を評価したい。可用性のニーズと、可用性を構成する品質特性に関わる構成要素の特性(MTBF 等)からの故障のリスクの見積りを行いたい。</li> </ul> |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・ 副特性の直接的な測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <math display="block">\text{稼働率} = \frac{\text{実際に稼働した総時間}}{\text{運用スケジュールで規定された稼働時間}}</math> </div>                   |

|                  |   |
|------------------|---|
| <p>品質副特性名／定義</p> | <p><b>副特性 障害許容性（耐故障性）</b><br/>(fault tolerance)</p> <p style="text-align: right;"><b>4.2.5.3</b></p> <p>ハードウェア又はソフトウェア障害にもかかわらず、システム、製品又は構成要素が意図したように運用操作できる度合い。</p> <p><b>副特性 回復性（recoverability）</b></p> <p style="text-align: right;"><b>4.2.5.4</b></p> <p>中断時又は故障時に、製品又はシステムが直接的に影響を受けたデータを回復し、システムを希望する状態に復元することができる度合い。</p>  |
| <p>説明</p>        | <p>システムが正常運用できなくなり、運用困難や停止となった場合への対応が有効・迅速にできるか。</p> <p><b>【詳細】</b></p> <p><b>障害許容性</b>は、耐故障性ともいう。異常なデータの検出やソフトウェアの意図しない誤動作の検出時に、利用者へのシステムのサービスを何等かの内容で継続することができるように対処しておくこと、利用者は安心してシステムを利用することができる。このような対処は、アプリケーションのレベル、基本ソフトウェアのレベルなど、様々なレベルが考えられる。</p> <p><b>回復性</b>は、システムが提供するサービス（関連するデータを含む）が実際に回復するまでの時間等に注目している（関係が深い特性に「保守性」があるが、こちらは、保守作業がどの程度容易かという視点から見ている）。そのため、システムを回復させるような機能の実装など、システム全体としての動きや、保守の在り方とも関係する。</p> |
| <p>ニーズ</p>       | <ul style="list-style-type: none"> <li>故障の発生率とともに、障害許容性のニーズを把握したうえで、システム／ソフトウェアの停止時に取り扱っているデータの保全を図るための定期自動データセーブの機能を実装する。</li> </ul>   |
| <p>品質測定量</p>     | <ul style="list-style-type: none"> <li>副特性の直接的な測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; text-align: center; margin: 10px 0;"> <math display="block">\text{平均回復時間} = \frac{\text{回復に要した時間総計数}}{\text{回復が必要となった件数}}</math> </div> <p>回復時間の目標値は RTO (Recovery Time Objective) と呼ばれる。</p>  |



|                      |   |
|----------------------|---|
| <b>品質特性名<br/>／定義</b> | <h2 style="margin: 0;">セキュリティ (security)</h2> <p style="text-align: right; margin: 0;"><b>4.2.6</b></p> <p>人間又は他の製品若しくはシステムが、認められた権限の種類及び水準に応じたデータアクセスの度合いをもてるように、製品又はシステムが情報及びデータを保護する度合い。</p>   |
| <b>説明</b>            | <p>システムが、情報や機能に対するアクセス制御と、情報やデータの保全とをどの程度行っているか。</p> <p><b>【詳細】</b></p> <p>ここでは主としてデータベースや、オペレーティングシステムなど管理された領域でのセキュリティを念頭に置いているため、セキュリティの説明にアクセスの保護という概念を用いている。</p> <p>この特性の記述に出てくる用語の意味は、セキュリティのコミュニティで用いられている用語の意味と完全には一致していないので、注意が必要である(例えばインテグリティの使い方については、副特性のインテグリティの解説箇所を参照)。</p>   |
| <b>ニーズ</b>           | <ul style="list-style-type: none"> <li>・ 現在では、セキュリティは多様な要求があるため、それらに対してどの程度、どのような手段で対応していくのが妥当か、見極めたい。</li> <li>・ セキュリティのニーズの程度とリスクの認識に応じて、セキュリティ機能の実装や、セキュアな設計、ソフトウェア実装におけるコーディング規約などの適用の程度を決定したい。</li> <li>・ 形式手法の適用など、セキュリティをある程度保証できるアプローチを考えたい。</li> <li>・ セキュリティに関する認証制度に対応した設計を行いたい。</li> </ul>  |
| <b>リスク</b>           | <ul style="list-style-type: none"> <li>・ セキュリティの具体的な対処法が分からないため、要求仕様に「セキュリティ事故が一切起こらないこと」とだけ記載したが、実装ベンダーは対応できなかった。</li> <li>・ ニーズが不明であるため、セキュリティの規格で要請されているすべての項目を実装しようとしたが、コストが膨大となり、途中で断念せざるを得なくなった。</li> <li>・ アクセス制御の方策として、システム管理者には全権限を与え、一般利用者には自分のデータへのアクセス権限だけを与える設計をしたが、それでは、利用者が利用時のシステムパラメータの変更等ができず、結局一般利用者もシステム管理者権限で利用することとなってしまった。</li> <li>・ 公開されている Web 画面において、システムの詳しいエラーメッセージ表示を行ったため、攻撃者が容易にシステムの構造を理解し、次の攻撃に利用した。</li> </ul> |

|                  |  |
|------------------|--|
| <b>品質副特性名／定義</b> | <p><b>副特性 機密性 (confidentiality)</b> <span style="float: right;">4.2.6.1</span></p> <p>製品又はシステムが、アクセスすることを認められたデータだけにアクセスすることができることを確実にする度合い。</p> <p><b>副特性 インテグリティ (integrity)</b> <span style="float: right;">4.2.6.2</span></p> <p>コンピュータプログラム又はデータに権限をもたないでアクセスすること又は修正することを、システム、製品又は構成要素が防止する度合い。</p>   |
| <b>説明</b>        | <p>システムが情報の機密保護やデータ保護の観点からどの程度アクセス制御を実施しているか。</p> <p><b>【詳細】</b></p> <p><b>機密性</b>は、システム／ソフトウェアの機能として、定められたアクセス権限の逸脱がないように管理する品質上の能力を定めている。</p> <p><b>インテグリティ</b>は、「定義」では、ソフトウェア製品等の品質特性としてとらえているため、攻撃防止の度合いとして定義している。同じインテグリティという用語を情報の側からとらえ、情報の全般的な整合性、また、破壊を免れている程度を表す用語として用いることもある。それについては、データ品質の「一貫性」等を参照。インテグリティには、いわゆるバッファオーバーフローの防止措置なども含まれる。</p>                     |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・ 個人情報など機密性の高いデータを特に保護するため、厳密な暗号化・復号の仕組みを設けたい。</li> <li>・ インテグリティ侵害をレビューできるようにするために、アクセスログの記録・保存の仕組み、侵害検出と通知の仕組みを用いたい。</li> <li>・ 広い意味でのインテグリティに対処するため、データベースの対応する機能を有効に使いたい。</li> <li>・ 広い意味でのインテグリティに対処するため、複数の主体のアクセスが競合する同時アクセス制御の課題が適切な技術で対処されるよう設計し評価したい。</li> <li>・ アクセスログやバックアップデータの機密保護や真正性保護の方針(長期方針を含む)を立て、実施したい。</li> </ul> |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・ 機密性に関わって暗号化実施の程度の測定量の例。</li> </ul>  |

$$\text{暗号アルゴリズムの強さ} = 1 - \frac{\text{使用リスクがあるか危殆化しているアルゴリズム数}}{\text{使用されている暗号アルゴリズム総数}}$$

