

SEC journal

39

巻頭言

坂内 正夫 独立行政法人 情報通信研究機構 理事長

所長対談

英国の鉄道業界にみるヒューマンエラーと システム安全性の取り組み

ヒュー・ギブソン 英国 RSSB (鉄道安全標準化機構) テクニカルスペシャリスト
シニアヒューマンファクターズスペシャリスト

論文

プラットフォーム依存種検索によるソースコードからの プラットフォーム依存部抽出手法

岡本 周之 株式会社日立製作所 横浜研究所 大阪大学大学院情報科学研究科 / 藤原 貴之 株式会社日立製作所 横浜研究所
楠本 真二 大阪大学大学院情報科学研究科 / 岡野 浩三 大阪大学大学院情報科学研究科

特集

品質について考える

システム・ソフトウェア品質標準 SQuaRE シリーズの歴史と概要

東 基衛 早稲田大学 理工学術院 名誉教授 ISO/IEC 250nn SQuaRE シリーズ プロジェクト統括エディタ

システム及びソフトウェア品質の要求定義と評価のプロセス

込山 俊博 日本電気株式会社 ソフトウェア生産革新本部 エグゼクティブエキスパート ISO/IEC JTC1 SC7/WG6 Convener

ユーザーエクスペリエンスを指向するサービスの開発と運用に向けて

平沢 尚毅 小樽商科大学 商学部 社会情報学科 教授 ビジネス創造センター UX 研究部門長

ソフトウェア品質プロフェッショナルに求められる専門性と倫理性

鷺崎 弘宜 早稲田大学 グローバルソフトウェアエンジニアリング研究所 所長

システム及びソフトウェアの品質測定量とその測定方法

谷津 行穂 株式会社シンフォーム

小規模開発組織と品質課題 - VSE 国際規格の課題に即して -

伏見 諭 東海大学 情報教育センター 講師 情報規格調査会 SC7/WG24 主査

報告

Embedded Technology 2014 (ET2014) 出展報告

ET ロボコン 2014 観戦記

米国の NIST、MIT、SEI の 3 組織を訪問して

IPA グローバルシンポジウム 2014 開催報告

欧州におけるソフトウェア高信頼化技術の適用状況

Column

安心して使えるソフトを教えてください！



巻頭言 ……1

新しい価値の創造に向けて～情報通信における「第3のパラダイム」～

坂内 正夫 独立行政法人 情報通信研究機構 理事長

SECjournal 論文賞速報 ……3

所長対談 ……4

英国の鉄道業界にみるヒューマンエラーとシステム安全性の取り組み

ヒュー・ギブソン 英国RSSB（鉄道安全標準化機構）テクニカルスペシャリスト
シニアヒューマンファクターズスペシャリスト

論文 ……8

プラットフォーム依存種検索によるソースコードからのプラットフォーム
依存部抽出手法

岡本周之、藤原貴之、楠本真二、岡野浩三

特集：品質について考える

システム・ソフトウェア品質標準SQuaREシリーズの歴史と概要 ……18

東 基衛 早稲田大学 理工学術院 名誉教授 ISO/IEC 250nn SQuaRE シリーズ プロジェクト統括エディタ

システム及びソフトウェア品質の要求定義と評価のプロセス ……23

込山 俊博 日本電気株式会社 ソフトウェア生産革新本部 エグゼクティブエキスパート ISO/IEC JTC1 SC7/WG6 Convener

ユーザーエクスペリエンスを指向するサービスの開発と運用に向けて ……28

平沢 尚毅 小樽商科大学 商学部 社会情報学科 教授 ビジネス創造センター UX 研究部門長

ソフトウェア品質プロフェッショナルに求められる専門性と倫理性 ……32

鷲崎 弘宜 早稲田大学 グローバルソフトウェアエンジニアリング研究所 所長

システム及びソフトウェアの品質測定量とその測定方法 ……38

谷津 行穂 株式会社シンフォーム

小規模開発組織と品質課題 ……43

伏見 諭 東海大学 情報教育センター 講師 情報規格調査会 SC7/WG24 主査

報告 ……46

Embedded Technology 2014 (ET2014) 出展報告

荒川 明夫 SEC 企画グループ 主任

ET ロボコン 2014 観戦記

杉崎 真弘 SEC 研究員

米国のNIST、MIT、SEIの3組織を訪問して

中尾 昌善 SEC ソフトウェアグループ リーダー

IPA グローバルシンポジウム 2014 開催報告

八嶋 俊介 SEC システムグループ 主任

欧州におけるソフトウェア高信頼化技術の適用状況

中尾 昌善 SEC ソフトウェアグループ リーダー

Column ……61

安心して使えるソフトを教えて！

松田 晃一 IPA 顧問

編集後記 ……62

SECjournal 論文募集 / IT パスポート試験 (iパス) のご案内

新しい価値の創造に向けて ～情報通信における「第3のパラダイム」～

独立行政法人 情報通信研究機構
理事長

坂内 正夫



我が国の社会経済活動を支える基盤である情報通信は、現在、新しい時代である「第3のパラダイム」に入ってきていると、私は明確に意識しています。情報通信の歴史を振り返ると、第1のパラダイムはコンピュータをどう作り、いかにハードウェアやソフトウェア、アプリケーションをつないでいくかということがミッションでありました。第2のパラダイムではインターネット、いわゆるサイバー世界をいかに作って活用するかというフェーズでした。そして現在、サイバー世界と我々が生活している実世界とが融合し新しい価値を作り出すという新しいパラダイムに入っていると言えます。

例えば、自然災害が発生した場合、各地の被災状況を正確に把握して正しく判断して行動しなければなりません。そのためには多くの情報を迅速に収集し、それを俯瞰することで最適なコントロール方法を見出し、実世界にフィードバックする必要があります。実世界の情報収集には、多数のセンサを実社会に配置して自動的に情報を取得する方法や、スマートフォンなどの通信端末を通じて、人によって発信される情報を活用する方法などが考えられます。このようなビッグデータといわれる膨大な情報から、実世界で活用できる価値ある情報を導き出していくことが情報通信に期待される新たな役割であり、新たな価値の創造につな

がるものです。その際、一般生活への影響が大きいデータも扱いますので、情報セキュリティやプライバシーへの配慮も欠かせません。

情報通信におけるこのような第3のパラダイムは、正に我々「情報通信研究機構（NICT）」が目指す目標です。NICTでは、データの公共性の視点から、この目標を実現する情報通信技術（ICT）を「ソーシャルICT」と呼び、とくにビッグデータ系のアプリケーションを産学官連携で実証し実用化していくことを目指して、研究開発の効果的な推進体制を構築するために「ソーシャルICT推進研究センター」を立ち上げて活動を開始しました。NICTの様々な部署に分散しているリソースを俯瞰し、組織が一丸となって研究のシナジー効果を発揮して社会公共に貢献することを目指しています。また、各地域との連携についても積極的に取り組み、去年は「地域ICT研究懇談会」を開催し、地方公共団体の首長の方々からNICTの研究成果を地方公共団体に活用していただく方法や、NICTの研究に期待することなど、率直なご意見をいただくことができました。

第3のパラダイムにおいては、社会に配置されたセンサからいかに効率的にデータを収集するかが課題となります。その解決には無線通信の活用が不可欠であ

り、NICT が主導的に研究開発・標準化を推進してきた国際的な無線通信規格である「Wi-SUN」は、そのソリューションの大きな候補と考えられます。Wi-SUN は、多くの電力会社やガス会社などのメータの無線通信方式として採用される方向が決まり、一部の電力会社では Wi-SUN を実装したスマートメータ（次世代電力量計）を各家庭、企業に設置し始めています。更に農業用のワイヤレスセンサー設備として活用されつつあるなど、Wi-SUN は今後センサネットワークの基盤としての発展が期待されています。

第3のパラダイムの結果として、我々の生活における「壁」も相当に取り除かれていくことでしょう。オリンピック・パラリンピックの東京開催に関連し、総務省は ICT を用いて世界の「言葉の壁」をなくすことをミッションとした「グローバルコミュニケーション計画」を進めており、開発した技術を 2020 年までに社会実装できることを目標にしています。NICT においても、従来から研究開発を進めてきた多言語音声翻訳システムの精度をこれまで以上に高めるとともに、これまで翻訳の対象としてきた旅行会話のほかに、防災・減災、医療、ショッピングなど様々な分野での社会実証を産学官連携で推進することを計画しています。昨年、NICT 内に多数の企業からの研究者と一体となって活動する「先進的音声翻訳研究開発推進センター」を立ち上げ、このセンターを中心に産学官メンバから構成される「グローバルコミュニケーション開発推進協議会」を設けて、グローバルコミュニケーション計画を推進しています。

第3のパラダイムにおいてデータの利活用が進めば進むほど、強固に行わなければならないのが情報セキュリティ対策です。近年の政府機関や民間企業を狙ったサイバー攻撃では、技術的に高度な潜在型のマルウェアなどが使用されており、既存の技術では対処が極め

て困難な状況となっています。国を跨いだサイバー攻撃も新聞などで頻繁に報道されており、情報セキュリティの確保は国際的にも喫緊の課題となっています。NICT では国内外において協力いただける組織のネットワーク上にセンサを設置し、本来はないはずの通信（サイバー攻撃の可能性が高い通信）を観測する技術を研究開発しています。他方、企業内のイントラネットにおいては、NICT が開発したネットワークリアルタイム可視化システム NIRVANA を活用していただき、異常なトラフィックの監視に役立てていただいています。更に、地方自治体においても自ら所有するネットワークへのサイバー攻撃が大きな関心事となっており、サイバー攻撃に対して NICT から提供するアラートを多くの自治体においてもご活用いただいています。

現代社会では、交通や金融などの重要なインフラにおいても ICT の利活用により、より便利に利用できるシステムが構築されています。それらを支える大きな存在として、システムを制御するソフトウェアが位置づけられています。もし、ソフトウェアに致命的な欠陥があった場合、社会のインフラに重大な障害を発生し、経済活動への影響も広範囲に及び、人々の生活にも大きな支障をきたすことになるでしょう。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 (IPA/SEC) では社会を支える情報処理システムのソフトウェアの信頼性向上に向けて積極的に取り組んでいくことと思いますが、NICT も我が国における成長戦略の柱の一つである ICT によって新たな価値を創造すべく、ICT の研究開発にチャレンジしていきます。

【速報】 SECjournal 論文賞 受賞論文決定

2014年度 SECjournal 論文賞 受賞論文が決定しました。

今回の受賞論文は、規定により 2013年8月から2014年7月までに採録となった対象論文の中から、厳正に審査され、以下の通り決定しました。

(以下敬称略)

最優秀賞

該当なし

優秀賞 (受賞論文: 1編)

論文テーマ ソフトウェアプロダクトラインのエンタープライズ・システムへの適用と評価

筆頭著者 中村 伸裕 (大阪大学/住友電気工業株式会社)

共著者 谷本 収 (住友電工情報システム株式会社)

楠本 真二 (大阪大学)

SEC 所長賞 (受賞論文: 2編)

論文テーマ ピアレビュー有効時間比率計測によるピアレビュー会議の改善と品質改善の効果

筆頭著者 久野 倫義 (三菱電機株式会社 設計システム技術センター)

共著者 中島 毅 (三菱電機株式会社 設計システム技術センター)

松下 誠 (大阪大学 大学院情報科学研究科)

井上 克郎 (大阪大学 大学院情報科学研究科)

論文テーマ プラットフォーム依存種検索によるソースコードからのプラットフォーム依存部抽出手法

筆頭著者 岡本 周之 (株式会社日立製作所 横浜研究所/大阪大学 大学院情報科学研究科)

共著者 藤原 貴之 (株式会社日立製作所 横浜研究所)

楠本 真二 (大阪大学 大学院情報科学研究科)

岡野 浩三 (大阪大学 大学院情報科学研究科)

(詳細は次号に掲載予定です。)

英国の鉄道業界にみる ヒューマンエラーと システム安全性の取り組み

英国 RSSB（鉄道安全標準化機構）
テクニカル スペシャリスト
シニア ヒューマンファクターズ スペシャリスト
ヒュー・ギブソン



SEC 所長
松本 隆明

英国には、鉄道の安全性向上と事業活動を支援する機関「RSSB」がある。様々な事故分析や障害情報の調査から、システムやソフトウェアの安全性を研究し、業界にその知見を提供している団体だ。鉄道のように、人命にかかわる重要なインフラシステムでは、安全性の確保が最優先となる。鉄道先進国の英国では、RSSB が中心となって、その命題に体系的かつ組織的な取り組みを実践しているのだ。今回は、RSSB のシニア ヒューマンファクターズ スペシャリストであるヒュー・ギブソン氏をお招きし、本国での事例や活動内容を伺った。ヒューマンファクター（人的要因）を重視するその分析手法から、今後のシステム安全設計に必要な活動について考えてみたい。



