

3. システムおよびソフト ウェア品質向上のための 品質測定技術

山田 淳((株) 東芝) 谷津行穂((株) シンフォーム)

和田典子(情報処理学会・情報規格調査会・SC 7/WG 6) 福住伸一(日本電気(株))

システム・ソフトウェアの品質測定 技術の概念

ソフトウェアや、ソフトウェアが組み込まれたシステムの品質について、開発後半のテスト段階から不具合やバグの件数を主とした測定をはじめ、品質を評価・改良することが通常よくある。これに対して、システムの利用時や運用開始時、テスト時には、「どのような品質が、どの程度まで良い」ことが求められているのか、品質面の要求とその測定法として仕様化するよう、開発初期に顧客や利用者の要求を分析する。そして開発前半から段階的に品質を測定・評価する。より早くから多面的に品質改良することにより、品質向上を促進できる。

また、開発コストの制約からも、顧客や利用者と開発者とが合意し、共に確認できるよう、品質要求仕様を明確にしておかなければ、品質を満足する開発は難しい。そこで、品質が「どの程度まで良い」かを定量的に表す測定法によって品質要求を詳細化し、満たせた程度を評価できるようにする。また故障に遭遇しないだけでなく、使いやすさ、処理の速さ、保守しやすさも含めた「どのような品質」を表すためには、多種多様なシステム・ソフトウェアの品質を特性で表す ISO/IEC 25010 品質モデル¹⁾を参照できる。モデルにより、品質を分類した品質(副)特性別に測定法を設計して適用する。

測定法の設計では、まず要求を明確化し、測

定・評価する品質(副)特性を選ぶ。次にシステム・ソフトウェア製品や作業成果物などから測定対象を選び、品質(副)特性に影響を与える特徴となる属性を抽出し、数量化する測定方法を定める。そして測定タイミング、粒度、範囲などの制約を選び、レビューやテストで測定する詳細な方法を定める。それらを品質測定法の構成要素とし、複数の要素を組み合わせて計算し導出する(図-1)。

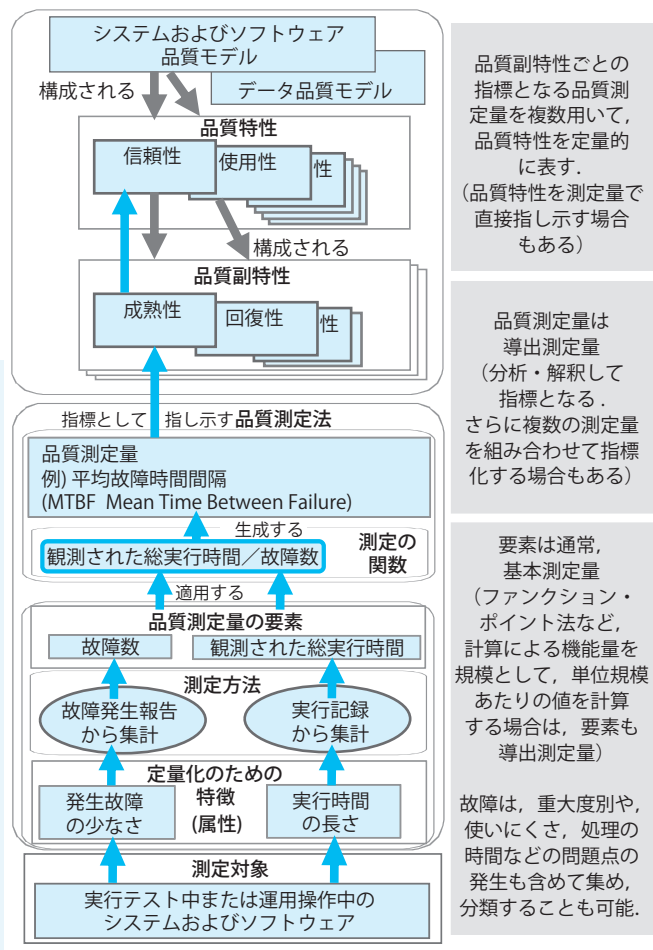


図-1 システム・ソフトウェア品質測定法の構成

このような構成で品質測定法を定義したものを品質測定量（品質メジャー／Quality Measure）と呼び、品質（副）特性そのものは直接数量化できないが、このように導出した品質測定量を、品質（副）特性を定量的に表現する指標（インディケータ／Indicator）として用いることができる。品質（副）特性ごとに、こうした指標を1つ以上用いて代表させることによって、品質を定量的に表す。