主な暗号アルゴリズムには公的な評価がされている。

- ・インテグリティ実現の程度の測定量の例。

$$\text{データインテグリティ適合性} = 1 - \frac{\text{データが実際に損傷したアクセス件数}}{\text{データ損傷を防止すべきアクセス総数}}$$

|                  |  |
|------------------|--|
| <b>品質副特性名／定義</b> | <b>副特性 否認防止性 (non-repudiation)</b><br><b>4.2.6.3</b><br>事象又は行為が後になって否認されることがないように、行為又は事象が引き起こされたことを証明することができる度合い。  |
| <b>説明</b>        | 情報アクセスや情報伝達などの行為とその内容が偽って否認されないようにシステムが作られているか。<br><br><b>【詳細】</b><br>コンピュータ利用やインターネット上のサービスでは、システム利用者やビジネスのやりとりの相手などが、以前のシステム利用ややりとりの内容を否認して自己に有利な主張をすることがありえる。それを効果的に防止するには特別な仕組みが必要である。それが否認防止である。<br>否認防止には、単純にログ等でメッセージのやりとりを記録する方法や、厳密に電子署名応用によって否認できなくする方法などがある。        |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・ビジネス上のメールの送・受信はすべて電子署名付きのものに限り、署名を常に検証することで、効果的に否認防止をしたい。</li> <li>・システムのログイン記録を管理者の電子署名で保護することにより、ログの改ざんを防止したい。</li> </ul>  |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・発生イベントの否認防止測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; text-align: center; margin-top: 10px;"> <math display="block">\text{電子署名利用率} = \frac{\text{電子署名処理が行われているイベント件数}}{\text{否認防止を要するイベント件数}}</math> </div> |

|                  |   |
|------------------|---|
| <b>品質副特性名／定義</b> | <b>副特性 責任追跡性 (accountability)</b><br><span style="float: right;">4.2.6.4</span><br>実体の行為がその実体に一意的に追跡可能である度合い。   |
| <b>説明</b>        | 情報アクセスや情報伝達などの行為が記録され追跡できるか。<br><br><b>【詳細】</b><br>アクセス制御の一つの側面として、この責任追跡性 (accountability) は最近特に重要となってきた。各種のインシデントが起こった場合に、その原因の特定と、品質上の問題点の抽出、再発防止策の策定といった事柄が求められる。ここでは、ログ等の追跡 (traceability) にとどまらず、真正性がある行為者にたどり着けることを意図している。<br>(注) 英語の accountability という用語は、もともとは会計的な説明責任という意味の用語だが、今では広く社会的な説明と原因究明ができるようにする仕組みを指して用いられている。   |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・システム停止などの事故に備えることが必要なログ記録では、発生したイベントが確実に(システム停止にいたっても)記録されるように、ソフトウェア／ハードウェアを選択・設計したい。</li> <li>・セキュリティを重視するシステム内のリソースへのアクセスは漏らさずすべて記録(誰が、何に、いつ、どうアクセスしたか)したい。</li> </ul>   |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・アクセスログの網羅性を問題とする測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; margin: 10px 0; text-align: center;"> <math display="block">\text{アクセス監査性} = \frac{\text{システムログに記録されているアクセス数}}{\text{実際に発生しているアクセス数}}</math> </div> <p><b>【品質目標設定例】</b><br/>         例えば「ログインと主要な外部記憶・周辺機器・ネットワークアクセス、データベースアクセスはすべて記録し、そのログは RAID ディスクを経由してクラウド上に永久に保存する」という要求により、責任追跡性の実施目標とする。</p> |

|                  |  |
|------------------|--|
| <b>品質副特性名／定義</b> | <b>副特性 真正性 (authenticity)</b><br><p style="text-align: right;"><b>4.2.6.5</b></p> ある主体又は資源の同一性が主張したとおりであることを証明できる度合い。  |
| <b>説明</b>        | いわゆるアイデンティティを表明した主体が、本人・本物であることを、証明により確認しているか。<br><br><b>【詳細】</b><br>定義の中にある「同一性」とはアイデンティティ identity の訳語で、多くの場合は通常 ID と呼ばれるものに当たる。<br>ここでは、人の identity のほかに、メッセージの内容等が「主張した通りである」という、電子署名／電子証明書の目指す品質特性も含まれている。<br>なお、関連する authentication という英語は主体認証などと訳されることもある。  |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・ Web アプリケーションの作成において、利用者認証方式を厳密にし、またセッション乗っ取りを防止したい。</li> <li>・ アプリケーションが独自に利用者パスワード管理をするときに、パスワード自体の機密性管理をどのような手段でどの程度実施すべきか、現在の攻撃の水準や標準的な防護技術の状況と照らし合わせて明確にしたい。</li> <li>・ ネットバンキングやネットショッピングにおいて、事業者から利用者への通知は、電子署名などにより真正のものであることを保証する方針としたい。</li> </ul>   |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・ 真正性がうまく手順として実現しているかどうかの測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <math display="block">\text{真正性手順適合率} = \frac{\text{正しく実現されている真正性確認手順件数}}{\text{仕様中で真正性確認手順が必要な件数}}</math> </div> <p><b>【品質目標設定例】</b><br/>         真正性確保に使われる電子証明書にも証明の度合いなどの種別があるため、リスク分析や国際的な規制の状況を考慮して適正な証明書による真正性の検証を行うことが品質目標設定の一例となる。</p> |

|                      |   |
|----------------------|---|
| <b>品質特性名<br/>／定義</b> | <b>保守性 (maintainability)</b> <p style="text-align: right;"><b>4.2.7</b></p> 意図した保守者によって、製品又はシステムが修正することができる有効性及び効率性の度合い。  |
| <b>説明</b>            | システム保守・修正が容易か。 <p><b>【詳細】</b></p> システム／ソフトウェアは多くの場合、リリース後も修正することが必要となる。その時に、その修正の状況に応じて、ソースプログラムが修正しやすいか、過去のプログラムの構造などが解析しやすいか、状況変化に合わせたアドオンやアップデートがしやすいかなどが問題となる。<br>保守性は、システム／ソフトウェアの本体だけでなく、マニュアルや関連ドキュメントを含める。  |
| <b>ニーズ</b>           | <ul style="list-style-type: none"> <li>・ 長期にわたり、いろいろな開発要員が対象のシステムを修正するために、高い保守性を持ちたい。</li> <li>・ 運用時の不具合発生を予防的に、また効果的に発見し、修正作業を容易にするために、モニタリング機能を持ちたい(ソフトウェアの振舞い、会話的な操作、データ入出力、実行時の性能、重大イベントの兆候、ログ中の要注意事項発生、業務データとしての健全性など)。</li> <li>・ システム／ソフトウェアの中核部分とは別にアドオン機能を装備し、利用者が状況に合わせてアドオン部品を選択利用し、また、アップデートできるようにしたい。</li> </ul>   |
| <b>リスク</b>           | <ul style="list-style-type: none"> <li>・ システム／ソフトウェア開発チームの解散後、保守担当による修正作業がうまくできなかった。</li> <li>・ システム／ソフトウェアが巨大で、全体を把握している要員がいない上に、効果的な技術資料もなく、事故等の対応で、不具合箇所を突き止め、効果的に修正できなかった。</li> <li>・ 毎回同じようなアプリケーションを開発しているにも関わらず、共通部品化等ができないため、いつも工数をかけてスクラッチから開発し、それによりリリースする製品の品質も低下している。</li> <li>・ 構成管理(システム全体の変更管理や、システム一式の構成要素とその要求事項の管理)を的確に行っていないため、リリース製品自体やそれに含まれる構成要素の版番号が不明であり、開発者の修正作業や利用者側での構成要素のアップデートがいつも適切に実施できない。</li> <li>・ 要求から試験に至るまで、トレーサビリティが確立されていない</li> </ul> |