ヒュー・ギブソン

英国バーミンガム大学にてワークデザインと人間工学の修士号、航空管制の通信エラーに関する博士号を取得。
また、原子力産業分野・鉄道分野・航空管制分野の信頼性評価に関するヒューマンエラーの定量化技術の開発に従事。
現在、英国 RSSB においてヒューマンファクターの専門家として、鉄道のインシデントから、ヒューマンエラーとその根本原因、パフォーマンス要因との関係を分析するためのデータベースの開発責任者の役割を担う。
また、鉄道における運転及び保守業務に関するヒューマンファクター研究にも取り組んでいる。

松本：今回は、RSSB（Rail Safety and Standards Board）のヒュー・ギブソンさんにお越しいただき、英国における鉄道システムの安全性確保についてお話を伺います。人の命が最優先となる鉄道システムに対し、鉄道先進国の英国では、体系的な取り組みを実践していると聞いています。社会を支えるインフラの安全性は、私たちも注力している重要な課題です。今日は、英国の事例を参考にしながら、今後重要になってくるであろうポイントを明確にしていきたいと思います。まずは、RSSB の紹介からお願いできますか？

ギブソン：RSSB は、英国の鉄道業界が提供する資金で成り立っている非営利組

織です。鉄道業界の長期的な安全計画の中、リーダーシップを発揮することを目的として、2003年に設立されました。

松本：設立のきっかけとなるような出来事があったのでしょうか？例えば大きな鉄道事故があり、業界内での気運が高まったなど……。

ギブソン：はい、大きな事故がありました。1999年にパディントン駅で発生したラドブローク・グローブという事故です。信号の見落としによって列車が急行列車に衝突し、多くの死傷者が出ました。これをきっかけに、業界の独立した安全委員会が必要だという認識に至ったのです。

松本：RSSBにはどのような参加者がいるのでしょうか。

ギブソン：約40社の鉄道関連企業です。鉄道運営事業者だけでなく、線路などを管理するインフラマネジメント企業や、そうした企業へのサプライヤーも登録しています。

松本：サプライヤーがメンバに入っているのはとても重要なことですね。鉄道サービスの提供者だけでなく、もともとの物を作って納める企業も仲間に入れていかないと、全体的な安全への取り組みが前に進みません。

ギブソン：そうですね。システムの安全を設計する段階で、鉄道を作っているメーカーの意見や情報を取り入れ

ていくのが重要だと考えています。RSSBの狙いは、そのように様々なメンバ企業への支援を提供し、安全性やパフォーマンスを高めていくことです。すべての活動では、いわゆる「バリュー・フォー・マネー (Value for Money)」——コストに対して、最も価値のあるサービスを提供することを重視しています。

RSSBが取り組んでいる活動は、主に4つです。まずは、鉄道業界にかかわるリスクの把握。次に、エンジニアリングやオペレーション、マネジメントなどの調査管理です。3つ目は、英国の鉄道業界全体に、安全性のガイドラインを指導していくこと。そして最後は、業界内のコラボレーションのサポートです。

松本：安全性を設計するためのガイドラインもRSSBが策定しているのですか？また、ガイドラインを普及するための手引きや情報提供なども行っているのでしょうか。

ギブソン：ガイドラインに関しては、やはり業界と連携しながら策定すべきだと考えているため、一方的に「これを守りなさい」といった指示は出せません。「こういった方法はどうか？」と投げかけて、業界から意見をもらいつつ作り上げていくのが大切です。そうすることで、各事業者も当事者意識を持って取り組んでくれるようになるでしょう。RSSBはそのために、公式・非公式の様々な形で、多くの指針や好例を提供しています。

ちなみに、策定されているガイドラインは多岐にわたります。例えば、プラットフォームの端から黄色い線までの距離、ヘッドライトの輝度、線路のエンジニアリングスペック——様々なものに“標準規格”が指定されているのです。

松本：それは「RAMS」のような国際標準ではなく、英国の鉄道業界における標準ですか？

ギブソン：そうですね。私たちが策定しているガイドラインは、英国の鉄道業界に向けたものです。ほかにヨーロッパが発信している標準規格もありますが、そうしたヨーロッパ全体のガイドラインに影響を与えていくのも、RSSBが担うひとつの役割だと思っています。

松本：確かに重要なことですね。また、業界内でのコラボレーションに力を注いでいるのも面白いと思いました。日本では鉄道会社がインフラやサービス、設備の提供をすべてワンストップで担当していますが、英国はそれぞれを担う企業が別々だからこそ、コラボレーションの重要性や必要性が高いのでしょうか。

ギブソン：おっしゃる通りです。英国の鉄道業界では、事業が分断化されています。効率を追求するための仕組みですが、分断されているが故に、コラボレーションの

必要性もまた高まるのです。

松本：協業に際して、インフラとサービスの提供事業者で要望が対立することはないのでしょうか。

ギブソン：確かに、ときには両者の要件や利害がぶつかり合います。ですが、そういった衝突もうまく活かしていくことが、コラボレーションの促進や価値の創出につながるのです。また、鉄道規制庁「ORR (Office of Rail Regulation)」が監督をしており、コラボレーションの結果、乗客にとってお金に見合う価値が得られるよう、適切なサービスが提供されているかをチェックしています。

システム安全性におけるヒューマンファクターの重要性

松本：鉄道システムの安全性を考える上で、RSSBではヒューマンファクター（人的要因）が重要な役割を果たすと考えていらっしゃいますね。具体的に、ヒューマンファクターとはどのような要素を指すのでしょうか。

ギブソン：最初に浮かび上がるのは「個人」です。鉄道業界では、とりわけ運転手や信号員などがシステムの中心部に位置しています。彼らが高いパフォーマンスで職務をまっとうするためには、装置や機器、仕事自体がベストな設計でなくてはなりません。また、彼らの働く環境やコミュニケーションがうまくいっているかという点にも、パフォーマンスの完成度は依存するでしょう。そういった様々な要素がヒューマンファクターとみなされます。

松本：そうした人的要因が安全性に影響を及ぼすという論には、具体的な根拠があるのでしょうか？

ギブソン：そもそも鉄道は長年にわたって開発され、十分に知識が普及している



松本 隆明 (まつもと たかあき)

1978年東京工業大学大学院修士課程修了。同年日本電信電話公社（現NTT）に入社、オペレーティング・システムの研究開発、大規模公共システムへの導入SE、キャリア共通調達仕様の開発・標準化、情報セキュリティ技術の研究開発に従事。2002年に株式会社NTTデータに移り、2003年より技術開発本部本部長。2007年NTTデータ先端技術株式会社常務取締役。2012年7月より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）所長。博士（工学）。

技術です。仕様も、長い時間をかけて最適化されてきました。私たちの分析によると、鉄道事故の90%は、何らかの人的ミスが原因の一部にあって起こるのです。

松本: 90%とは、予想以上に大きな割合ですね。一般的に、システムの事故や障害は、装置の故障や不具合によって起きることが多いと思われそうですが……。

ギブソン: 確におっしゃる通りです。しかし、メンテナンスや設計の段階で起きるエラーも、広い意味でヒューマンファクターの影響に含まれています。

松本: なるほど。オペレーションをするスタッフの操作ミスだけでなく、システムを作るときの設計ミスも含め、すべてがヒューマンエラーということですね。ハードの部品が故障するという事故も、部品の設計ミスだと捉えればヒューマンファクターとなる。それにしても、90%はかなり多いように感じます。

ギブソン: 鉄道業界における事故分析では、いわゆる“フロントライン”を定義することが重要です。事故の原因がオペレーターなのか、機器なのかという切り分けですね。しかし、鉄道の設計や事故では、単なる機器の故障というよりもヒューマンファクターに影響されて故障する場合も多いのです。

松本: 確かにヒューマンエラーを幅広く定義することは必要だと思いますが、同じく人的要因があるとはいえ、オペレーションとデザインのミスでは、対策がずいぶん変わってきますよね。

ギブソン: そうなのです。その差が、事故調査の制約となってくるケースも多々あります。オペレーター絡みのヒューマンエラーはきちんと取り上げられますが、デザイン面でのヒューマンエラーは十分に定義されないケースが多いようです。

松本: 2011年に、日本で飛行中に航空機がひっくり返ってしまった事件がありました。操縦士がドアの開閉スイッチと間違えて、隣のボタンを押したのが原因です。ボタンの配置というデザイン面でのヒューマンエラーだといえると思います。

ギブソン: その場合、第一次の原因は、やはりパイロットの操作ミスという人的エラーです。その背後に潜んだ二次的原因として、インターフェースのデザインエラーが挙げられるでしょう。デザインしていくプロセスで、そういったエラーに注目することが重要ですね。

松本: システムの安全設計に必要なポイントを考えると、実に様々な要素が出てきます。オペレーションのヒューマンエラーを防ぐという問題ひとつ取っても、オペレータの訓練や処遇、システムデザインなど、様々な問題と

対策がありますね。

ギブソン: はい。システムの全体を俯瞰的に考えていかなければならないでしょう。RSSBでは、装置機器や設計、トレーニング、仕事量、コミュニケーションの問題といった間接的な原因についても、それぞれに必要な取り組みを検討しています。

事故分類システム「IFCS」や スタッフから声を集める「CIRAS」

松本: 事故分類システム「IFCS (Incident Factor Classification System)」は、RSSBの取り組みでも非常に興味深い制度です。どのような事象を対象にしているのですか？

ギブソン: 事故調査報告として登録される年間7万件のデータベースから、ハイリスクな事象を年300件ほどサンプルにしています。内容は脱線や死亡事故から、危険信号が発信されたというものまで様々です。何かしらオペレーションの問題で故障が生まれたと見られる場合や、業界としてリスクが高いとみなされたときには、実際の事故までは発展しなくとも事象として含めます。

松本: 事故にならなくてもリスクが高いと思われる事象には、鉄道事業者から政府やRSSBへの事故調査報告義務があるのでしょうか。また、報告を受け取るためにNDAなどを結ぶのですか？

ギブソン: 特別な文書での契約はしていませんね。リスクがある程度のレベルに達した事象は、すべて正式な調査が必須です。鉄道事業者から政府とRSSBに挙がってきた報告で、重大事故と考えられるものを年に20件ほど選定し、国家の独立機関が更なる調査に当たります。

松本: そのような仕組みが確立されているのはうらやましいですね。私たちも事故情報を集めようとしていますが、なかなかうまくいきません。企業は、自社で起きた事象、とくに事故につながらなかった事象は外部に公開したがるのではないのです。鉄道事業者にとって、RSSBに情報を提供するメリットはどこにあるのでしょうか？ぜひ、参考にさせていただきたいです。

ギブソン: 自分たちの持つリスクに対して、理解を深められるのが鉄道事業者のメリットです。

松本: 事故事例やリスクを共有するべきだということは、一般論では理解できます。しかし、企業の個別事例になると、話が変わってくる。「社員教育が悪いから、オペレーションにミスが生まれたのではないか」などと、その企業を責める材料に使われてしまうことがあるのです。だから、企業が情報を出したがりません。

ギブソン：その傾向はありますね。ですから私たちも、情報の集計時には匿名化を行い、具体的な企業名がわからない形にしています。本来なら、より良い成果を得るために詳細までシェアしていきたいのですが、業界から賛同を得られるかはわかりません。

松本：そうした事故分析を通じて、例えばコミュニケーションに起因する問題が多いなどとわかったとき、結果はどのように企業へフィードバックされているのですか？

ギブソン：RSSBは、「運用の経験からの教訓」と題した全国規模のサマリーレポートを持っています。その中に、事故分類に関するセクションを設けているのです。例えばコミュニケーションの問題については、業界をまたいで協業する「コミュニケーション戦略グループ」を組織し、更に詳細な情報を活用しています。「人間の疲労」を扱うチームなどもあります。メンバは、ほとんどRSSBと鉄道事業者のみです。

松本：つまり、詳細に報告された分析結果をどのように役立てていくかは、基本的に鉄道業者自身のミッションになるということですね。私たちもできるだけ事故事例を集めて分析していますが、今後の糧になる教訓が見つかって、普及していくのがまた難しいところです。RSSBで何か工夫されている点はありますか。

ギブソン：個人的な経験ですが、企業に大量の詳細情報を渡すだけで関心を持ってもらうのは、なかなか難しいことです。とくにリスクの高い領域がどこなのか、わかりやすく特定したり、協業を通じて関心を引き出していく。それがうまくいく秘訣だと思います。

松本：また、取り組みの効果をうまく伝えていかないと、関心は得られないですね。例えば「コミュニケーションの問題が大きいから、こんなふうに手を打てば、これだけ損失が減る」といった明確な効果を示す必要があります。