たとえば、どの程度まで信頼性の成熟性を向上できたかを知るために、実行可能なソフトウェアを測定対象とし、実行中に発生した故障件数を計数し、実行時間の合計を割って数値化する測定法を「平均故障時間間隔」として設計できる。特徴や測定の方法の詳細化には測定の目的と適用状況から、ライフサイクルプロセスや作業プロセスの特徴や段階に関連付けて、主に測定の範囲と粒度を考慮する。たとえば、利用やテストで実行した作業・機能の範囲による故障件数の違いを知る必要があれば、作業・機能別に測定する場合やテスト網羅範囲を示す情報を付加する場合がある。またテスト実行が断続的であれば、1日単位の粒度なら、実際にテストが実行された日だけを選んで計算する。また、処理ごとの実行時間を秒単位で測定するのがよい場合もある。そして、どのような期間ごとに計算して時間変化を追跡すれば、信頼性が向上する様子や変動したことを分析できるかを決める。

システム・ソフトウェア製品品質ライフサイクルと品質測定

● 品質測定法設計で品質要求定義

システム・ソフトウェアの開発過程の各段階で品質測定を適用することによって、システム・ソフトウェア製品の品質を開発の段階に沿って要求仕様化し実装し確認していくという、品質ライフサイクルプロセスを形成できる（図-2）²⁾。まず上流の要求

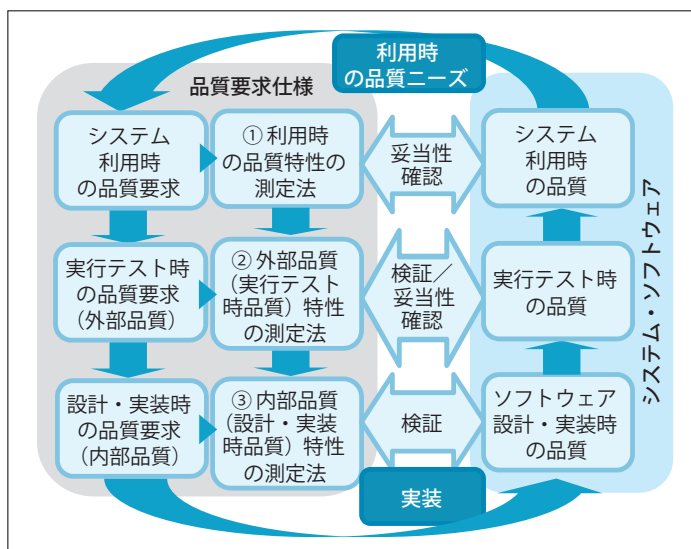


図-2 品質ライフサイクルと品質測定

定義・仕様作成段階で、顧客や利用者などステークホルダの品質のニーズや要求を引き出して品質要求を定義するとき品質測定法を用いる（図-2左）²⁾。品質要求仕様を詳細化する作業の一部として、重点的に測定する品質特性を選定し、その品質測定法を設計または選定・改造して仕様化するというように用いる。品質要求を定量的な表現の仕様として詳細化できるように、測定対象、属性と測定要素、測定環境条件、測定の計算方法、そして測定値を分析・解釈するための品質達成目標と許容可能範囲を決める。たとえば、信頼性なら、故障の重大度を区別するサービスレベルの詳細や、発生頻度の許容範囲の設定などである。

ISO/IEC 25010の品質モデルを用いると^{1), 2)}次の観点で測定法を設計できる。①利用者がソフトウェアを含むシステムを利用したときの品質を測定する「利用時の品質特性の測定法」は、利用中のシステムが利用者の行動や作業、またシステムを利用したサービスやビジネスに及ぼすリスク、効用、効率といった影響を分類した特性の属性について測定する。②「外部品質（テスト実行時の品質）特性の測定法」は、利用前のテスト時に達成するシステム・ソフトウェアの振舞いや動作の動的な特性の属性について測定する。③「内部品質（設計・実装時の品質）特性の測定法」は、システムとソフトウェアの