め、一部の要求事項が満足されていなかったり、要求や実装の変更がどのように波及するかが不明確であったりすることが多数あった。



|                  |  |
|------------------|--|
| <b>品質副特性名／定義</b> | <b>副特性 モジュール性 (modularity)</b><br><b>4.2.7.1</b><br>一つの構成要素に対する変更が他の構成要素に与える影響が最小になるように、システム又はコンピュータプログラムが別々の構成要素から構成されている度合い。   |
| <b>説明</b>        | <p>システムやプログラムが適切にモジュールや基本構造に分割され、構成されているか、またそのモジュール間の独立性はどの程度か。</p> <p><b>【詳細】</b><br/>         モジュール性は、システム／ソフトウェアが大きいときに、いくつかのまたは多数の基本的な単位の集まりとしてシステムを構成する方がよいという考えにもとづいている。モジュール同士は情報のやりとりをして連携動作するが、情報のやりとりが複雑すぎるといろいろな場面で制御しきれないため、モジュール間の独立性や、モジュール間の情報交換の単純化が要求される。</p> <p>歴史的には、モジュール性は、不適切な go to 文や、複雑な条件分岐がソースプログラムを複雑にしたことへの反省として行われた「構造化プログラミング」や「モジュール化プログラミング」といった考え方と関連している。また、それらの延長で、「オブジェクト指向」の考え方も関連している。「構造化プログラミング」では、プログラムのロジックが複雑となった場合に、機能の一部を別の関数／サブルーチンに仕立てて外出ししたりする。オブジェクト指向では、さらに再利用性を重視して、機能面の分割だけでなく凝縮性や結合性のような性質からモジュール性を評価する。モジュール性はより詳細化した単位であるプログラムの内部構造（選択構造や繰返し構造など）について問題とすることもあるし、より大きな視点で、システム／ソフトウェアのアーキテクチャ面で考えることもある。</p> <p>このモジュール性を、各構成要素を修正した際のインパクトの面から見ると、「定義」で述べられている「変更の影響が最小になるようにする」という条件となる。</p> <p>(注)「定義」で述べている「構成要素 (component)」とは、多くの場合、システム／ソフトウェアのコンパイル単位等のいわゆるモジュールと言われるものや、機能的に密接に関連したソフトウェアの集合した単位で、システムを構成する部品となるものである。SQuaREでは、構成単位のことを、少し一般化した表現で、「別々の構造（例えば、アセンブリ又はソフトウェアモジュール）で、特別な水準での分析で考慮されたシステム内のもの」と定義している。</p> |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・システム／ソフトウェアを適切な機能単位／データ単位に分割し、プログラムの複雑性に起因する機能適合性の低下を防止したい。</li> <li>・開発に関わる多くのプログラマごとに、長大なプログラムや、複雑</li> </ul>  |

## 品質測定量

な論理を持ったプログラムなどのばらつきがあると、ソフトウェア全体の保守性が落ちるため、モジュール性に関する指針を決め、定量的に評価することが望ましい。

- ・ソフトウェアモジュール内の複雑性の測定量の例。

$$\text{システム複雑度(サイクロマチックから見た)} = \frac{\text{基準以上のサイクロマチック複雑度を持つモジュール数}}{\text{製品中のソフトウェアモジュール総数}}$$

ここで、サイクロマチック複雑度とは、プログラムのロジックの複雑さを表すもので、プログラム構造をグラフ表現したときに、グラフの頂点数(n)、辺数(e)、接続されたプログラム要素数(p)から次式で計算される。

$$e - n + 2p$$

- ・構成要素の結合度の測定量の例。

$$\text{構成要素結合度適合性} = \frac{\text{変更の他への波及が基準以下の構成要素数}}{\text{独立性が要求される構成要素数}}$$

|                  |  |
|------------------|--|
| <b>品質副特性名／定義</b> | <b>副特性 再利用性 (reusability)</b><br><p style="text-align: right;">4.2.7.2</p> 一つ以上のシステムに、又は他の資産作りに、資産を使用することができる度合い。  |
| <b>説明</b>        | ある状況に合わせて、または、汎用に作ったシステム、ソフトウェア構成要素などが、多様な状況でも利用できるようになっているか。<br><br><b>【詳細】</b><br>再利用性は、特定の目的を持って開発したシステム／ソフトウェアやその構成要素を、他の目的に転用して利用する場合にも、また、最初から多くのシステム／ソフトウェアで利用されることを目指す部品の開発の場合にも問題になる。<br>再利用部品は、多くの場合、利用できる環境に多少の制約があるが、再利用部品開発の上では、できるだけ多様な環境で利用できることが望ましいといえる。制約の例としては、稼働可能なプラットフォーム、入力データの数とデータ範囲・データ種別、時間効率と資源効率、などがある。 |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・ 開発プラットフォーム提供ベンダーの立場や、自社開発ソフトウェアの開発効率化・高品質化の立場から、共通利用部品を開発したい。</li> <li>・ 移植性の品質特性とも関連して、多くの稼働環境への移植を最初から想定し、再利用性を高めたい。</li> <li>・ システム／ソフトウェアの設計を比較的抽象的なモデルのレベルで行い、いろいろな状況でモデルを再利用し、実装はそれぞれの状況に応じて対応するというアプローチをとりたい。</li> </ul>   |
| <b>品質測定量</b>     | ・ 副特性の直接的な測定量の例。<br><br><div style="border: 1px solid black; padding: 10px; text-align: center;"> <math display="block">\text{資産再利用性} = \frac{\text{再利用可能なように設計／実装されている資産数}}{\text{システム中の総資産数}}</math> </div>  |

品質副特性  
名／定義

## 副特性 解析性 (analysability)

4.2.7.3

製品若しくはシステムの一つ以上の部分への意図した変更が製品若しくはシステムに与える影響を総合評価すること、欠陥若しくは故障の原因を診断すること、又は修正しなければならない部分を識別することが可能であることについての有効性及び効率性の度合い。