ギブソン：確かにそうですね。私たちは調査・研究をする際、具体的な測定値や仕様も考えていくことが多いです。また、良い慣行や事例を持つ企業と協力して、様々なベストプラクティスを集め、他の企業にわかりやすく効果を示すようにしています。

松本：「CIRAS (Confidential Reporting System for Industry)」の取り組みにも興味があります。これは、一般のお客様からクレームを受け付けるようなシステムだと思えばいいのでしょうか。

ギブソン：いいえ、違います。CIRASは、鉄道会社の職員から、会社を通ることなく懸念の声を吸い上げるシ

ステムです。社内でうまく提起できなかつたり、会社が取り上げてくれなかった課題を集める仕組みになっています。年間3,000件ほどの連絡が寄せられていますね。

松本：多くのスタッフが声を挙げてくるのは、素晴らしいですね。安全性に関して、それだけ現場の関心が高いということです。良い仕組みだと思いますが、自社のスタッフから都合の悪い情報が出てくる可能性もあるため、企業側はあまり前向きに考えないような気がします。この仕組みは、政府と業界のどちらが主導して作ったのですか？

ギブソン：ラドブローク・グローブの事故調査報告で提言された内容から、業界主導で生まれた仕組みです。導入レベルは異なるものの、主要企業はすべて参加しています。安全を重視する文化の中で必要性は十分認識されているし、「お互いにCIRASへ参加するべきだ」という会社間のプレッシャーもあります。

松本：まず、安全性へ取り組む文化を育てていくことが重要なのかも知れませんね。実際にCIRASを通じて挙がってきた声に対して、対策はなされているのですか？

ギブソン：はい。スタッフから懸念が挙がってくると、RSSBがその声を匿名化しつつ、企業と対話します。その後、課題を解決するために新たな仕組みを導入したり、既存のルールを変更したり、それぞれに様々な対策が取られています。

松本：よくわかりました、ありがとうございます。今日は、鉄道先進国ならではの進んだ取り組みを聞かせていただき、とても参考になりました。私たちも障害情報の分析や収集、対策の共有に取り組んでいますが、ヨーロッパのオープンカルチャーと、日本のクローズドカルチャーといった文化の問題も大きいように感じました。文化の差によって、同じようには進められない部分も多そうです。本日のお話や事例を参考にしつつ、我が国の実情に即した仕組み作りに取り組んでいきたいと思っています。



プラットフォーム依存種検索による ソースコードからのプラットフォーム 依存部抽出手法

岡本 周之⁺¹⁺²藤原 貴之⁺¹楠本 真二⁺²岡野 浩三⁺²

組込みシステム開発では、プラットフォーム（PF）変更に伴うソフトウェア移植作業の効率向上が求められる。本稿では、PF 間移植工数の削減を目的とし、PF 依存種検索によるソースコードからの PF 依存部抽出手法を提案する。本手法に従って PF 依存部抽出ツールを開発し、製品ソースコードの移植に適用した結果、PF 依存部の検索・判定・修正工数を 49%、同判定工数を 40% 削減でき、本手法の有効性を検証できた。

Abstracting Method of Platform Dependent Code with Analyzing Platform Dependency in Source Code

Chikashi Okamoto⁺¹⁺², Takayuki Fujiwara⁺¹, Shinji Kusumoto⁺², Kozo Okano⁺²

The embedded system development requires efficiency improvement in software migration with platform (PF) change. We propose an abstracting method of PF dependent code (PFD) with analyzing PF dependency in source code for reducing the migration cost. We have developed a PFD finding tool using the method and have migrated production source code with the tool. It can reduce PFD finding, judging and modifying cost, and PFD judging cost by 49% and 40%, respectively.

1. はじめに

組込みシステムでは、部品コスト削減、供給停止に伴う代替、性能向上のために CPU などのハードウェアを変更する必要がある。また、機能追加や海外展開の容易化、新技術への追従性向上、ライセンス料削減、開発効率向上のために OS を変更する必要がある。つまり、CPU や OS 等のプラットフォーム (PF) の変更を行う必要がある。

一方、組込みシステムにおけるソフトウェア開発では、既存ソフトウェアを活用する拡張 (差分 / 派生) 開発が多い。

プロジェクト件数比率を図 1.1 に示す [METI2011]。

また、組込みシステムでは機能と開発費の多くをソフトウェアが占めており、工数（コスト）削減のためソフ

【脚注】

- + 1 (株)日立製作所 横浜研究所
Yokohama Research Laboratory, Hitachi, Ltd.
- + 2 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
- a) 本稿で述べられたシステムおよび製品名は、一般に各社の商標または登録商標である。

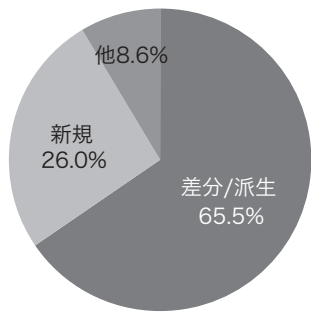


図 1.1 プロジェクト件数比率

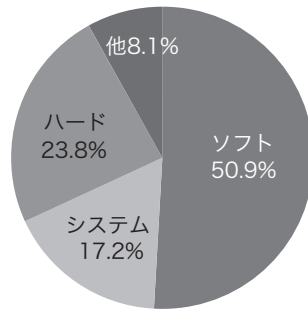


図 1.2 開発費内訳

ト開発効率向上が必要である。開発費内訳を図 1.2 に示す [METI2011]。

さらに、組込みシステムでは、厳しいコスト制約のため限られたハードウェアリソース（CPU の演算処理能力、メモリ容量など）で動作するようソフトウェアを開発しなければならない。

このため、開発済みの組込みシステムにおいて、現行 PF と同程度のハードウェアリソースを持つ新 PF へ変更する際のソフトウェア開発工数削減が課題となる。

本稿では、この課題の解決に必要な、PF 間移植における既存ソフトウェアからの PF 依存部抽出について、「PF 依存種検索によるソースコードからの PF 依存部抽出手法」を提案する。本手法では、あらかじめ PF 依存種（とその検索方法）の一覧を作成しておき、これを用いて既存ソフトウェアのソースコードを解析して PF 依存部候補を検索した後、候補から PF 依存部を抽出する。

ソフトウェア開発工数削減に対する既存の手法としては、ソフトウェアプロダクトライン (SPL) 開発 [Clements2002] [Pohl2005], モデル駆動型開発 (MDD) [Selic2003] などがある。本稿での提案手法は、これらの既存手法と比較して、PF 変更時の既存ソフトウェア移植に最適化している。また、組み合わせることも可能である。

本提案手法に従って PF 依存部抽出ツールを開発し、実際の製品ソースコードの移植に適用した。この結果、PF 依存部の検索・判定・修正工数を 49%、同判定工数を 40% 削減できるとの見通しを得、本提案手法の有効性を検証できた。

2. 課題とアプローチ

2 章では、PF 間移植手順について述べた後、手順の 1 つである PF 依存部抽出について、従来手法とその問題点、ならびに本稿で提案する手法とその効果を述べる。

2.1. PF 間移植手順

PF の変更方法には次の (1) ~ (3) がある (表 2.1 参照)。

- (1) 既存ソフトウェアを新 PF に合うように新規に作成し直す場合、新規作成工数 (大) が必用である。

表 2.1 PF の変更方法

PF 変更方法	概要	開発工数	要求リソース
(1) 既存ソフト → 新ソフト 既存 OS → 新 OS 既存マイコン → 新マイコン	全ソフトを新 PF 向けに新規作成	× (大)	○ 既存ソフトと同等
(2) 既存ソフト → 既存ソフト 既存 OS → 既存 OS 既存マイコン → 仮想マイコン	仮想化技術で既存ソフトをそのまま使用	○ (小)	× 仮想化用 CPU/メモリが必要
(3) 既存ソフト → 既存ソフト 既存 OS → 新 OS 既存マイコン → 新マイコン	既存ソフトを新 PF 向けに一部改修	△ (中)	○ 既存ソフトと同等

- (2) 仮想化技術を用いて既存ソフトウェアをそのまま使用する場合、仮想化のための CPU/メモリが必要である。

- (3) 既存ソフトウェアを新 PF に合うように一部改修して (PF 間で移植して) 使用する場合、改修工数 (中) がかかる。

新 PF が、現行 PF と同程度のハードウェアリソースしか持たない場合、(2) は不向きである。ソフトウェア開発工数を考慮すると (1) より (3) が現実的である。そこで本稿では、PF 変更方法に (3)、すなわち PF 間移植を用いる場合を対象とする。

PF 間移植において、工数削減のため移植作業を必要最小限にするには、既存ソフトの PF 依存部と PF 非依存部を分離し、PF 依存部のみを修正する必要がある。

ここで、デバイスドライバ層を PF 依存部とするなど、階層構造からある程度 PF 依存部を分離できるが、組込みシステムでは、理想的な階層構造を維持できない場合が多い。例えば、厳しいコスト制約のため限られたリソースの中でソフトウェア処理の高速化が求められることがある。この場合、ソフトウェアの階層構造を理想的に保つことよりも、例外的な呼び出し方を許してでも処理の高速性が優先されてしまうことがある。

また、出荷間際で開発期間が不足する中で、不具合に対するソフトウェア修正が求められることもある。この場合、不具合解消に有効な手段の内、実装時間がかかる理想的な手段よりも、理想的ではないが短時間で実装可能な手段が優先されてしまうことがある。

拡張開発を繰り返した場合は、このような状況が積み重なるため、理想的な構造の維持がさらに困難となる [Fowler1999][Ochiai2002]。

この結果、PF 依存部が集約された PF 依存層と、PF 依存部が含まれていない PF 非依存層について、PF 層のすぐ上位に集約されるべき PF 依存部分が、ソフトウェア全体に離散してしまい、一括して把握することが困難になる。

このような状態に対応するための、一般的な PF 間移植手順を図 2.1 に示す。

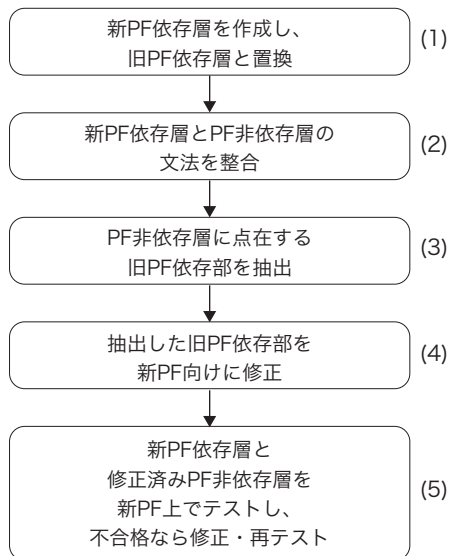


図 2.1 PF 間移植手順

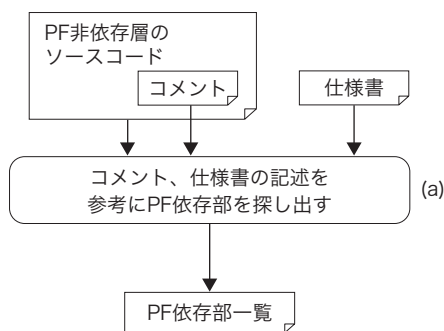


図 2.2 従来の PF 依存部抽出手法

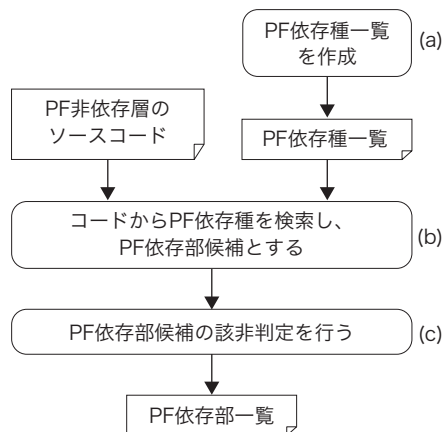


図 2.3 提案する PF 依存部抽出手法

- (1) PF 依存層（ドライバ層など）を新 PF（ハード、OS 層など）用に作成し、旧 PF 依存層と置換する。
- (2) 新 PF 依存層と、PF 非依存層（ミドル層、アプリ層など）について、文法上の整合を取る。
- (3) PF 非依存層に点在する旧 PF 依存部を抽出する。
- (4) 抽出した旧 PF 依存部を新 PF 向けに修正する（新規作成関数との置換もある）。
- (5) 新 PF 依存層と修正済みの PF 非依存層を新 PF 上でテストし、不合格なら修正して再度テストする。

2.2. 従来の PF 依存部抽出手法の問題

PF 間移植手順 (3) について、社内の過去の移植事例と参考文献を基に、従来の PF 依存部抽出手法の問題点を述べる。

従来の PF 依存部抽出手法を図 2.2 に示す。従来手法では、ハードウェアや OS 等 PF との関連が記された仕様書、コード内の PF 依存に関するコメントなどと、ソースコードを照らし合わせて PF 依存部を探し出し、PF 依存部一覧を作成する。