内部の構造や機構の特徴の静的な属性について、外部品質特性に対応付けて測定する。なお重要性を増してきた「データ品質特性の測定法」¹⁾も、システムが生成、蓄積、提供する情報データの属性を前記の①、②、③に関連させて測定する。これらの測定法は、次の章で解説する。

● 品質測定による検証・妥当性確認

設計・実装段階では品質要求定義で設計した③内部品質特性の測定法を用いて検証を行う。実行動作はさせずに、仕様との照合やチェック項目、レビュー観点別の確認項目をもとに測定する。ソフトウェアは画面遷移のレビュー、ソースコードのレビューや静的解析ツールによる測定も行える。システムコンポーネントやソースコード、システムが蓄積する情報データに作り込んだ設計・実装時の品質を、目標値・許容範囲と比較して検証することができる(図-2右下段)。

実装段階からテスト段階では、②外部品質特性の測定法を用いて検証・妥当性確認を行う。実行可能なプロトタイプも含め、システム・ソフトウェアをシミュレーションやテスト、または利用の環境で実際にテスト実行動作をさせて測定を行う。単体テスト、結合テスト、システムテスト、運用操作テストと段階的に測定できる。測定結果を目標値・許容範囲と比較して評価し、品質要求仕様を満たすかを検証し、意図した利用に供することができる品質かの妥当性確認ができる(図-2右中段)。

利用者による運用操作テストや試行運用、または運用を開始しビジネスに利用している段階では、①利用時の品質特性の測定法の目標値・許容範囲と比較して、意図した利用時の品質を満足しているか妥当性確認を行うことができる(図-2右上段)。

品質測定法の設計

● システム利用時の品質特性の測定

ユーザエクスペリエンス (User Experience : UX) は、システムの利用前、利用中、利用後、利用時

間全体の各段階での利用者の体験、と定義される³⁾。しかし品質の観点、利用者の要求との合致が重要であるため、利用時のみを扱っている。まず、システム利用者を対象とし、利用者の人間としての反応、行動、内省や作業状況を、観察や質問して測定できる。代表的な測定法の例は、作業目標を正しく満たして完了できた割合、誤操作を起こす頻度、正しく作業を完了できる時間が学習効果で短縮されていく作業効率性の時間変化である。利用者が作業を行うためにシステムを操作している最中や運用期間中の一定期間を選んで測定を行うが、運用操作テストや運用開始時に測定する場合もある。システムを操作する直接利用者だけでなく、保守者や操作しないがサービスを受ける間接利用者(たとえば駅構内で案内表示を見る一般利用者)を対象に測定する場合もある^{1), 4)}。これらは、ユーザビリティ評価と関係が深い。

ユーザビリティの測定(評価)には、チェックリストが用いられることが多い。しかし、チェック項目は、ユーザインタフェース要件が実現できているかどうかの判断だけであり、作業プロセスまで含めた全体のユーザビリティの高さや、不足のある視点を評価することはできない。そこで、作業プロセスを反映したユーザビリティの視点を4つ(効率性、エラーしにくさ(有効性)、操作の学習しやすさ、操作の記憶しやすさ)に分類し、評価対象とするシステム・製品の領域(運用監視、業務系、コンシューマ系など)に応じて、チェックリストの各項目に重み付けを行う。これにより、評価対象がユーザビリティのどの視点が十分でどの視点が不足しているかを視覚的に表現することが可能となる。図-3は運用監視システムの製品間評価例である。この図の各軸ごとの値の違いから、製品間でユーザビリティの強化ポイントの違いがあることが分かる⁵⁾。

次に、システムを利用したサービスやビジネス、ステークホルダや利用者へ与える影響についても測定する。代表的なものに経済面、健康面、環境面でのリスクを緩和できる可能性の測定がある。たとえば「システムリスク緩和度」として、「システム