## 副特性 修正性 (modifiability)

4.2.7.4

欠陥の取込みも既存の製品品質の低下もなく、有効的に、かつ、効率的に製品又はシステムを修正することができる度合い。

説明

システムの変更や障害対応に際して、変更の影響範囲や障害原因を解析し、修正するのが容易か。

【詳細】

**解析性**は、動作時の不具合などに対する原因の理解が容易にできるという特性を表していて、プログラムコードの静的な性質も、動作時の動的な性質も含む。

解析性は、いわゆるリバースエンジニアリングの容易性について述べたものではない。正当な保守者によるシステム／ソフトウェアの修正のためには、その動作及び構造をよく理解し、一貫性をもって必要な修正を行う必要がある。モジュール性が高度であれば修正箇所が一か所または狭い範囲で済むが、一般的には修正箇所が広い箇所に波及したり、分かりにくい波及を招いたりすることがある。全体の動作及び構造がよく理解できれば、修正必要箇所を特定し、網羅するうえでの間違いを少なくすることができる。

**修正性**は、修正実施においていわゆるデグレードが生じないで、迅速・的確に修正ができることを述べている。解析性が高度であっても修正が容易であるとは限らない。修正すべき内容が膨大となったり、隠れた依存関係があることを見逃したり忘れていたりして修正漏れが発生し、デグレードが生じることがある。

オブジェクト指向プログラミングでは、制御の流れが見えにくくなり、実行時のデータの計算・変更結果がどのように波及するか見えにくくなることがある。

ニーズ

- ・ 解析性と修正性の向上のために、コーディング規約を導入したい。
- ・ 実行時の不具合の状況を分かりやすく検出するために、適切な実行ログをとり、また自己診断機能をシステム／ソフトウェアに実装したい。
- ・ 解析性と修正性をモニタリングするため、静的解析ツールで出荷

|              |   |
|--------------|---|
|              | <p>前の製品品質チェックを行い、それにもとづく開発チームへのフィードバックプロセスを実施したい。</p> <ul style="list-style-type: none"><li>・保守作業に伴うデグレードの防止のため、構成管理の厳密化やモジュール性の高度化を試みたい。</li></ul>   |
| <b>品質測定量</b> | <ul style="list-style-type: none"><li>・自己診断機能の実装に関わる測定量の例として次のものがある。</li></ul> <div data-bbox="348 347 866 475" style="border: 1px solid black; padding: 10px; text-align: center;"><math display="block">\text{診断機能有効性} = \frac{\text{原因分析に有効な診断機能数}}{\text{実装されている診断機能数}}</math></div> <p>【品質目標設定例】</p> <p>例えば、障害対応の解析性を高めるために診断機能を充実させることにし、すべてのプログラムに診断実行モードという自己診断機能の厳密実施とログ抽出を目的とした実行オプションを設けることを方針とする。</p> |

|                  |   |
|------------------|---|
| <b>品質副特性名／定義</b> | <b>副特性 試験性 (testability)</b><br><br><b>4.2.7.5</b><br>システム、製品又は構成要素について試験基準を確立することができ、その基準が満たされているかどうかを決定するために試験を実行することができる有効性及び効率性の度合い。  |
| <b>説明</b>        | ある機能やシステムの構成要素が的確に試験できるように構成されているかどうか。<br><br><b>【詳細】</b><br>試験が可能かどうか、及び試験の容易性は、内部特性としても、外部特性としても評価できる。すなわち前者では、試験実施を容易にする支援機能(モニタリング機能、プロファイリング機能等)が充実しているかどうかの評価などが含まれる。後者に関する測定では、試験従事者や利用者(受入れ試験等を担うとした場合の利用者)が実地にどの程度の努力を要したのかの測定が行われる。試験実施結果の信頼性(テストカバレッジなど)は、成熟性などの品質特性で扱われる。<br>試験性には、ライフサイクルの各段階でのものがある。要求の仕様化時の試験性は、モデルベース試験、レビューやプロトタイプングと関係がある。製品出荷時の試験性は、受入れ試験と関係がある。運用時の試験性は、運用システム動作のモニタリングと関係がある。<br>試験が進行したのちの修正に対応する試験をリグレッションテスト(回帰試験)という。リグレッションテストについてもこの「試験性」が重要となる。 |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・ 試験工数の見積もりは、プロジェクトの工期管理の上で大きな留意事項となることがある。そのときに試験性を考慮した試験工数の短縮可能性を検討したい。</li> <li>・ 受入れ試験実施において、効率的で的確な顧客側の試験実施は、ビジネス上も、技術上も有効な効果を生む可能性があり、試験の効率性を検討したい。</li> <li>・ 運用時の不具合の修正に伴う高い試験性(リグレッションテストの試験性)は、解析性ととも、全般的な保守性の向上に大きく役立つため、その効率化を検討したい。</li> </ul>  |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・ 他システムに依存する試験実施環境に関わる試験性の測定量の例として次のものがある。</li> </ul>  |

自立試験可能性 =

$$\frac{\text{依存していてもスタッフ等で実施できる試験数}}{\text{他システムに依存している試験項目数}}$$

|                      |   |
|----------------------|---|
| <b>品質特性名<br/>／定義</b> | <b>移植性 (portability)</b> <p style="text-align: right;"><b>4.2.8</b></p> 一つのハードウェア、ソフトウェア又は他の運用環境若しくは利用環境からその他の環境に、システム、製品又は構成要素を移すことができる有効性及び効率性の度合い。   |
| <b>説明</b>            | 異なる環境にシステムを移すことが、どの程度容易にできるか。<br><br><b>【詳細】</b><br>移植性は、再利用性とも関係するが、製品そのものというより、異なる環境へシステム／ソフトウェアを移す作業、または新規インストール、アップデートインストールする作業に焦点を当てている。<br>移植で問題となる「他の環境」とは、異なるホストや異なる OS、異なる運用設定のほかに異なる部署・組織等を含む。<br>移植性も、試験性と同様に、移植の作業を容易にする支援機能が充実しているかという側面と、実際に移植を経験した際に作業が容易だったかという実情評価の側面の両面で評価できる。   |
| <b>ニーズ</b>           | <ul style="list-style-type: none"> <li>・ 様々な環境での利用が可能なことをうたうソフトウェアでも、実際にインストールしようとする実務的障害に出会うことがある。利用者から見た場合には、スムーズなインストールが期待されるので、その評価を重視したい。</li> <li>・ 初期インストール手順だけでなく、アップデートインストール手順や、アンインストール手順についても容易性を確認・評価したい。</li> <li>・ 異なるプラットフォームへのソフトウェアの移植を容易にするため、様々な仮想化技術を積極的に利用したい。</li> </ul>   |
| <b>リスク</b>           | <ul style="list-style-type: none"> <li>・ 製品を動作させ、インストールを成功させるために必要なプラットフォームやリソースがリリースに際して明示されていなかった。このリソースには、CPU 種別・性能、メモリ量、ハードディスク容量、周辺機器、ネットワーク環境などを含む。</li> <li>・ インストールに必要な準備作業や実施に要する時間が、インストール開始以前に、分かりやすく明示されていなかった。</li> <li>・ アプリケーションのアンインストール時に、利用者が作成したデータやパラメータは保存されたままになることがあり、そのことが利用者の不都合となることがあった(別途、利用者データ／パラメータの削除が必要だった)。</li> <li>・ アプリケーションのインストールに必要なアクセス権限等が明示されていなかった。また、インストールがシステム管理者一人分しか行えず一般利用者として利用できないためセキュリティ方針に沿わなかった。</li> </ul> |