従来手法では、仕様書にハードウェア依存部である旨が記載されている部分を、PF 依存部と認識していた。また、ソースコードのコメントに同様の記述があることをもって、PF 依存部と認識していた。

しかし、仕様書やソースコードのコメントに、PF 依存部に関する記載がない、または記載があっても誤っていることが多い [Spinellis2006]。例えば、将来の拡張のことを十分に考慮していない場合（OS が変更になることは想定していなかった等）や、開発時間の不足でコードのみ修正し、仕様書やコメントを修正しなかった場合などである。その結果、PF 依存部の抽出が不完全となり、テスト不合格で再度修正するケースが増え、手戻り工数が増える。

また、近年、組込みシステムの多機能化、大規模化、複雑化に伴い、完全に網羅的にテストを実施することは、困難になっており、潜在的な不具合がテストで顕在化せず、製品出荷後に問題を引き起こすことがある [Sugiyama2003]。出荷後の不具合は対策費の増大を招き、経産省による 193 事業部門への調査では、2008 年度に不具合が発生した事業部門の 17% において対策費総額が 1 億円以上に上ったことが判明した [METI2010B]。

2.3. 提案する PF 依存部抽出手法とその効果

上記問題点を解決するため、本稿で提案する「PF 依存種検索によるソースコードからの PF 依存部抽出手法」では、ドキュメントやコメントではなく、ソースコード（実行命令部分）の情報を用いて PF 依存部を判定する。

ソースコードは、更新されていない可能性がある仕様書やコメントと異なり、旧 PF で動いているソフトウェアを作成するのに用いられる最新情報であるため、現状のプログラムと乖離することを避けられる。

本稿で提案する PF 依存部抽出手法を図 2.3 に示す。本提案手法では、(a) あらかじめ PF 依存種（とその検索方法）の一覧を作成しておき、(b) PF 非依存層のソースコードから PF 依存種を検索した結果を PF 依存部候補とする。(c) この PF 依存部候補について、PF 依存部か否かの該非判定を行って PF 依存部を得、一覧を作成する。

本提案手法の特徴は、次の (1) ~ (3) にまとめられる。

(1) PF 依存種が一覧化されているため、過去の事例から判明した PF 依存種を早い段階で漏れなく抽出でき、同じテストを何回も実施しなくて済み、潜在的な不具合がテストで顕在化しないリスクを減らせる。

(2) 複数の PF 依存部をまとめて対応することができるため、工数が削減でき、対応方法を最適化できる。すなわち、リファクタリングが効率的に実施できる。

(3) PF 依存種を定型化することで、自動化ツールによる作業支援が可能になるため、更なる工数削減が可能となる。

3. PF 依存種検索によるソースコードからの PF 依存部抽出手法

3 章では、提案する PF 依存部抽出手法のキーアイデアである PF 依存種とその一覧を説明する。

3.1. PF 依存要因

PF 間移植における、PF 依存要因ごとの既存アプリケーション（アプリ）への影響と、対応方法を図 3.1 に示す。

ライブラリ及び OS への依存部は、ライブラリ API やシステムコールの I/F 変換ラップを用いることで、アプリケーションの修正を行うことなく対応することができる。ただし、変換ラップを用いる I/F を抽出する必要がある。

一方、デバイスドライバ（ドライバ）などのソフトウェア、マイコンやマイコン周辺デバイス（デバイス）などのハードウェア、コンパイラなどの開発環境への依存部は、これを抽出して個別に修正する必要がある。

3.2. PF 依存種

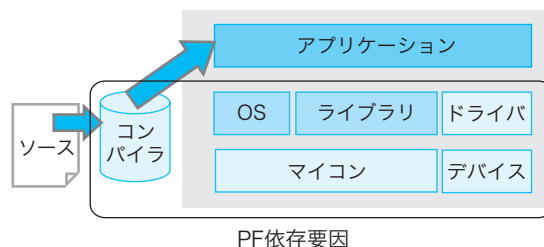
本研究では、PF 移行に際して修正が必須となる可能性のある PF 依存部を、PF 依存内容別に 39 種類に分類した。この分類では、社内の過去の移植事例から 23 種を集め、その他 MISRA-C[MSR] にある他の 16 種と合わせた。なお、移植性を高めるためのリファクタリングなど、PF 移行に必ずしも必要ではない修正は含めていない。

本研究で整理した PF 依存種一覧の抜粋を表 3.1 に示す。例えば、旧 PF 用の既存コードを検索した結果「ソケット通信用構造体へのデータ入出力」という PF 依存種に該当するコードが検出された場合、PF の「エンディアン」の種別（ビッグ_ENDIAN か）、または「パディング有無」に依存している可能性があることを示している。もし旧 PF 用の既存コード

がビッグ_ENDIAN 依存であり、かつ、新 PF がリトル_ENDIAN である場合は、PF 移行に際して PF 依存部の修正が必要となる。

なお、PF 依存種の一覧を付録の表 Ex.1 に記す。

また、移植時の修正難易度を「易」「普」「難」の 3 段階で分類し、大まかな修正工数把握を可能とした。修正難易



PF 依存要因

PF 依存要因	対応方法	アプリ修正
ライブラリ	I/F 変換ラップ を利用	不要
OS		
ドライバ	PF 依存部を抽出して個別に修正	要
デバイス		
マイコン		
コンパイラ		

図 3.1 PF 依存要因と対応方法

表 3.1 PF 依存種一覧（全 39 種）の抜粋

PF 依存種 (検索対象コード)	PF 依存内容	修正 難易度	修正 重要度
ソケット通信用構造体へのデータ入出力	エンディアン、パディング有無	難	2
負整数の除算、剰余算	小数部の切上げ切捨て	普	1
レジスタへの直接アドレス参照	アドレス値	難	3
ポインタから固定幅整数型へのキャスト	ポインタサイズ	普	2
構造体メンバへの直値オフセット参照	パディング有無	難	1
システムコールの呼出	システムコール名	易	4

表 3.2 修正難易度の基準と該当 PF 依存種数

修正 難易 度	基準	該当依存種数	
		移植事例	MISRA-C
易	移植前後のコードがほぼ 1:1 で機械的に変換可能	7	5
普	変更方法が 2 通り以上あり、状況に応じた選択が必要。PF 依存部 1 行に対し変換後コードが数行	8	11
難	PF 依存部以外に関連箇所への影響あり。ハードウェアに依存。PF 依存部 1 行に対し数十～数百行のコード変更が必要	8	0

度の基準と該当する PF 依存種数を表 3.2 に示す。更に、移植時の修正重要度を 1～4 の 4 段階で分類し、大まかな修正順の決定を可能とした。修正重要度の基準と該当する PF 依存種数を表 3.3 に示す。これにより、例えば、修正難易度が高く修正重要度が低い PF 依存部よりも、修正難易度が低く修正重要度が高い PF 依存部を優先して対応すべきであるが、このような判断を容易化できる。

表 3.3 修正重要度の基準と該当 PF 依存種数

修正重要度	基準	該当依存種数	
		移植事例	MISRA-C
1	コンパイルは通るが、動作に問題がでる可能性がある	9	13
2	コンパイルは通るが、ほぼ確実に動作に問題が生じる（値が不正になるなど）	8	3
3	コンパイルは通るが、動作に致命的な問題が生じる（メモリの範囲外アクセス、無限ループなど）	3	0
4	コンパイルが通らない	3	0

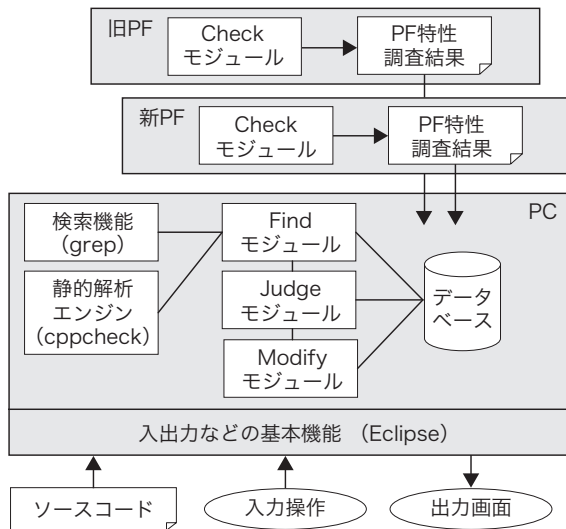


図 4.1 PF 依存部抽出支援ツールの機能構成

4. PF 依存部抽出支援ツールの開発

本稿で提案する PF 依存部抽出手法の適用を支援するための PF 依存部抽出支援ツールを開発した。4 章では、本ツールについて述べる。

4.1. PF 依存部抽出支援ツールの概要

本ツールの概略機能構成を図 4.1 に示す。本ツールは、新旧それぞれの PF 上で動作する (1)Check モジュールと、PC 上で動作する (2)Find モジュール、(3)Judge モジュール、(4)Modify モジュール、検索機能、静的解析エンジン及びデー

タベースからなる。

組込み機器開発で取り扱われる言語の 60～70% は C 言語であるため [METI2010A][IPA2013]、本ツールの解析対象は C 言語ファイルとした。

また、入出力などの基本機能を容易に実装するため、OSS の統合開発環境として広く普及している Eclipse[ECL] をベースに用いた。また、同じく OSS である Cppcheck[CPC] を改良し、ソースコードの静的解析エンジンとして用いた。検索機能としては grep コマンドを用いた。以下、各モジュールについて説明する。

(1) Check モジュール

Check モジュールは、新旧それぞれの PF 上で動作し、表 3.1 の PF 依存内容に相当する PF 特性を調査する。例えば、旧 PF のエンディアンがビッグで、新 PF のエンディアンがリトルであることなどを得る。PF 特性調査結果はファイルとして出力される。

(2) Find モジュール

Find モジュールは、データベースに保存された PF 依存種に該当するコード (PF 依存部) を、入力されたソースコードから検索する。検索結果は、PF 依存部候補一覧としてデータベースに保存される。

なお、Find モジュールは、新旧 PF における PF 特性調査結果を元に、両方で PF 特性が異なる PF 依存種のみを検索対象とする。本ツールでの出力画面例を図 4.2(a) に示す。この画面は、旧 PF のエンディアンがビッグで、新 PF のエンディアンがリトルであるため、エンディアンに関連する PF 依存種が自動選定された場合の例を示している。

(3) Judge モジュール

Judge モジュールは、データベースに保存された PF 依存部候補一覧を利用者に提示する。また、利用者がソースコードを解析し、それぞれの PF 依存部候補が PF 依存部か否かの該当判定を行った後、Judge モジュールはその該当判定結果を受け付ける。本ツールでの出力画面例を図 4.2(b) に示す。この画面は、PF 依存部候補の内 ID13 は PF 依存部である、と利用者が判定しチェックをつけた場合の例を示している。該当判定結果で PF 依存部とされた箇所は、PF 依存部一覧としてデータベースに保存される。

(4) Modify モジュール

Modify モジュールは、データベースに保存された PF 依存部一覧を利用者に提示する。また、利用者が PF 依存部の 1 つを選択した場合、該当するコードと、PF 依存種およびその修正案を提示する。本ツールでの出力画面例を図 4.2(c) に示す。この画面は、PF 依存部一覧の内 ID 4 の PF 依存部を利用者が選択し、該当する「applet_tables.c の 45 行目」とその前後のコード、PF 依存種およびその修正案が表示された場合の例を示している。

なお、図 4.2(a), (b), (c) に示した画面例は、各機能の説明用にそれぞれ典型的なものを選んだ。同一解析過程での画面ではないため、表示項目は相互に対応しているとは限らない。