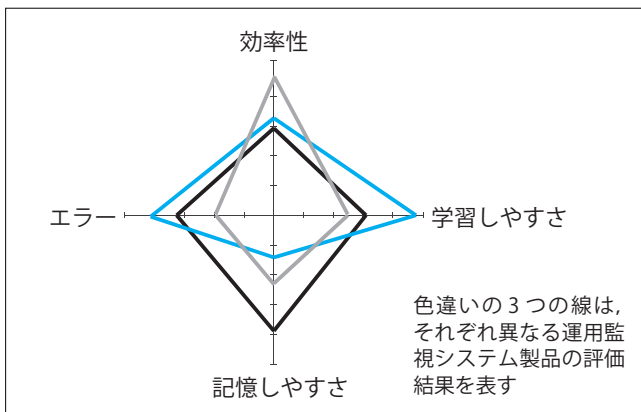


図-3 製品の差異を表すユーザビリティ視点軸

仕様に適合している度合いが品質の目標値や許容範囲に達しているかを分析することで、本格的なシステム・ソフトウェア利用を始める前に、利用に供せる品質が実現できているか評価できる。

代表的な例としては、信頼性の障害許容性の測定について、「障害通知時間」として、システム・ソフトウェア内部での障害生起時から障害を利用者に通報するまでに必要な時間が、指定した許容時間内に対し、どれくらいの時間内に収まるように分布しているかを測定できる。また、単位時間あたりの処理要求量の最大値が平均値の数倍になるといったピーク時パターンなど、故障発生につながりや

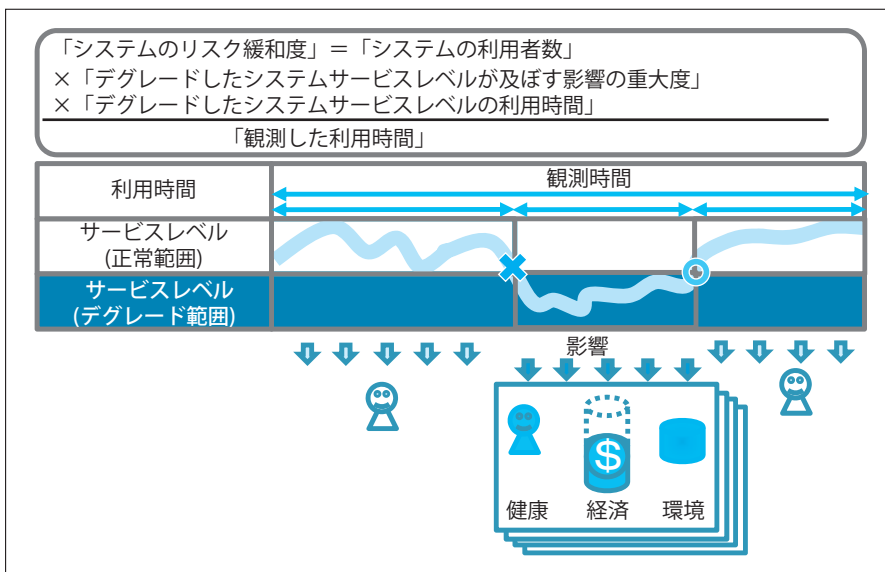


図-4 利用時の品質測定法の例

の利用者数」×「デグレードしたシステムサービスレベルが及ぼす影響の重大度」×「デグレードしたシステムサービスレベルの利用時間」／「観測した利用時間」による測定ができる(図-4)。利用中の一定期間を選定して、縮退運転やシステムダウンによってサービスレベルが低下した場合の、利用制限期間の影響を測定する。影響の重大度は、金額や健康被害、環境汚染の水準で重み付けする。

外部品質特性の測定

利用者が意図している利用方法を実現するために必要な技術的な仕様を、要求分析で外部仕様またはテスト仕様としてあらかじめ作成し、品質特性に対応付ける。そして実際にテスト実行したときのシステム・ソフトウェアの振舞い・動作がその仕様から相違している程度を測定する。テスト実行時に外部