|                         |   |
|-------------------------|---|
| <p><b>品質副特性名／定義</b></p> | <p><b>副特性 適応性 (adaptability)</b><br/>4.2.8.1<br/>異なる又は進化していくハードウェア、ソフトウェア又は他の運用環境若しくは利用環境に、製品又はシステムが適応できる有効性及び効率性の度合い。</p> <p><b>副特性 設置性 (installability)</b><br/>4.2.8.2<br/>明示された環境において、製品又はシステムをうまく設置及び／又は削除できる有効性及び効率性の度合い。</p>   |
| <p><b>説明</b></p>        | <p>利用環境の変化に対応して、システムを移動させたり、(再)インストールさせたりするのがどの程度容易か。</p> <p><b>【詳細】</b><br/>これらの副特性は、システム／ソフトウェアをいろいろな利用者の環境に持ち込んだ際に適切に稼働できるかどうかを問題としている。</p> <p><b>適応性</b>では、環境が多用であったり、変化していったりする時に、それに追従できるかを問題としている。</p> <p><b>設置性</b>は、稼働環境にインストール／アンインストールする手順が容易かどうか、思い通りのインストールができるかどうかを問題としている。</p>                               |
| <p><b>ニーズ</b></p>       | <ul style="list-style-type: none"> <li>・製品が多様な環境への対応を目標としている場合には、その対応範囲を明確にしたい。</li> <li>・製品が一般利用者自身の手によるインストールやアンインストールを想定している場合、その手順を明確にし、また利用者の個別の利用環境の状況や、考えうるエラー操作等に対して柔軟に対応できるか明確にしたい。</li> </ul>   |
| <p><b>品質測定量</b></p>     | <ul style="list-style-type: none"> <li>・n 箇所での平均インストール時間を評価する測定量の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <math display="block">\text{インストール時間効率性} = 1/n * \frac{\sum_{i=1,n} i \text{ 番目のインストールの実際の時間}}{i \text{ 番目のインストールの期待時間}}</math> </div> <p>期待時間は、過去のデータや業界標準により決まる。</p> |

|                  |   |
|------------------|---|
| <b>品質副特性名／定義</b> | <b>副特性 置換性 (replaceability)</b><br><p style="text-align: right;">4.2.8.3</p> 同じ環境において、製品が同じ目的の別の明示された製品と置き換えることができる度合い。   |
| <b>説明</b>        | ある製品を、同じ目的の別のバージョンや別の製品と容易に置き換えることができるか。<br><br><b>【詳細】</b><br>置換性は、システム／ソフトウェアの稼働環境／操作者が継続的に利用できることを評価する。システム／ソフトウェアがバージョンアップしたり、他社の類似製品と交換したりする場合などに問題となる。  |
| <b>ニーズ</b>       | <ul style="list-style-type: none"> <li>・多くの既存利用者がいる製品のバージョンアップが必要になったときに、新機能や新インタフェースを積極的に導入したいが、利用者が新製品への移行に困難を感じることは回避したいとの立場で、置換性の程度を評価したい。</li> <li>・同様に、製品の機能向上に伴ってデータファイルの形式を変更したいときに、データファイルのコンバータ機能を新旧両製品に持たせる(追加する)など、置換性に配慮したい。</li> </ul>  |
| <b>品質測定量</b>     | <ul style="list-style-type: none"> <li>・データファイルの後方置換性の評価の例。</li> </ul> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <math display="block">\frac{\text{データの再利用性/インポート能力} = \text{従来同様に引続き使える件数}}{\text{旧ソフトウェア中データで引続き使われる件数}}</math> </div> <p>ここで、後方置換性とは、新製品が旧製品の機能やデータをカバーできることをいう。前方置換はその逆である。</p> |

# 第7章

## データ品質 品質特性表

以下にデータ品質の各特性の内容をそれぞれ表の形で示す。この章では、特性表は、「説明」を中心に簡略化して示す。

<本章のまとめ方について>

データ品質全体の性格づけを、特性表に似せて最初にまとめた。以下の分類でデータ品質特性を示している。

◆固有のデータ品質特性

正確性、完全性、一貫性、信ぴょう性、最新性

◆固有及びシステム依存のデータ品質特性

アクセシビリティ、標準適合性、機密性、効率性、精度、追跡可能性、理解性

◆システム依存のデータ品質特性

可用性、移植性、回復性

(用語の説明)

「固有のデータ品質」とは、データを取り扱うシステム機能とは独立して存在する、データの品質特性である。

「固有及びシステム依存のデータ品質特性」とは、それら両者の性質を両方とも持つデータの品質特性である。

また、「システム依存のデータ品質特性」とは、システム機能に関連し、システムと独立でないデータの品質特性である。

なお、データ品質の特性項目と、利用時の品質特性(副特性)や製品品質特性(副特性)の名称が同じ場合があるが、名称が同じでも、多くの場合、同じ事柄をデータの利用を主眼として問題にしているか、システムの果たすべき働きを主眼にして問題にしているか、という基本的な違いがある。それに従って、具体的な品質特性の内容も異なっている。

|        |  |
|--------|--|
| 品質モデル名 | <h2>データ品質 (data quality)</h2> <p style="text-align: right;">5.2</p> <p>指定された状況で使用するとき、明示されたニーズ及び暗黙のニーズをデータの特性が満足する度合い。</p>   |
| 説明     | <p>(この項目では、データの品質に関する事項をまとめてとりあげる。)</p> <p><b>【詳細】</b></p> <p>データ品質は次の理由から重要だとされている。</p> <ol style="list-style-type: none"> <li>a) データ生成プロセスの品質が分からない組織、または品質が劣っている組織からデータを獲得する場合がある。</li> <li>b) 不満足な情報、使用できない結果、また顧客の不満を生み出す不完全なデータがある。</li> <li>c) 様々な所有者／利用者のところにデータが分散していることがある。</li> <li>d) データ自体に意味の曖昧性があったり、あるデータと他の関係するデータとの間に一貫性が欠如していたりすることがある。</li> <li>e) レガシーなコンピュータシステムと、分散オープンシステムとの共存において、データや通信の考え方が異なる場合がある。</li> <li>f) Web サイトのシステムのように頻繁なデータ交換/統合が重要な場合がある。</li> </ol> |
| ニーズ    | <ul style="list-style-type: none"> <li>・ データ品質特性は、次のような目的に使用される。 <ul style="list-style-type: none"> <li>— データの生成、取得、統合のプロセスで、データ品質要求事項を定義し、評価する。</li> <li>— データの品質保証基準を決定する。</li> <li>— データのリエンジニアリング、総合評価及び改善に役立てる。</li> <li>— データの標準適合性を法律／契約上の要求事項で評価する。</li> </ul> </li> <li>・ データ品質が悪いと判断されるような場合(セキュリティ上の攻撃者が悪意あるデータを用意するような場合を含む)には、それに対応できるソフトウェア品質を考慮する必要がある。</li> <li>・ Web サイト開発・運用等のコンテンツプロバイダも、データ品質に対するニーズを持っている。</li> </ul>   |
| リスク    | <ul style="list-style-type: none"> <li>・ 入力データの品質を見誤ったため、品質の悪いデータでシステムが誤動作した。</li> <li>・ セキュリティ上の攻撃者が仕込んだ攻撃データを画像データ、文書データとして読み込み、セキュリティ事故を起こした。</li> <li>・ 情報の受け渡しをするシステムにおいて、受け渡し情報に含まれるデータの精度や、プロトコルの厳密性についての考え方が異なるために情報交換が実施できなかった。</li> </ul>  |