4.2. PF 依存部抽出支援ツールの利用手順

PF 依存部抽出支援ツールの利用手順を以下に示す。

(1) ツールの Check 機能を用いて旧 PF と新 PF の PF 特性

を調査する。

(2) ツールの Find 機能を用いて PF 特性が異なる PF 依存種をソースコードから検索し、PF 依存部候補を得る。

(3) ツールの Judge 機能を用いて PF 依存部候補の該非判定を行い、判定結果をツールに入力する。

(4) 各 PF 依存部について、ツールの Modify 機能が出力した PF 依存種別の修正方法を参考に修正する。



図 4.2(a) PF 依存部抽出支援ツールの出力画面例

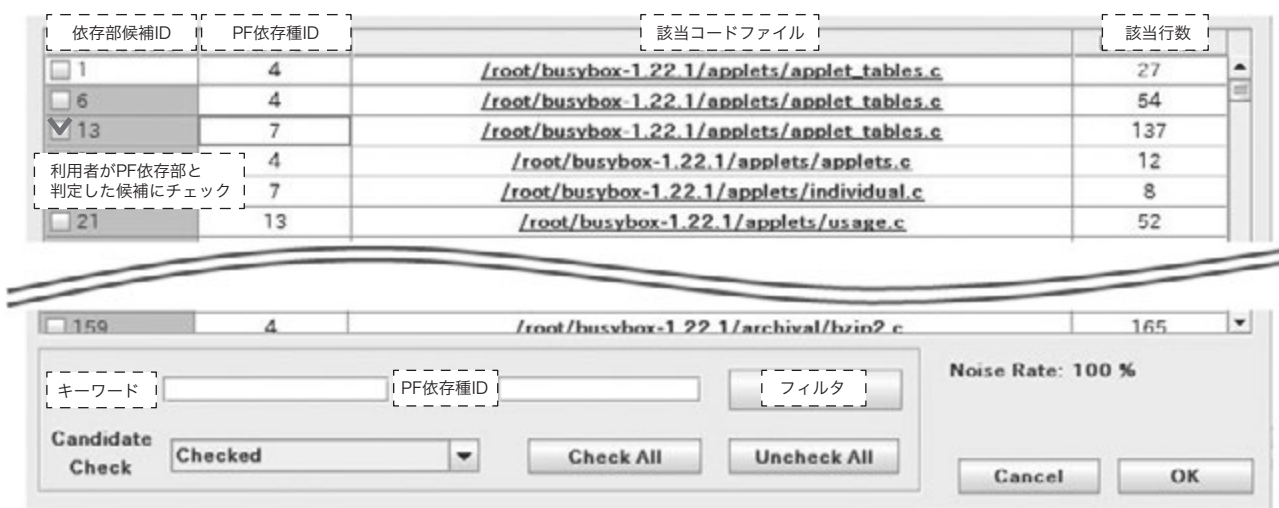


図 4.2(b) PF 依存部抽出支援ツールの出力画面例

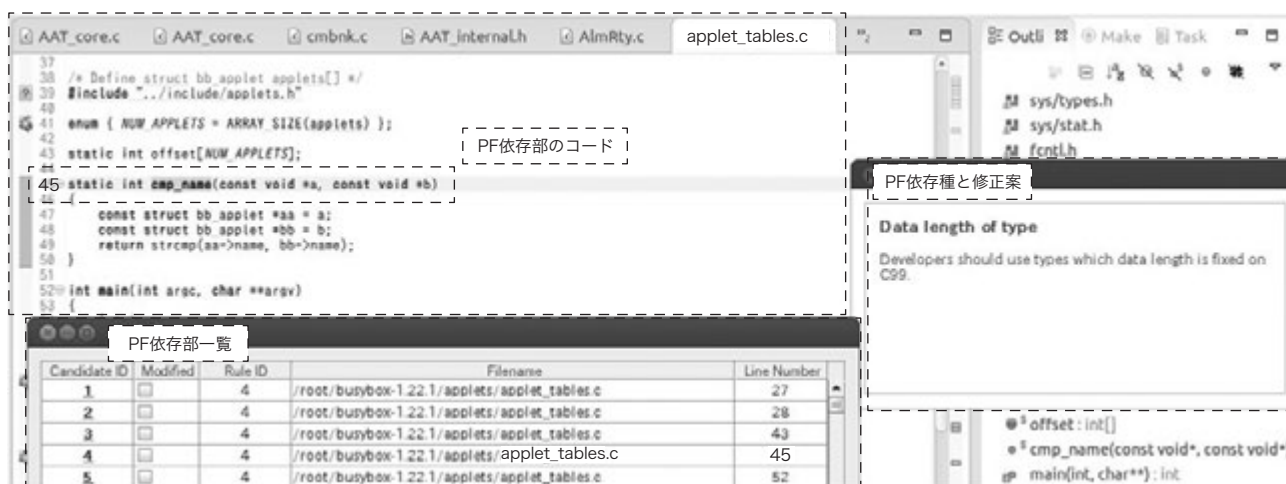


図 4.2(c) PF 依存部抽出支援ツールの出力画面例

4.3. PF 依存部抽出支援ツールの処理方式

PF 依存部抽出支援ツールの代表的なモジュールである Check モジュールと Find モジュールの処理方式について、以下に示す。

(1) Check モジュール

Check モジュールでは、例えば PF 特性としてエンディアンを調べる場合、エンディアンがビッグリトルかによって値が変化するようなプログラムのコードを用意しておく。コードを新旧 PF 用にそれぞれコンパイルして、各 PF 上で実行し、その値からエンディアンの種別を判定する。型のサイズを調べる場合なども同様に行う。本 Check モジュールでは、判定処理もプログラム化し、判定結果を PF 特性結果ファイルとして出力させた。

(2) Find モジュール

PF 依存部の抽出を行う Find モジュールの処理方式、すなわちソースコードの検索方式は、抽出する PF 依存種により異なる。代表的な方式を以下に示す。

(i) 単純キーワード検索方式

本方式では cppcheck を使わず、grep により特定キーワードをソースコードから検索する。例えば「#pragma」という文字列による検索が該当した部分を、「プラグマの使用」という PF 依存部の候補として抽出する。

(ii) 数値検索+値確認方式

本方式では cppcheck を使わず、grep により特定の種類の数値を検索し、その値が特定範囲に含まれているものを抽出する。例えば「0x」という文字列で検索した 16 進数が、レジスタに割り当てられたメモリアドレスの範囲に含まれていた場合、その数を利用する部分を「レジスタへの直接アドレス参照」という PF 依存部の候補として抽出する。

(iii) 関数検索+名前確認方式

本方式では、cppcheck を使って関数シンボルを検索し、そのシンボルが特定の文字列と一致するものを抽出する。例えば関数シンボルが、システムコール一覧の「semTake」と一致した場合、そのシンボルの利用部分を「システムコールの呼出」という PF 依存部の候補として抽出する。

(iv) 関数検索+名前確認+引数確認方式

本方式では、(iii)に加えて、引数の型も条件に加える。例えば、関数シンボルが「send」であり、かつ、第 2 引数の型が「char または unsigned char の、ポインタまたは配列」でない場合、「ソケット通信用構造体へのデータ入出力」という PF 依存部の候補として抽出する。

(v) 代入検索+キャスト確認+サイズ確認方式

本方式では、cppcheck の機能を複合的に使う。「変数 = 変数」の形を検索し、該当部分が「unsigned」「signed」間でキャストしており、かつ、右辺のサイズが左辺のサイズ

より大きい場合、「明示的、暗黙的な型変換」という PF 依存部の候補として抽出する。

5. 製品ソースコードへの適用評価

産業用インフラシステム向け情報機器の製品ソースコードの移植に、提案する PF 依存部抽出手法を適用し評価した。

5.1. 適用対象

本評価における提案手法の適用対象は、産業用インフラシステム向け情報機器の製品ソースコード（コード規模は数百 KB、言語は C）のうち、主要機能（コード規模は製品全体の約 50%）部分とした。本手法を適用した移植では、マイコンのアーキテクチャを SH から ARM に、OS を VxWorks から Linux に、それぞれ変更した。

また、MISRA-C で定義されている PF 依存部は、例えば富士通ソフトウェアテクノロジーズの PGRRelief[PGR] やテクマトリックスの C++Test[CPT] など、既存の静的解析ツールで抽出可能なため、これらを除いた 23 種の PF 依存種を評価用の自動化対象とし、4 章で述べた PF 依存部抽出支援ツールとして実装した。

5.2. 適用手順

本評価における PF 間移植では、図 2.1 に示した PF 間移植手順 (3) において PF 依存部抽出支援ツールを用い、提案手法を適用した。

5.3. 評価項目・評価方法・評価結果

本評価の評価項目ごとに、評価方法、評価結果を述べる。

(1) 移植工数

PF 依存部抽出支援ツールを用いた PF 間移植手順を 3 つの移植工程に分類し、本評価における各移植工程の工数を計測した。具体的には、手順 (1)(2) を移植工程 (A)「文法上の整合」、手順 (3)(4) を移植工程 (B)「PF 依存部修正方法の検討と実装」、手順 (5) を移植工程 (C)「動作確認とデバッグ」とした。移植工程 (A) には、ビルド環境構築、OS・ドライバ整備、リンクエラー修正などが含まれる。

移植工数の評価結果を表 5.1 に示す。本評価の移植では、移植工程 (A)「文法上の整合」に 77.5 時間（全工程に対し 24%）、移植工程 (B)「PF 依存部修正方法の検討と実装」に 115.0 時間（同 35%）、移植工程 (C)「動作確認とデバッグ」に 132.0 時間（同 41%）を費やした。また、移植工程 (B) の内、ツールで抽出可能な PF 依存部に関する作業工数 (B1) は 104.5 時間（同 32%）、それ以外の工数 (B2) は 10.5 時間（同 3%）であった。

(2) PF 特性調査機能の有効範囲

移植前後で特定の PF 特性が同じである場合に、その PF 特性に関連する PF 依存種は、PF 間移植時の修正が必須ではない。例えば移植前後の PF がどちらもビッグエンディア

ンの場合、移植対象コードがビッグエンディアン依存であったとしても、そのPF依存部を修正することなくPF間で移植できる。つまり、PF依存種に関連するPF特性をあらかじめ得ておけば、不要な(移植に必須ではない)PF依存部調査・修正工数の削減につながる。

そこで、PF特性調査機能、すなわちPF依存部抽出支援ツールのcheck機能を用いて得られるPF特性について、関連するPF依存種の数と割合を求めた。

「PF特性調査機能の有効範囲」の評価結果を表5.2に示す。全PF依存種39項目中19項目(49%)について、関連するPF特性を得られた。この19項目のPF依存種全てが、社内の過去の移植事例から得た23種に含まれており、MISRA-Cから得た16種のものはない。

(3) PF依存部候補数

PF間移植手順(3)(b)において、PF依存部候補の抽出件数が少なければ、その分、手順(3)(c)の該否判定工数が削減できる。そこで、PF依存部候補の抽出件数について、手動時の場合とツール利用時の場合を測定し、比較した。

PF依存部候補数の評価結果を表5.3に示す。抽出された候補数は、手動時の場合の約28k個(3a)に対して、ツール利用時の場合は約17k個(3b)(3aに対し60%)に低減されていた。

(4) 提案手法の再現率

移植工程(B)(C)を合わせて、最終的に移植に修正が必要だった箇所を、提案手法による抽出可否で分類した。

「提案手法の再現率」の評価結果を表5.4に示す。全修正1107件の内、(4a)提案手法で抽出可能なPF依存部の修正が5種935件(全修正に対し85%)、(4b)提案手法では抽出不可だが技術的に抽出可能なPF依存部の修正が6種159件(同14%)、(4c)完全に対応するのは技術的に困難なPF依存部の修正が3種3件(同0.3%)、(4d)PF依存と無関係の修正が4種10件(同0.9%)であった。また、(4a)で修正したPF依存部(5種)は、修正難易度が「易」2種、「普」1種、「難」2種であり、修正重要度が「2」1種、「3」2種、「4」2種であった。また、(4a)で修正したPF依存部(5種)の全てが、社内の過去の移植事例から得た23種に含まれており、MISRA-Cから得た16種のものはない。

(3)PF依存部候補数の評価との関係を述べると、「(3b)ツール利用で得た16,904個のPF依存部候補」の内、実際にPF依存であり移植に際して修正を要したのが(4a)の935個である。その他の修正(4b)(4c)(4d)(合計172箇所)は、(4a)に関する移植工程(B)(C)で見つかったものであるが、「(3a)手動で得た28,139個のPF依存部候補」は比較用に求めたものであり、移植には利用していない。

(5) 手動時とツール利用時の再現性の比較

本評価の移植において実際に変更が必要だったPF依存

表 5.1 移植工数

移植工程	PF間移植手順	工数(時間)	全工程に対する割合	
(A) 文法上の整合	(1)(2)	77.5	24%	
(B) PF依存部の修正方法の検討と実装	(3)(4)	115.0	35%	
		(B1) ツール抽出可	104.5	32%
		(B2) ツール抽出不可	10.5	3%
(C) 動作確認とデバッグ	(5)	132.0	41%	
合計		324.5		

表 5.2 PF特性調査機能の有効範囲

提案手法で用いる全PF依存種	PF特性調査機能で得られるPF特性に関連するPF依存種	全PF依存種に対する割合
39項目	19項目	49%

表 5.3 PF依存部候補数

抽出されたPF依存部候補数(3a:手動時)	抽出されたPF依存部候補数(3b:ツール利用時)	手動時に対する割合
28,139個	16,904個	60%

表 5.4 提案手法の再現率

修正内容	修正種類	修正件数	全修正に対する割合
全修正	18種	1107件	-
(4a)提案手法で抽出可能なPF依存部の修正	5種	935件	85%
(4b)提案手法で抽出不可だが技術的に抽出可能なPF依存部の修正	6種	159件	14%
(4c)完全に対応するのは技術的に困難なPF依存部の修正	3種	3件	0.3%
(4d)PF依存と無関係の修正	4種	10件	0.9%