すい障害生起パターンをテストシナリオに用いる。そして「故障回避度」として、障害があっても冗長系への切替えや修復する機構が動作し、障害が重大な故障やシステムダウンにつながるのを回避できたパターンの割合を測定できる⁴⁾。

ユーザビリティ測定については、知見を基にどのような操作方法や操作支援機能を用いて向上させるかを検討し、使用性の品質(副)特性の要求事項として、ユーザインタフェース仕様、画面仕様や運用操作テスト仕様の中であらかじめ明確にしておく。そしてテスト実行で不適合点を検出し数量化することによって測定する。

たとえば、運用操作性については、「モニタ能力度」として、システムのどの機能についての進行状況を運用操作中にモニタできるようにすべきかを仕様中で指定しておき、テスト実行してモニタできて

3. システムおよびソフトウェア品質向上のための品質測定技術

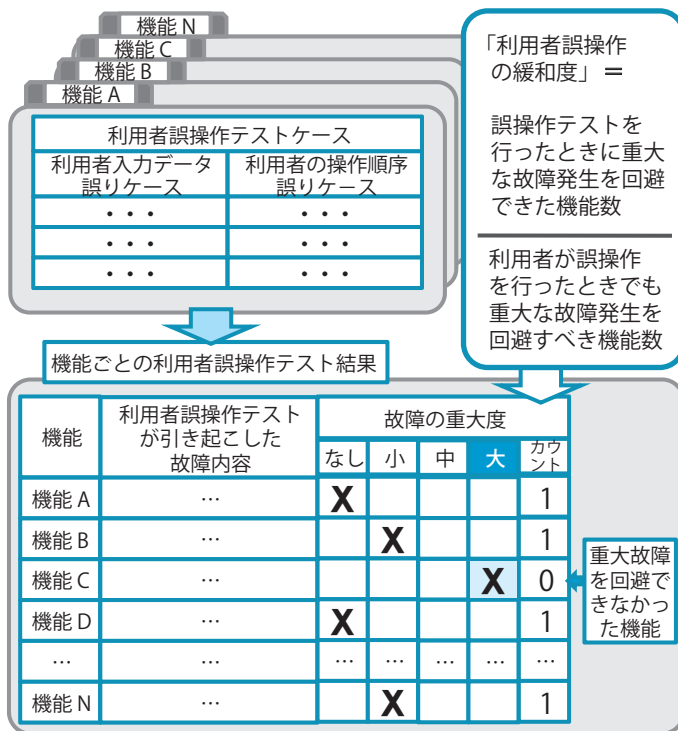


図-5 外部品質測定法の例

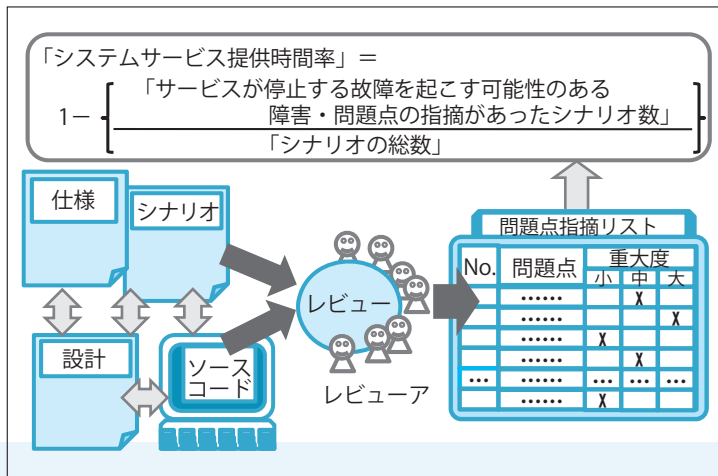


図-6 内部品質測定法の例

いる機能の割合を測定することができる。また利用者エラー防止性については、「利用者誤操作の緩和度」として、利用者が誤操作をしても重大な故障に至らないようにすべき機能をあらかじめ洗い出しておいて、誤操作テストを行ったときに回避できた機能の割合を測定できる（図-5）。