- ・ ID 情報が適切に管理されていないために、名寄せによる適切なサービスやアクセス管理が実施できなかった。
- ・ 測定条件の異なるデータが、データ名称が同じまたは類似という理由だけで統合され、統計的な処理対象としての意味を失った。
- ・ データベース上に記録されている個別項目の計算上の集計値が、集計値として別途記録されている値と整合しなかった。
- ・ データが改ざんされ、デジタル的なメッセージの本体と電子署名データが整合しなかった。
- ・ 文字コードや語彙の整合性が取れていなかったり、いわゆる外字が多用されていたりするため、他のシステムでの利用が困難になった。

|              |   |
|--------------|---|
| 品質特性名<br>／定義 | <h2>正確性 (accuracy)</h2> <p style="text-align: right;">固有のデータ品質 5.3.1.1</p> <p>特定の利用状況において、意図した概念又は事象の属性の真の値を正しく表現する属性をデータがもつ度合い。構文上の正確性、意味的な正確性という二つの主要な局面をもつ。</p>  |
| 説明           | <p>意図した概念や事象を、どの程度正しく、表現できているか。</p> <p><b>【詳細】</b><br/>日常用語の意味で、データが正しいかどうかを示している品質特性である。その内容としては、例えば、次の例のような2つの場合がある。</p> <p>(1) 姓と名を記入する2つの欄があるときに姓と名が反対に記入されている。または、ハイフン抜きに郵便番号を記載すると決まっている欄にハイフン付きで記入されているといった不正確さ</p> <p>(2) そもそも名前が違う、番号が違うといった不正確さ</p> |

|              |   |
|--------------|---|
| 品質特性名<br>／定義 | <h2>完全性 (completeness)</h2> <p style="text-align: right;">固有のデータ品質 5.3.1.2</p> <p>実体に関連する対象データが、特定の利用状況において、全ての期待された属性及び関係する実体インスタンスに対する値をもつ度合い。</p>  |
| 説明           | <p>対象が複数のものの集合であるときに、どの程度その集合の全体のデータをカバーできているか。</p> <p><b>【詳細】</b><br/>典型的には名簿データや、各種一覧表等のデータにおいて必要なデータが揃っているかどうかを問題としている。調査結果のデータ等では、未調査データや、回答が得られなかったデータ等があり、完全性をよくすることがデータの有用性を高める。</p> |

|              |  |
|--------------|--|
| 品質特性名<br>／定義 | <h3>一貫性 (consistency)</h3> <p>固有のデータ品質 5.3.1.3</p> <p>特定の利用状況において、矛盾がないという属性及び他のデータと首尾一貫しているという属性をデータがもつ度合い。それは、一つの実体に関するデータ相互間、又は同等の実体に対する類似のデータをまたがったデータ同士間の、いずれか一方又は両方となる場合がある。</p>   |
| 説明           | <p>対象データやそれを用いた統計値等が複数記録されているときに、それらはどの程度相互に矛盾がないか。</p> <p><b>【詳細】</b></p> <p>本来同じデータが、データ集合の複数の場所に違った形で記録されているときに、一方がアップデートされても他方のアップデートが遅れたり、行われなかったりすることがある。また違った角度から収集されたり計算されたりしたデータが本来は一致しているべきなのに異なっていたり矛盾していたりすることがある。例えば、2015年に調査された年齢 25 歳の人の生年月日が 1970 年 1 月 1 日だった場合は一貫性がないということになる。</p> |

|              |  |
|--------------|--|
| 品質特性名<br>／定義 | <h3>信ぴょう(憑)性 (credibility)</h3> <p>固有のデータ品質 5.3.1.4</p> <p>特定の利用状況において、利用者によって真(実)で信頼できるとみなされる属性をデータがもつ度合い。</p>  |
| 説明           | <p>データの内容の正しさがどの程度信頼できるか。</p> <p><b>【詳細】</b></p> <p>データの表す内容が信頼するに足るかどうかを示している品質特性だが、正確さとは異なり、何らかの信頼性に関する保証があるかどうかといった点を問題としている。データの収集方法や検証方法について信頼性に関わる情報が提供されている必要がある。信ぴょう性の概念は、データの真正性 (authenticity) の観点を含むとされている。すなわちデータ提供者／保証者の身元情報や、保証の手法の正しさを含むとされる。</p> |



|              |   |
|--------------|---|
| 品質特性名<br>／定義 | <b>最新性 (currentness)</b><br>固有のデータ品質 5.3.1.5<br>特定の利用状況において、データが最新の値である属性をもつ度合い。  |
| 説明           | システムの持つ／表示するデータが、どの程度、現実の変化に追従できているか。<br><br><b>【詳細】</b><br>交通機関の運行に関する時刻データのように、予定と現実が異なることがある。また、株価のように刻々と変化していくデータもある。システムが変化する現実のデータをタイムリーに収集・表示することをデータ側の固有属性として表している。 |

|              |  |
|--------------|--|
| 品質特性名<br>／定義 | <b>アクセシビリティ (accessibility)</b><br>固有の及びシステム依存のデータ品質 5.3.2.1<br>特に、幾つかの障害が原因で、支援技術又は特別な機器構成を必要とする人々が、特定の利用状況において、データにアクセスできる度合い。   |
| 説明           | 障がいのある人々が、どの程度、そのデータにアクセスできるか。<br><br><b>【詳細】</b><br>データがその特定の表現形式(画面上の文字、スピーカーから流れる音声等)に関わらず、意図した情報そのものとして、確実に、伝えられることを問題としている。一つの情報が複数のチャネル(文字、画像、音声、しぐさ、異なる言語などを含む)で伝えられれば、一般にアクセシビリティは高いといえる。利用者が実際に利用できるかどうかは、特定の装置(スピーカー等)をシステムが装備しているかどうかにも依存する。<br>この特性は、製品品質特性のアクセシビリティに関連する。 |

|                      |  |
|----------------------|--|
| <b>品質特性名<br/>／定義</b> | <b>標準適合性 (compliance)</b><br>固有の及びシステム依存のデータ品質 5.3.2.2<br>特定の利用状況において、データ品質に関係する、規格、協定又は規範、及び類似の規則を遵守する属性をデータがもつ度合い。   |
| <b>説明</b>            | データに関する法令や各種の基準にどの程度適合しているか。<br><br><b>【詳細】</b><br>個人データや金融商品(株式等)に関わる情報(データ)は、法令の規制を受ける。あるデータが法令違反であるかどうかは、例えばそのデータが違法に取得されたものであれば、データ固有の品質として違法であるといえるが、厳重に管理すべきデータがシステムのずさんな管理機能により機密漏えいしたのであれば、システムの機能に依存して法令違反となったといえることができる。 |

|                      |  |
|----------------------|--|
| <b>品質特性名<br/>／定義</b> | <b>機密性 (confidentiality)</b><br>固有の及びシステム依存のデータ品質 5.3.2.3<br>特定の利用状況において、承認された利用者によってだけ利用でき、解釈できることを保証する属性をデータがもつ度合い。                |
| <b>説明</b>            | データ利用が、どの程度、正当な利用者によりのみ限られているか。<br><br><b>【詳細】</b><br>データの機密性は、例えばデータが暗号化されているといった状況を想定している。<br>この特性のシステム依存の側面は、製品品質モデルの機密性副特性と関連する。 |