表 5.5 手動時とツール利用時の再現性の比較

PF依存部の区分	該当数 ^(*)	(5a)に対する割合
(5a)変更が必要	741個	-
(5b)手動で抽出可能	741個	100%
(5c)ツールで抽出可能	732個	99%

*1:比較可能なPF依存部のみ

箇所内、手動時の場合とツール利用時の場合とで、それぞれ何箇所をPF依存部候補として抽出できたかを測定し、比較した。なお、キーワード検索と静的解析を併用して抽出したPF依存部は、正確に比較できないため評価対象外とした。

「手動時とツール利用時の再現性の比較」の評価結果を表5.5に示す。(4)提案手法の再現率の評価における「修正したPF依存部(4a)(4b)(4c)(合計1097箇所)」の内、比較対象となるPF依存部は741個あった(5a)。この741個のPF依存部は、(3)PF依存部候補数の評価における「(3a)手動で得た28,139個のPF依存部候補」の中に741個(5aに対し100%)含まれており(5b)、(3)PF依存部候補数の評価における「(3b)ツール利用で得た16,904個のPF依存部候補」の中に732個(同99%)含まれていた(5c)。

5.4. 考察

(1) 移植工数

本評価では、移植工程(A)「文法上の整合」の工数は全体の24%であったが、今回の移植対象ではOS・ドライバ整備の工数が発生していない。

汎用OSを搭載した汎用ハードウェア機器に移植する場合は、標準ライブラリや標準ドライバが入手可能であり、本工数はそれほど大きくならないが、特殊な周辺デバイスを持つ機器への移植などの場合は、ドライバの新規作成が必要となり、本工数がより大きくなると想定される。

また、今回の移植では、(4)提案手法の再現率の評価における「提案手法で抽出可能なPF依存部の修正(4a)」件数(935件)の9割以上を、修正難易度「易」と「普」のPF依存部が占めていた。修正難易度「難」のPF依存部が多い場合は、同程度の修正件数であっても移植工数がより大きくなると想定される。

(2) PF 特性調査機能の有効範囲

本評価では、PF依存部抽出支援ツールにより、全依存種の49%(19項目)に関連するPF特性を得られ、この19項目全てが、本ツールの自動化対象であった。

PF依存箇所が全PF依存種に平均的に分布しており、かつ、ツールで得られたPF特性がPF間移植前後で全て同じである場合は、PF依存リストに従って闇雲に抽出・判定するのと比較して、49%の工数削減が可能であると言える。

また、ツールで抽出可能なPF依存部に関する修正方法の検討と実装作業の工数(B1)は全移植工程の32%であったので、同程度の割合の工程においてこの工数削減効果が得られると言える。

(3) PF 依存部候補数

PF依存部候補の抽出件数は、手動時の場合に対して、ツール利用時の場合は60%に低減されていた。これは、手動時がキーワード検索のみであるのに対して、ツール利用時は

ソースコードの静的解析結果を併用することで不要な候補が除外されているためである。例えば、VxWorksで利用されているsocketのような一般名称やiなどの短い名称の関数を抽出する際、キーワード検索だけでは同名の変数も抽出されてしまうが、静的解析との併用によりこれを避けられる。

PF依存部候補の該否判定工数が均一な場合は、ツール利用によりPF依存の該否判定工数が40%削減可能であると言える。

(4) 提案手法の再現率

本評価では、提案手法の再現率は(4a)85%であった。ただし本評価の移植では、全PF依存種に関して修正が必要だったわけではないので、今回の移植で評価されなかったPF依存種の再現率が85%とは限らない。

また、本評価では(4b)提案手法では抽出不可だが技術的に抽出可能なPF依存部の修正があった。(4b)の例としては、関数に対して仕様で未定義の値が引数として渡されている場合があった。これは、仕様未定義の関数をリストアップしておけばツールで対応可能である。

また、本評価では(4c)完全に対応するのは技術的に困難なPF依存部の修正があった。(4c)の例としては、ROMコードを書き換えている場合があった。const変数を非const変数にキャストしている場合などは文字列検索により抽出可能だが、const変数のアドレスを直接参照している場合などへの対応は困難と考える。

また、提案手法の再現率が85%であり、かつ、「提案手法で抽出可能なPF依存部の修正(4a)」件数(935件)の9割以上を、修正重要度が「3」と「4」のPF依存部が占めていた。(4a)に該当するPF依存部の種類が5種類と少ないため参考情報としてだが、修正重要度の高いPF依存部から修正する方針は正しそうである。

(5) 手動時とツール利用時の再現性の比較

本評価では、手動時とツール利用時とで再現性はほぼ同じであった。これにより、考察(3)で示した「ツールによるPF依存部候補の除外」は、PF依存部を残しつつPF非依存部を除外しており、正しく機能していると言える。

なお、ツール利用時に抽出できなかった9個のPF依存部(= (5a)741 - (5c)732)は、従来通りのテストで検出された。これらは、ポインタ関数など、静的解析エンジンとして用いたcppcheckの仕様が原因で抽出できなかったものであり、ツールの改造により対応可能である。

5.5. その他の議論

(1) ツールの作成、教育 / 習熟コスト

提案手法の場合、ツールの作成、教育 / 習熟コストが増えることが考えられる。しかしこれは最初の1プロジェクトのみに必要なコストであり、次回以降の適用では不要と

なる。

(2) その他の工数削減効果

本評価では、PF 依存部候補の検索を手動で行う場合、PF 依存種の一覧が既にあることを前提にした。PF 依存種の一覧がなく、多数のPF 依存部分を動作確認テストの不具合解析で見出し修正する場合は、手戻り工数が大きくなるため、提案手法によりこの工数の削減が期待できる。

(3) 工数削減以外の効果

提案手法は、既存ソフトウェアのPF 間移植を容易化するだけでなく、ソースコードの移植性を高めることで、実機レス開発の実現 / 導入を容易化できる。

(4) 汎用性

提案手法は、C 言語で記述されたソースコードならば他のPF 間移植にも適用可能である。つまり、SH/ARM から

表 Ex.1 PF 依存種一覧

分類	PF 依存種 (検索対象コード)
社内の過去の移植事例から得たもの	レジスタへの直接アドレス参照
	SRAM、DRAM への直接アドレス参照
	FlashROM への直接アドレス参照
	型のデータ長
	ポインタから固定幅整数型へのキャスト
	明示的、暗黙的な型変換
	char の符号
	符号付き整数のシフト
	エンディアン変換コード
	整数型をバイト配列にキャストしてのアクセス
	unicode 文字列の入出力
	ソケット通信用構造体へのデータ入出力
	キャラクタデバイスとの通信
	バイナリファイルの入出力
	構造体メンバへの直値オフセット参照
	ビットフィールド
	負整数の除算、剰余算
	double のバイト幅
	浮動小数点の整数キャスト時の丸め誤差
	プラグマの使用
アセンブラの使用	
システムコールの呼出	
OS 依存のデータ構造	
MISRA-C より得たもの	自動変数の初期化
	標準識別子、定義、マクロの変更
	オブジェクトの重複領域でのコピー
	31 文字以上の識別子
	複数の異なる外部定義
	シフトによるビット拡張
	関数ポインタのキャスト
	const、volatile 変数のキャスト
	式の実行順序
	型のビット幅以上のシフト
	浮動小数点のビット演算
	インクリメント、デクリメント演算子の演算順
	浮動小数点の比較演算
	異なる配列ポインタの演算、比較
	消滅したオブジェクトの使用
	共用体のメモリ配置

VxWorks/Linux への移植以外にも応用できる。この場合、関連キーワードの設定ファイルを追加 / 変更するだけで良い。

6. おわりに

開発済みの組込みシステムにおいて、PF を現行と同程度のハードウェアリソースを持つ新PF へ変更する際のソフトウェア開発工数削減を目的に、ソースコードのPF 依存部抽出による既存ソフトウェアのPF 間移植手法を提案した。

本手法を用いたPF 依存部抽出支援ツールを開発し、実際の製品ソースコードに適用した。典型的な条件では、PF 依存部の検索・判定・修正工数を49%、PF 依存の該否判定工数を40%削減可能であるとの見込みを得、高い工数削減効果が得られることを実証した。

実際の開発では移植方式の検討にかけられる工数は小さいため、正確で効率的な見積り方法の確立が今後の課題である。

また、複数プロジェクトでの提案方式の適用と評価も必要である。例えば、PF 依存種ごとの対応工数の違い、適合率の違いを分析し、プロジェクトごとに変化が大きい要素と、共通的な要素を分類することで、より正確な修正支援、見積りを行うことが可能となる。

【参考文献】

- [Clements2002] P. Clements and L. M. Northrop, Software Product Lines: Practices and Patterns. Addison-Wesley, 2002.
- [CPC] Sourceforge: Cppcheck.
<http://cppcheck.sourceforge.net/>
- [CPT] Techmatrix: C++Test C/C++ 対応自動テストツール.
<http://www.techmatrix.co.jp/quality/ctest/>
- [ECL] The Eclipse Foundation: Featured Eclipse Project.
<http://www.eclipse.org/>
- [Fowler1999] M. Fowler: Refactoring: Improving The Design of Existing Code, Addison-Wesley (1999).
- [IPA2013] IPA: 2012 年度「ソフトウェア産業の実態把握に関する調査」調査報告書, (2013).
- [METI2010A] 経済産業省: 2010 年版 組込みソフトウェア産業実態調査報告書 - プロジェクト責任者向け調査 -, (2010).
- [METI2010B] 経済産業省: 2010 年版 組込みソフトウェア産業実態調査報告書 - 事業者責任者向け調査 -, (2010).
- [METI2011] 経済産業省: 「平成 22 年度中小企業システム基盤開発環境整備事業 (組込みシステム産業の施策立案に向けた実態把握のための調査研究)」事業報告書, (2011).
- [MSR] The Motor Industry Software Reliability Association: MISRA C.
<http://www.misra-c.com/Activities/MISRAC/tabid/160/Default.aspx>
- [Ochiai2002] 落合竜一, 鈴木正人: リファクタリングとコンポーネント技術による既存ソフトウェアの拡張手法, 情報処理学会研究報告 ソフトウェア工学研究会報告 2002(23), pp.87-94, (2002).
- [PGR] 富士通ソフトウェアテクノロジーズ: PGR Relief C/C++.
[http://jp.fujitsu.com/group/fst/services/pgr/](http://jp.fujitsu.com/group/fst/services/pg/)
- [Pohl2005] K. Pohl, G. Böckle, and F. J. v. d. Linden, Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag New York, Inc., (2005).
- [Selic2003] B. Selic, The pragmatics of model-driven development, IEEE Software, vol.20, no.5, pp.19-25, (2003).
- [Spinellis2006] D. Spinellis: Code Quality: The Open Source Perspective, Pearson Education, (2006), トップスタジオ (訳): コード・クオリティ, 毎日コミュニケーションズ, (2007).
- [Sugiyama2003] 杉山泰一, 携帯電話のバグを無線ネット経由で修復, 日経コミュニケーション, 2003 年 8 月号, pp.70-72, (2003).