● 内部品質特性の測定

内部品質特性の測定は、「開発者視点で品質特性を測定する」と言える。システムやソフトウェアの

アーキテクトや論理設計者・プログラム作成者の視点から、システムまたはソフトウェアの外部仕様・外部品質特性を満たすように内部のコンポーネントを設計し、ソフトウェアをコーディングして実装するときの、アーキテクチャ、設計やシステムのコンポーネント、ソフトウェアのソースコードなどが備えるべき特徴を品質特性に対応付けて数値化して表す。これらは機能確認テストなどに入るまでに行われるプログラムそれ自体の設計審査やウォークスルー、コードインスペクションなどの品質確認作業を通して測定され、開発者が外部仕様から展開した機能や振舞いを実現する設計・構造になっているか、という観点から検証するときに評価する品質特性としても用いる。

代表的な例として信頼性の評価では、アベイラビリティ（可用性）の場合を見てみよう。

信頼性とは別の品質特性として捉えられる場合もあるが、ここでは広い意味で可用性を信頼性の中の品質特性と捉える品質モデルを適用する¹⁾。可用性を表す品質特性の1つとして「サービス提供時間」や「サービス提供時間率」などがあり、プログラムの実装にあたってソースコードでその実装を確認するためにコードインスペクションが適用可能である。次の

ような評価ケースを準備しておき、そのシナリオに対応してプログラムを検証していく（図-6）。

- 平日と休日のサービス提供の違いへの対処は？
- 早朝や深夜での違いへの対処は？
- システムダウンから回復するときの処理と通常の開始処理との違いに対する配慮は？
- システム規模の変化による配慮は？

など、平常時と異常時の対応の違いに注目したインスペクションの結果により、サービス時間の拡張あるいはダウンタイムの削減などに関する指摘点の改

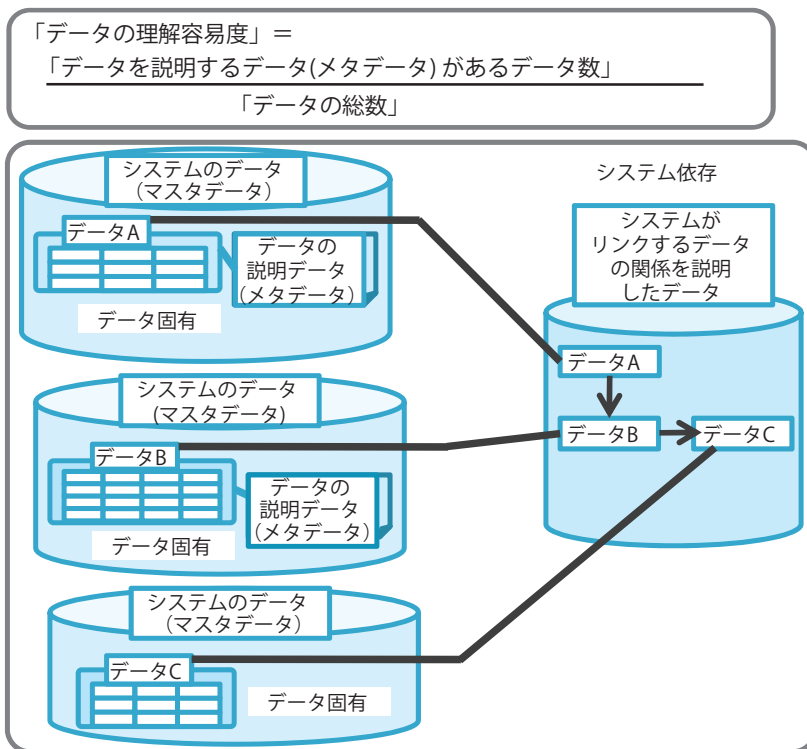


図-7 データ品質測定法の例

ムを構成するデータについて、データが作られるとき、蓄積・保存されるとき、加工されるとき、移送されるとき、利用されるとき、抹消されるとき、などのそれぞれの局面でデータの品質をその特性を用いて数値化して表す。さらに、データの特徴として、データ自身が保有している品質特性と、所属しているシステムに依存している品質特性という考え方があり、それぞれ測定法を定義できる。