|              |  |
|--------------|--|
| 品質特性名<br>／定義 | <h3>効率性 (efficiency)</h3> <p>固有の及びシステム依存のデータ品質 5.3.2.4<br/>         特定の利用状況において、適切な量及び種類の資源を使用することによって処理することができ、期待された水準の性能を提供できる属性をデータがもつ度合い。</p>  |
| 説明           | <p>データの取扱いにおいて、システム上のリソース(メモリ等)をどの程度効率的に利用しているか。</p> <p><b>【詳細】</b><br/>         同じことを表すにも、冗長な表し方と、効率的な表し方がある。冗長性をあえて大きくして、理解性、移植性、回復性を上げたりすることもあるが、不必要な冗長性は、処理時間や消費するリソースの量を過大なものとし、システムの全般的な効率性を下げる要因となる。また、データベース等が一定のデータを管理するために用いる管理データのオーバーヘッドがどの程度かなどもデータ効率性の問題となる。</p> |

|              |  |
|--------------|--|
| 品質特性名<br>／定義 | <h3>精度 (precision)</h3> <p>固有の及びシステム依存のデータ品質 5.3.2.5<br/>         正確な属性、又は特定の利用状況において弁別を提供する属性をデータがもつ度合い。</p>  |
| 説明           | <p>データはどの程度精密か。</p> <p><b>【詳細】</b><br/>         データは誤差を伴うものとみた場合に、その誤差の想定される幅や、表示されている有効数字の桁数が問題となる。システム依存の面でいえば、システムがどの程度のデータ精度を提供できるかは、製品品質モデルの機能正確性と関連する。</p> |

|                      |   |
|----------------------|---|
| <b>品質特性名<br/>／定義</b> | <b>追跡可能性 (traceability)</b><br>固有の及びシステム依存のデータ品質 5.3.2.6<br>特定の利用状況において、データへのアクセス及びデータに実施された変更の監査証跡を提供する属性をデータがもつ度合い。   |
| <b>説明</b>            | <p>データへのアクセス事象やデータの変更事象をどの程度まで、また容易に追跡できるか。</p> <p><b>【詳細】</b><br/>         あるデータに着目した場合に、そのデータがどの程度の信頼性や機密性が保たれているかといった観点は、そのデータがどの程度アクセス制御されているのかに関連する。この特性は、製品品質モデルの責任追跡性副特性と関連するが、ここでは、どの範囲のデータ値、データ種類が(あるいはあらゆるデータが)きちんと追跡されているかどうかという観点から追跡性を問題としている。</p> |

|                      |   |
|----------------------|---|
| <b>品質特性名<br/>／定義</b> | <b>理解性 (understandability)</b><br>固有の及びシステム依存のデータ品質 5.3.2.7<br>利用者がデータを読み、説明することができる属性で、特定の利用状況において、適切な言語、シンボル及び単位で表現された属性をデータがもつ度合い。               |
| <b>説明</b>            | <p>データの内容がどの程度容易に判読・理解できるか。</p> <p><b>【詳細】</b><br/>         利用者にとってシステムが持ち、表示するデータ、あるいは入力操作に伴うデータが容易に理解できるかどうかを問題としている。この特性は製品品質モデルの使用性特性と関連する。</p> |

|              |   |
|--------------|---|
| 品質特性名<br>／定義 | <b>可用性 (availability)</b><br>システム依存のデータ品質 5.3.3.1<br>特定の利用状況において、承認された利用者及び／又はアプリケーションがデータを検索できる属性をデータがもつ度合い。   |
| 説明           | データを利用したいときにどの程度の割合で利用できるか。<br><br><b>【詳細】</b><br>可用性の課題には、データが保全されていてすぐ使える状態となっているといった状況のほかに、同時アクセスがどの程度スムーズに可能となるか(待ち時間が長くないか)、夜間や週末、年末年始、定期点検日といった特定の日時にどの程度使えるかといったことも含まれる。<br>この特性は、製品品質モデルの信頼性(特に可用性)と関連する。 |

|              |  |
|--------------|--|
| 品質特性名<br>／定義 | <b>移植性 (portability)</b><br>システム依存のデータ品質 5.3.3.2<br>特定の利用状況において、既存の品質を維持しながら、データを一つのシステムから他のシステムに実装したり、置き換えたり、移動したりできる属性をデータがもつ度合い。  |
| 説明           | データが、どの程度、プラットフォームや組織を越えて実装したり、移動したりできるか。<br><br><b>【詳細】</b><br>データは、オフラインやオンラインの各種手段によりシステム間で共有されたり交換されたりする。そのことがどの程度スムーズに実現できるかを問題としている。レガシーなシステムでは、文字コードの相違や外字の取扱い等から移植性に問題が生じる例が多数あった。 |

|                      |  |
|----------------------|--|
| <b>品質特性名<br/>／定義</b> | <b>回復性 (recoverability)</b><br>システム依存のデータ品質 5.3.3.3<br>特定の利用状況において、故障発生の場合でさえ、明示された水準の操作及び品質を継続し、維持することを可能にする属性をデータがもつ度合い。   |
| <b>説明</b>            | システムにトラブルが生じたとき、どの程度、本来のデータを回復できるか。<br><br>【詳細】<br>データが何らかの理由で破壊されたり喪失したりした際にも、当初の目的を果たせる状態に戻れることを求めている。<br>この特性は、データにエラー修正コードのような冗長性を持たせたり、RAID のような装置を利用したり、またデータバックアップをとることによって達成される。データ管理の立場からは、どの範囲のデータがどのような手段で回復性を保持できているかどうかの問題となる。また、回復に要する平均時間や、トラブル発生前のどの時点の状態に戻れるのかも問題となる。 |

## 参考資料

---

(第一章末の「主な役に立つ参考文献」も参照)

- [1] **JIS X 25000:2010** ソフトウェア製品の品質要求及び評価 (SQuaRE) – SQuaRE の指針
  - ◆ 対応国際規格： **ISO/IEC 25000:2005** , *Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE*
- [2] **JIS X 25001:2012** ソフトウェア製品の品質要求及び評価 (SQuaRE) – 計画及び管理
  - ◆ 対応国際規格： **ISO/IEC 25001:2007** , *Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Planning and management*
- [3] **JIS X 25010:2013** ソフトウェア製品の品質要求及び評価 (SQuaRE) – システム及びソフトウェア品質管理モデル
  - ◆ 対応国際規格： **ISO/IEC 25010 : 2011**, *Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE) – System and software quality model*
- [4] **JIS X 25012:2013** ソフトウェア製品の品質要求及び評価 (SQuaRE) – データ品質モデル
  - ◆ 対応国際規格： **ISO/IEC 25012:2008** , *Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Data quality model*
- [5] **ISO/IEC 25020:2007**, *Software engineering – Software product Quality Requirements and Evaluation(SQuaRE) – Measurement reference model and guide*
- [6] **JIS X 25021:2014** ソフトウェア製品の品質要求及び評価 (SQuaRE) – 品質測定量要素
  - ◆ 対応国際規格： **ISO/IEC 25021:2013** , *Software engineering – Software product Quality Requirements and Evaluation(SQuaRE) – Quality measure elements*
- [7] **ISO/IEC DIS 25022:201x** , *Software engineering – Software product Quality Requirements and Evaluation(SQuaRE) – Measurement of quality in use*