システム・ソフトウェア品質標準 SQuaRE シリーズの歴史と概要



早稲田大学 理工学術院 名誉教授
ISO/IEC 25000 SQuaRE シリーズ プロジェクト統括エディタ

東 基衛

1 はじめに

コンピュータ及び通信の高速化、大容量化、画面の高精細化など、情報通信技術（ICT）の急速な発達、ICT 応用システム及びソフトウェア（S&S）の高度化、多様化をもたらした。現在では、S&S は人々の生活の隅々まで浸透し、生活を快適で便利なものになっている。しかしその一方で、S&S 利用によるリスクもまた多様化し、増大している。S&S 利用のリスクの多くは、ソフトウェアの品質不良に起因しており、ソフトウェア品質の向上は現代社会の最重要課題の一つである。

優秀な人材、良い技術・ツールの利用、及び適切な管理によるソフトウェアの品質向上は、**開発者の責任**であり、高品質の S&S の利用は、**取得者・利用者**の権利と責任である。本稿では、ソフトウェアの品質問題の背景と課題、品質向上のための国際標準の例として ISO/IEC25000 SQuaRE シリーズをとりあげ、その歴史、概念及び利用方法を考察する。

2 ソフトウェアの品質問題の背景と標準化の必要性

情報通信技術（ICT）の革新と S&S の多様化、複雑化

ICT の革新は、ほかのどのような産業分野でも経験したことがないように急激なものであり、それ以前には不

可能であった新しい概念に基づく多様な S&S の実現を可能にした。

最初の品質モデル ISO/IEC9126 が刊行された 1991 年から、ISO/IEC25010 SQuaRE 品質モデルが刊行された 2011 年までの 20 年間に達成された ICT の革新は、例えば、マイクロプロセッサの速度は、20MIPS から 50 ～ 100GFlops、ノートパソコンの主記憶容量は 640KB から 4GB、ハードディスク容量は 20MB から 500GB ～ 数テラ Byte にまで急激な進歩を達成し、また 128 ～ 256GB の SSD もハードディスクとして普及している。

通信の進歩も目覚ましい。1991 年に一般の利用者が使用できる通信速度は 9600bps であったが、現在では光通信が普及し、家庭で使用できる速度は毎秒 100Mbps 程度、Wi-Fi は毎秒 60Mbps 程度である。また 1990 年代初頭には、多種のネットワークがインターネットに統合され、通信の利用が飛躍的に便利になった。

S&S の利用しやすさ（Usability）の技術の進歩も見逃せない。GUI（Graphical User Interface）の普及、タッチセンサ式のディスプレイを使用した User Interface は、パソコン、スマートパッド、スマートフォン、デジタルカメラなど多様な製品の急激な発達と普及を促進し、ICT 応用 S&S を一気に拡大し、多様化させた。

S&S の多様化、複雑化と品質向上のニーズ

ICT を活用した多様な S&S は、真に情報化社会を実現し人々の生活を一変させてきた。例えば、Google な

どの情報検索サービス、楽天市場などの e-commerce、YouTube などの動画像サイト、Facebook その他の SNS (Social Networking Service)、Cloud Computing など枚挙にいとまがない。ICT 応用 S&S は、その他の分野でも急速に進化を続けている。自動車交通関連では、カーナビゲーションや衝突被害軽減自動ブレーキシステム、都市広域交通信号制御、ETC など、fMRI (functional magnetic resonance imaging) などの医療機器、自動離着陸などの航空機関連、列車の運行管理、自動改札、ハイビジョンテレビ、DVD プレイヤー、電子レンジ、エアコンなどの家庭用電器製品、ロボットその他のコンピュータ組込み装置など、様々な分野で技術革新が続いている。

その一方で ICT 応用 S&S は、そのシステムの利用による影響、つまりシステム利用のクリティカルリティが一段と大きく、広範囲に及ぶようになり、その品質に不良があれば、利用者やその他の一般市民に多大な影響を与える場合も少なくない。トロイの木馬、その他の悪意による個人情報の盗用、サイバーテロなどによる組織的なシステム破壊などによる損害は少なくない。使い難いシステムによる作業能率の低下、コンピュータ組込み商品の品質不良による商品の売れ行き低下、リコールによる損失なども品質不良の悪影響の例である。一般の利用者も、品質欠陥 S&S 利用により、被害者になったり、加害者になったりすることのないように、高品質の S&S 製品を導入し、とくにセキュリティには十分注意して利用しなければならない時代になっている。

利用者、利用目的が異なる S&S は、当然要求される品質も異なる。例えば航空管制システムやレンタカーの窓口業務のように、利用に先だって利用者の訓練を行うことが可能な場合もある。また、インターネットを通じて

表1 クリティカルリティの分類例

クリティカルリティと重要な品質特性	システムの例
国益・公益クリティカル セキュリティ、信頼性	防衛システム 国家・自治体の予算管理システム
人命クリティカル 正確性、安全性	医療システム 航空機操縦制御システム
社会環境クリティカル 機能性、信頼性、セキュリティ	広域都市交通・電力制御システム 電話交換・銀行システム
企業経営クリティカル 正確性、効率性、セキュリティ	サプライチェーンマネジメントシステム 顧客データベースシステム
利用者の健康・財産クリティカル 使用性、魅力性	一般利用者対話型システム インターネット、Eコマース

買い物を行ったり、劇場の入場券の予約を行ったりする一般の利用者の場合には、利用に先立っての訓練は不可能である。従って S&S の用途と利用者に応じて、必要なユーザインターフェースは異なる。その結果、S&S の品質へのニーズも多様化することになる。新概念の ICT 応用 S&S には、より大規模、複雑、かつ高品質のソフトウェアが必要であり、そのためには、より高度なシステム技術とソフトウェア技術の研究開発を必要とする。

3 ソフトウェア品質標準化のニーズと歴史

システム・ソフトウェアのクリティカルリティと品質特性

以上の例で示したように、現代では多くの品質問題は、「ソフトウェア品質の影響の重大性 (Criticality) の増大に起因する。」といえることができる。クリティカルリティとは、システム・ソフトウェアの使命の重大さを意味する。クリティカルリティには色々あり、品質不良の影響によって分類することが可能である。

例えば、国際的なテロ組織が、国家の政治的な混乱、または経済的な損失を惹起することを狙いとして、国家レベルの情報システムに侵入して、機密情報を盗んだり、システムの破壊を試みたりする例も報道されている。また、インターネットによる通信販売を行う企業の顧客の個人情報を盗み、悪用した結果、情報の盗難に遭った企業の信用が失墜し、業績に影響した例も幾つか報道されている。例えば、国家、行政レベルの国益・公益への影響、多くの人命・健康被害への影響、交通渋滞・停電などの社会環境への影響、企業の信用・業績への影響、利用者個人レベルの健康・作業能率・財産などへの影響などに分類することができる。表1に筆者がクリティカルリティの分類を試みた例を示す。

S&S 品質関連の国際標準と SQuaRE シリーズ

高品質の S&S を開発、保守するための技術を標準化する試みが、これまで多くの国際組織により行われてきた。広く一般製品のための品質関連の国際標準としては、ISO 9000 シリーズが広く知られている。ISO/TC176 技術委員会は、1979 年に組織され、最初に、ISO 8402:1986 (Quality Vocabulary) が制定された。ソフトウェア関連では、ISO 9000 シリーズの一環として、

ISO 9000-3 (Quality management and quality assurance standards -- Part 3: Guidelines for the application of ISO 9001:1994) が 1997 年に発行された。この業務はその後 ISO/TC176 から ISO/IEC JTC1/SC7 に移管され、ISO/IEC 90003:2004 (Software engineering -- Guidelines for the application of ISO 9001:2000 to computer software) として発行されている。このほか、各種のコミュニティが品質に関連する国際標準の制定を行っている。例えば、IEC TC56 は、一連の Dependability Standards を発行している。また、ISO/IEC JTC1/SC27 IT Security techniques は、ISO/IEC 27000 シリーズ Standards for information security management systems を中心に制定を行っている。ISO TC159/SC4 Ergonomics は人間工学の視点から、システムの Usability に関する各種標準の制定を行っている。これらの標準化活動は、航空、電力、自動車、情報システムなど対象業界を規定している団体と、システム工学、安全工学、ソフトウェア工学、信頼性工学、セキュリティなど業界に横断的な、技術をベースにしている活動に分けられる。

SQuaRE シリーズの前身である ISO/IEC 9126 品質モデルは、日本の品質展開、ベームなどの品質モデルその他に啓発されて、筆者が当時の ISO/TC97/SC7 に提案して採択された。最初の品質モデルは、6つの品質特性とその定義から構成され、ISO/IEC JTC1/SC7/WG6 となった後の 1991 年に ISO/IEC 9126:1991 として発行された。その後品質モデル単独では使用できないという意見により、ISO/IEC 9126 Software engineering – Product quality シリーズ全 4 巻、及び ISO/IEC 14598 Software engineering – Product evaluation シリーズ全 6 巻が提案され刊行された。

4 SQuaRE シリーズの概念と概要

ISO/IEC 25000 SQuaRE シリーズは、ISO/IEC 9126 シリーズ、及び ISO/IEC 14598 シリーズの完成の目途がついたため、それらの統合と改定を目指して、1999 年の SC7/WG6 金沢会議で筆者が提案、検討の結果、2000 年の JTC1/SC7 マドリッド会議に WG6 から提案し承認された。この提案は ISO 本部の ITTF でも歓迎され、ISO/IEC 25000 から 25099 までの番号を SQuaRE シリーズのために自由に使用してよいという決定がくださった。

品質の定義

ISO/IEC JTC1/SC7/WG6 では、ISO/IEC9126 の当初より品質 (Quality) の定義を、ISO8402 Quality Vocabulary を出発点としている。この定義は SQuaRE シリーズにも継承され、ソフトウェア品質を次のように定義している。ここで異なる利害関係者にはその人のプロフィール、用途に応じて、同じ製品の品質に対して異なるニーズがあることに注意しなければならない。

ソフトウェア品質：明示された状況下で使用されたとき、明示的ニーズ及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。(JIS X25010 品質モデル)

また、明示されたニーズだけではなく暗黙のニーズ、つまり利用者が伝えられないニーズも満足させることが要求されていることに注意が必要である。品質は、例えば品質特性、及び品質副特性を定義した「JIS X25010 品質モデル」を用いることにより、詳細に定義することが可能である。

SQuaRE シリーズの構成と S&S 品質向上の着眼点

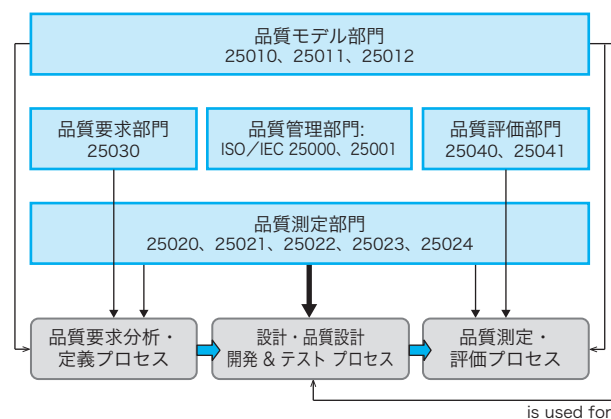


図1 SQuaRE シリーズの構成とその利用

SQuaRE シリーズは、図1に示すように 2500n: 品質管理部門、2501n: 品質モデル部門、2502n: 品質測定部門、2503n: 品質要求部門、及び 2504n: 品質評価部門の5つの部門、及び 2505n~2509n までの拡張部門から構成される。

S&S 製品の品質を向上するためには、利害関係者のニーズを選別して、機能要求に加えて、ISO/IEC 25010,25012 品質モデル及び 25021~25024 品質測定法を用いて、品質要求を品質評価基準と共に定義することが重要である。利害関係者、とくに利用者のニーズは、

製品の利用効果への期待であると考えられる。利害関係者の選定、及びそのニーズの収集と選別に当たっては、対象 S&S 製品が利用者が特定可能な二者間契約に基づく受注 S&S 製品の場合と、流通ソフトウェアパッケージや SaaS などを含めて、不特定多数を対象にする既製 S&S 製品では大きく異なるので注意が必要である。品質要求に関しては既刊の 25030:SQaRE- 品質要求を参照するとよい。また、既製ソフトウェア製品には「ISO/IEC 25051:2014 SQaRE – 既製ソフトウェア製品 (RUSP) に対する品質要求事項及び試験に対する指示」を参照するとよい。図 2 に品質向上の着眼点を示す。

次に開発中は、開発の各工程で、設計レビューや単体テスト、総合システムテストなどのあらゆる機会に、品質メジャー及び品質評価基準を用いて品質測定及び品質評価を行い、必要により早めの修正活動を実施することが必要である。そのために SQaRE シリーズでは、25021:QME (品質測定要素)、25022: 利用時の品質測定法、25023: 内部品質測定法 (本稿では品質測定量及び測定方法を合わせて品質測定法と書く)、外部品質測定法、及び 25024 データ品質測定法などを準備している。品質測定法の詳細は、この特集の品質測定部門に関する記事を参照されたい。

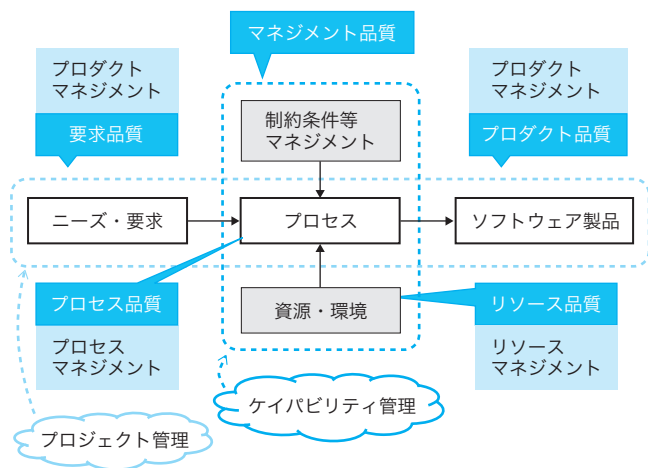


図2 S&S 品質向上の着眼点

システム&ソフトウェア品質のライフサイクルと視点

S&S 製品の品質は、それがライフサイクルのどの段階にあるかによって異なる。図3はシステム&ソフトウェア品質ライフサイクルを示す。一般にシステム・ソフトウェア開発は要求分析・定義から始まると考えられるこ