データのユーザビリティ評価という面では「理解容易性」が特性例として挙げられる。マスターデータがメタデータを持つことにより、データの理解しやすさが向上する。これはデータ自身が固有に持つ品質特性だが、データ同士にリンクを張ることによりシステム依存の属性を考慮に入れた理解しやすさという品質特性となる。このように「理解容易性」の測定法としては、データ自身の特性と、システム依存の特性という両面から2種類設計でき、測定可能となることが分かる(図-7)。

善を代用特性として測定できる⁴⁾。

プログラム作成の段階で記述されたソースコードの品質特性の測定のみから、可用性をX%向上したと表現することは難しいが、このような考え方で対応すれば実装段階から評価可能である。

ユーザビリティの評価では、前出のエラーのしにくさ(リスク緩和性)を例にとると、ユーザと入力の関係から4つ(不正操作の防御、不正データ入力の防御、入力後の正当性の確認、エラー状態からの回復力)の観点が提起できる。これら4つの観点から用意したシナリオに基づいたウォークスルーなどの集団討議によってプログラム検証を行うことにより、問題指摘数や改善数などを測定することによって品質特性の相対評価が可能となることが分かる。システムに求められるユーザビリティの強調度は千差万別であり、コンシューマ系やミッションクリティカルなシステムではこれら4つの観点に求められる充足度が異なる。

● データ品質特性の測定

データ品質特性の測定法は、コンピュータシステ

また信頼性という観点では「完全性」というデータ品質特性の測定法がある。一連のデータの中で部分的に不足データがあるようなときに完全性が意味を持つ場合にあたる。たとえばお客様データベースで一部のお客様の連絡先情報が不足しているような場合である。これはシステム依存でなくデータ自身の品質特性と考えられる。

今後の課題

システムによるサービスやアウトソーシングの品質など、より広い事象や目的での品質測定・評価への必要性も増している。このため、品質特性と測定

3. システムおよびソフトウェア品質向上のための品質測定技術

法のモデルの発展、品質を開発初期から作り込めるようシステム利用時、テスト時、設計・実装時の各段階を結び付ける測定法の展開、そして測定の精度とコストとの間のトレードオフを調整して測定する際に参照できる測定事例のさらなる充実と共有が今後必要である。

参考文献

- 1) ISO/IEC 25010 : Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models (2011) (JIS X 25010, 2013).
- 2) ISO/IEC 25030 : Software Engineering - (SQuaRE) - Quality Requirements (2007) (JIS X 25030, 2012).
- 3) Roto, V., et al (Eds) : USER EXPERIENCE WHITE PAPER (2011), <http://www.allaboutux.org/uxwhitepaper>
- 4) 経済産業省：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).
- 5) 池上, 岡田, 福住：ユーザビリティ定量化手法の構築～客観的評価のためのチェックリストと支援ツールの開発, ヒューマンインタフェース学会論文誌, Vol.14, No.1, pp.101-110 (2012).

(2013年10月1日受付)

● 山田 淳 (正会員) atsu.yamada@toshiba.co.jp

(株)東芝 ソフトウェア技術センター主幹。大阪大学大学院工学研究科修了。ソフトウェア工学、プロセス・品質技術に従事。ISO/IEC JTC 1/SC 7 委員。

● 谷津行穂 (正会員) yukio.tanitsu@synform.co.jp

(株)シンフォーム グローバルIT サービス部長。芝浦工業大学工業経営学科卒業。2012年日本アイ・ビー・エム(株)退社。ソフトウェア工学・形式仕様記述・品質技術に従事。ISO/IEC JTC 1/SC 7 国内委員長。

● 和田典子 nwada0925@gmail.com

ソニー(株)を経て、本会 情報規格調査会 ISO/IEC JTC 1/SC 7/WG 6 委員。ソフトウェア工学・プロセス・品質技術に従事。

● 福住伸一 s-fukuzumi@aj.jp.nec.com

NEC 情報・ナレッジ研究所技術主幹。慶應義塾大学大学院工学研究科修士課程修了。工学博士、認定人間工学専門家、人間中心設計専門家。HI学会理事。ISO TC 159/SC 4 (HCI) 国内委員会副主査および ISO/IEC JTC 1/SC 7/WG 28 国際セクレタリー。