- [8] **ISO/IEC DIS 25023:201x** , *Software engineering – Software product Quality Requirements and Evaluation(SQuaRE) – Measurement of system and software product quality*
- [9] **ISO/IEC DIS 25024:201x** , *Software engineering – Software product Quality Requirements and Evaluation(SQuaRE) – Measurement of data quality*
- [10] **JIS X 25030:2012** ソフトウェア製品の品質要求及び評価 (SQuaRE) –品質要求事項
- ◆ 対応国際規格 : **ISO/IEC 25030:2007** , *Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality requirements*
- [11] **JIS X 25040:2014** システム及びソフトウェア製品の品質要求及び評価 (SQuaRE) – 評価プロセス
- ◆ 対応国際規格 : **ISO/IEC 25040:2011**, *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Evaluation process*
- [12] **ISO/IEC 25041:2012**, *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Evaluation guide for developers and, acquirers, and independent evaluators*
- ◆ 対応国内規格 : 発行準備中 **JIS X 25041** システム及びソフトウェア製品の品質要求及び評価 (SQuaRE) –開発者, 取得者及び独立した評価者のための評価手引
- [13] **ISO/IEC 25045:2012**, *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Evaluation module for recoverability*
- [14] **JIS X 25051:2011** ソフトウェア製品の品質要求及び評価 (SQuaRE) – 商用既成 (COTS) ソフトウェア製品に対する品質要求事項及び試験に対する指示
- ◆ 対応国際規格 : **ISO/IEC 25051:2006**, *Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Requirements for quality of Commercial Off-The-Shelf (COTS) software product and instructions for testing*
  - ◆ 対応国際規格改訂版: **ISO/IEC 25051:2014**, *Software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing*



- [15] **ISO/IEC 25060:2010**, *Software engineering–Software product Quality Requirements and Evaluation (SQuaRE) – Common Industry Format (CIF) for usability: General framework for usability-related information*
- [16] **ISO/IEC 25062:2006**, *Software engineering–Software product Quality Requirements and Evaluation (SQuaRE) – Common Industry Format (CIF) for usability test reports*
- [17] **ISO/IEC 25064:2013**, *Software engineering–Software product Quality Requirements and Evaluation (SQuaRE) – Common Industry Format (CIF) for usability: User needs reports*
- [18] **JIS X 0141:2009** システム及びソフトウェア技術 – 測定プロセス  
◆ 対応国際規格 : **ISO/IEC 15939:2007**, *Systems and software engineering – Measurement process*
- [19] **JIS X 0151:1989** 流通ソフトウェアパッケージの利用者用文書及び外装表示  
◆ 対応国際規格 : **ISO 9127:1988**, *Information Processing Systems – User Documentation and Cover Information for Consumer Software Packages*
- [20] **JIS X 0160:2012** ソフトウェアライフサイクルプロセス  
◆ 対応国際規格 : **ISO/IEC 12207:2008**, *Systems and software engineering – Software life cycle processes*
- [21] **JIS X 0170:2013** システムライフサイクルプロセス  
◆ 対応国際規格 : **ISO/IEC 15288:2008**, *Systems and software engineering – System life cycle processes* (2014年に再改訂)
- [22] **JIS Q 9000:2006** 品質マネジメントシステム – 基本及び用語  
◆ 対応国際規格 : **ISO 9000:2005**, *Quality management systems – Fundamentals and vocabulary*
- [23] **JIS Z 8520:2008** 人間工学—人とシステムとのインタラクション—対話の原則  
◆ 対応国際規格 : **ISO 9241-110:2006**, *Ergonomics of human-system interaction – Part 110: Dialogue principles*
- [24] **JIS Z 8521:1999** 人間工学 – 視覚表示装置を用いるオフィス作業 – 使用性についての手引

- ◆ 対応国際規格：ISO 9241-11, *Ergonomics – Office work with visual display terminals (VDTs) – Part 11: Guidance on usability*
- [25] JIS Z 8524:1999 人間工学-視覚表示装置を用いるオフィス作業-メニュー対話
  - ◆ 対応国際規格：ISO 9241-14:1997, *Ergonomic requirements for office work with visual display terminals (VDTs) – Part 14: Menu dialogues (IDT)*
- [26] ISO 9241-210:2010, *Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems*
- [27] JIS X 0134:1999 システム及びソフトウェアに課せられたリスク抑制の完全性水準
  - ◆ 対応国際規格：ISO/IEC 15026:1998, *Information technology – System and software integrity levels* (なお、本国際規格は、2010年度より拡充・改訂中)
- [28] ISO/IEC/IEEE 24765:2010, *Systems and software engineering – Vocabulary*
- [29] JIS X 5004:1991 開放型システム間相互接続の基本参照モデル-安全保護体系
  - ◆ 対応国際規格：ISO 7498-2:1989, *Information processing systems – Part 2: Open Systems Interconnection – Basic reference model – Part 2: Security architecture*
- [30] JIS Q 13335-1:2006 情報技術-セキュリティ技術-情報通信技術セキュリティマネジメント-第1部：情報通信技術セキュリティマネジメントの概念及びモデル
  - ◆ 対応国際規格：ISO 7498-2:1989, *Information technology – Security techniques – Management of information and communications technology security – Part 1: Concepts and models for information and communications technology security management*
- [31] IEEE 1517:1999, *IEEE Standard for Information Technology – Software Life Cycle Processes – Reuse Processes*
  - (注 本国際規格は、2000年度に改訂)
- [32] 社団法人 日本情報システム・ユーザー協会編著「非機能要求仕様定義ガイドライン」(UVCプロジェクトII 2008報告書)

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  
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 | 日本ナレヅ株式会社             |
|       | 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        |
| デザイン  | 寺本 高德 |                       |

**SQQuaRE 品質モデル活用リファレンス編**

監修 東 基衛 早稲田大学 理工学術院 名誉教授

ISO/IEC 25000 SQQuaRE シリーズ プロジェクト統括エディタ

## 編者紹介

2004年10月に独立行政法人情報処理推進機構（IPA）内に設立されたソフトウェア高信頼化センター（SEC）は、エンタプライズ系ソフトウェアと組み込みソフトウェアの開発力強化に取り組むとともに、その成果を実践・検証するための実践ソフトウェア開発プロジェクトを産学官の枠組みを越えて展開している。

[所在地] 〒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  
URL <http://www.ipa.go.jp/sec/>

## 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

---

平成 27 年 5 月 29 日 1 版 1 刷発行

監修者 独立行政法人情報処理推進機構（IPA）技術本部  
ソフトウェア高信頼化センター（SEC）

発行人 松本 隆明

発行所 独立行政法人情報処理推進機構（IPA）

〒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

URL <http://www.ipa.go.jp/sec/>

©独立行政法人 情報処理推進機構 技術本部 ソフトウェア高信頼化センター 2015

---





〒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/>

定価: 本体 926 円(税別)