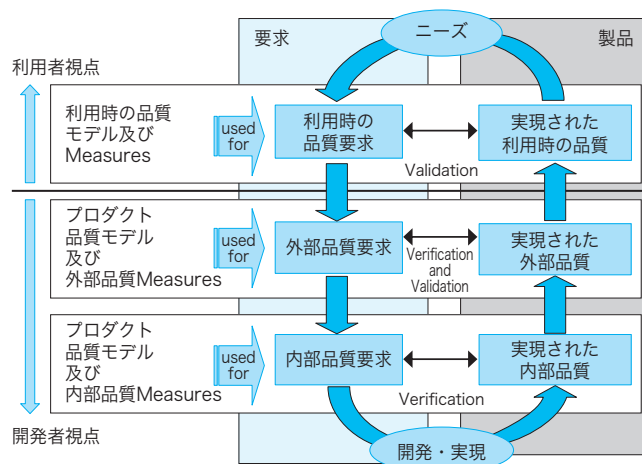


図3 システム&ソフトウェア品質ライフサイクル (ISO/IEC 25000 に基づく)

とが多い。しかし現実には、現在使用中のシステムまたはソフトウェア製品の品質 (Quality in Use) に対しての改善ニーズから始まることが多いと考えられる。

S&S 製品を実際に使用した際に利用者が感じる品質は、開発時に評価した品質とは区別して「利用時の品質」(Quality in Use) といい、広い意味での利用者への影響を調べ、測定することにより把握される。利用者のニーズは、先ず利用時の品質に対するニーズとして捉え、次に S&S 製品の品質に対するニーズに変換する必要がある。製品が顧客や利用者の開発要求に基づいて開発される場合でも、また、製品企画担当者によって計画される不特定多数の利用者向けの製品の場合でも、要求仕様段階では、利害関係者のニーズを要求仕様としてまとめたものにすぎない。

5 SQaRE シリーズの品質モデルとその応用

品質モデルは JIS X25000 によれば、「品質要求事項の仕様化及び品質評価に対する枠組みを提供する特性の定義された集合及び特性間の関係の集合。」と定義されている。またソフトウェア品質特性は、「ソフトウェア品質に影響を及ぼすソフトウェア品質属性の分類。」と定義され、「ソフトウェア品質特性は、複数の階層の副特性に詳細化され、最終的にはソフトウェア品質属性にまで詳細化することができる。」と注記されている。

最初の品質モデルは、1991年に ISO/IEC 9126: Information technology- Software product evaluation- Quality characteristics and guideline for their use として

発行された。その後 ICT の革新と時代の品質ニーズを反映して、2001 年には品質副特性及び利用時の品質特性を追加して、ISO/IEC 9126-1-Software engineering, Product quality- Part 1: Quality model として発行された。更に 2011 年には、インターネット利用の普及によるセキュリティ及び相互接続を重視して、6 品質特性を 8 品質特性にして、品質副特性もかなり変更を加えて、ISO/IEC25010:2011- Systems and software engineering- Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models として刊行された。25010 の製品品質モデルの品質特性は、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、移植性である。各品質特性はそれぞれ品質副特性に展開されている。また、利

用時の品質モデルにも大きな変更を加えており、有効性、効率性、満足性、リスク回避性、利用状況網羅性から構成される。有効性、効率性を除く 3 特性には、新たに品質副特性が定義された。

前述のように、品質要求は品質モデルをチェックリストとして用いて、対象 S&S の用途、利用者及び特徴を考慮して、品質測定法を用いた品質評価基準と共に、明確に定義するとよい。品質モデルの対象の概念図を図 4 に、システム・ソフトウェア製品品質モデルを図 5 に示す。しかしながら、上述のすべての品質特性に対応する品質要求を定義し、測定、評価するのは、コスト及び時間を考慮すると現実的ではない。従って、筆者の場合には、先ず利用時の品質から要求を検討して、次に製品品質モデルの品質特性及び副特性には優先順位をつけて、要求定義、測定、及び品質評価を実施することを推奨している。

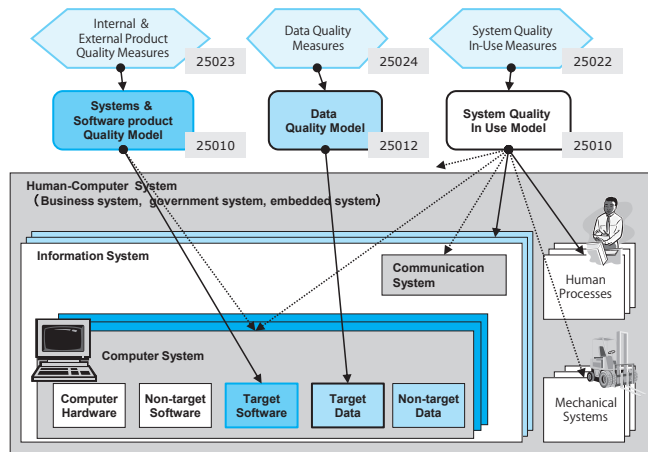


図 4 品質モデルの対象と品質メジャー

6 おわりに—ソフトウェア品質の今後の課題と取り組み

システム及びソフトウェア製品の品質不良は、多様なリスクの原因である。これまで品質向上の施策を、プロジェクトに横断的な支援組織の活動、及びプロジェクトの活動に分けて解説した。具体的には、品質モデル、品質測定法を活用して、品質要求を定義し、プロセスを設計・実施し、品質の定量的な評価を実施することと考える。より具体的な方法は本特集のほかの解説を参照していただきたい。

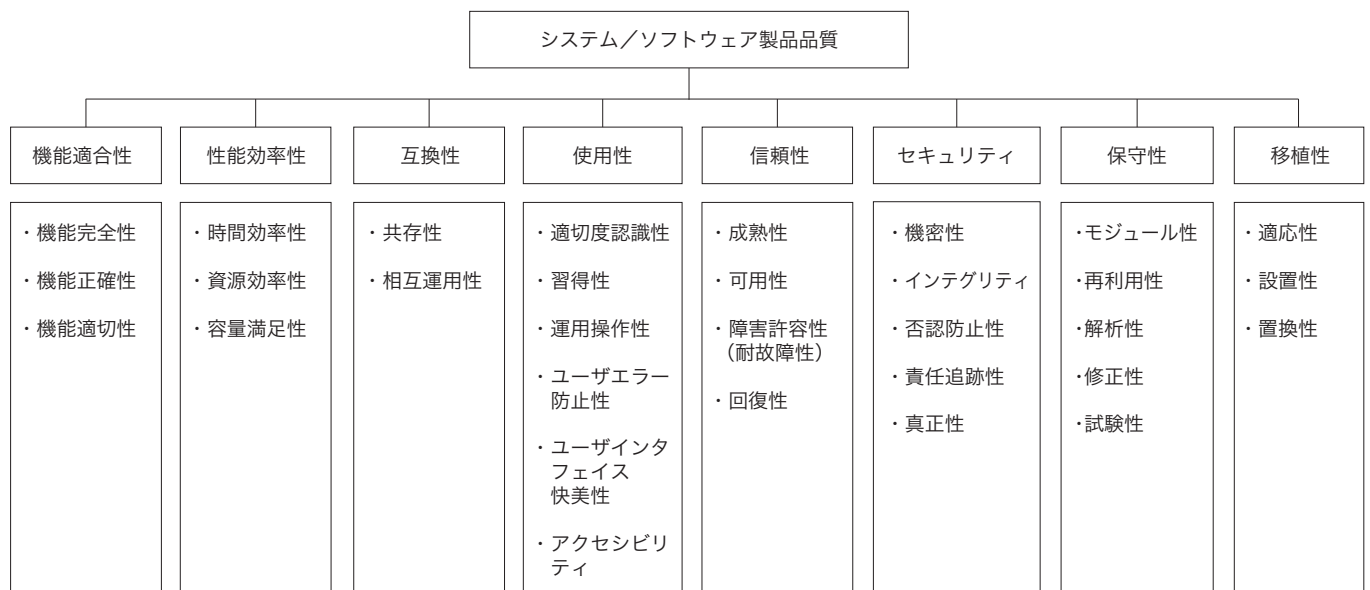


図 5 システム・ソフトウェア製品の品質モデル (JIS X 25010:2013 (IEC25010:2011) に基づく)

システム及びソフトウェア品質の 要求定義と評価のプロセス



日本電気株式会社 ソフトウェア生産革新本部 エグゼクティブエキスパート
ISO/IEC JTC1 SC7/WG6 Convener

込山 俊博

システム及びソフトウェア（S&S : Systems and Software）の品質向上では、要求される品質を仕様化し、要求品質を実装するプロセスをデザインし、開発された S&S の品質を評価して制御することが重要である。本稿では、国際規格 ISO/IEC 25000 SQuaRE^{※1} シリーズに基づき、これらの活動に取り組む上でのポイントを解説する。また、S&S 品質の要求定義と評価の組織的なプロセス改善に言及する。

1 はじめに

システム及びソフトウェア（S&S）の品質向上には、対象とする S&S に求められる品質の仕様化、実装、制御の3つの側面から取り組むことが重要である。ここで言う仕様化とは、顧客要求や組織目標に基づいて品質要求事項を定義することであり、実装とは、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あり、制御とは、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。

S&S の品質は、要求した機能が実装できてさえいればよい（機能性）、故障せずに動作しさえすればよい（信頼

性）というものではない。当該 S&S の使用目的や競合製品との差異化など考慮して、利用者にとっての使いやすさ（使用性）、処理要求から結果受理までの速度（性能効率性）といった品質特性が求められる場合もある。従って、仕様化局面では、求められる品質特性に照らして多角的な観点から品質要求事項を定義し、実装局面では、定義された品質要求事項を充足する設計・製造を行い、制御局面では、品質特性ごとの品質要求事項の充足を測定、評価することが必要である。

図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。

SQuaRE においては、品質の仕様化に関して、“ISO/IEC 25030（JIS X 25030）：品質要求事項”で品質要求事項の定義に関する基本的な考え方と作業のポイントを規定している。品質の制御に関して、“ISO/IEC 25040（JIS X 25040）：評価プロセス”で品質評価の普遍的なプロセスを規定しており、“ISO/IEC 25041：開発者、取得者及び独立した評価者のための評価手引”で立場に応じた品質評価作業のガイドを提供している。

また、品質要求事項の定義並びに品質評価を、多角的かつ網羅的な観点から定量的かつ客観的に実施するため、“ISO/IEC 2501n：品質モデル部門”で S&S が考慮すべき品

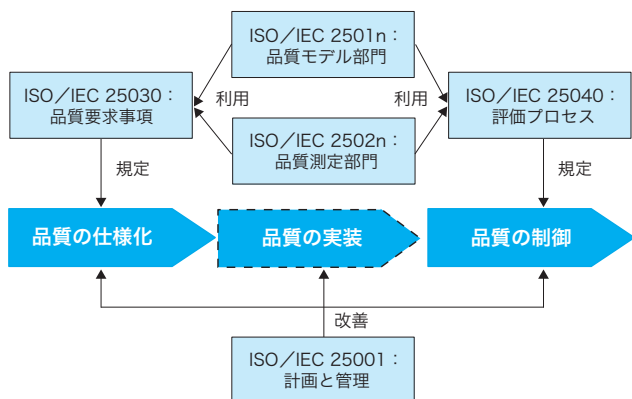


図1 品質の仕様化、実装、制御における SQuaRE の活用

【脚注】

※1 Systems and software Quality Requirements and Evaluation

質特性及び品質副特性を品質モデルとして規定すると共に、“ISO/IEC 2502n：品質測定部門”で各品質（副）特性の品質測定量（Quality Measure）を規定している。ただし、品質の実装に関する規定やガイドまでは、現段階ではカバーしていない。

更に、SQuaRE では、品質の仕様化、実装、制御に関する知識や経験を蓄積し、改善サイクルを回して、それらをより効果的に実践するための考え方や活動を“ISO/IEC 25001（JIS X 25001）：計画と管理”で規定している。

本稿ではまず、SQuaRE に基づき、品質の仕様化、及び品質の制御の実践ポイントを説明する。次に、仕様化された品質要求に基づく品質の実装における考え方を示す。最後に、品質の仕様化、実装、制御という一連の活動の改善について言及する。

2 SQuaRE における品質の仕様化

SQuaRE では、S&S の品質要求事項を表 1 の枠組みでとらえている。

システム要求事項を定義する際には、対象とするシステムの範囲を特定することが重要である。昨今の ICT（情報通信技術）応用システムでは、個々のシステムがそれぞれ単独に動作するケースはまれで、様々な通信を介して協調的に動作する。IoT（Internet of Things）の時代を迎え、その傾向や形態の多様性に更に拍車がかかることであろう。その際、開発や評価の対象とするシステムと連携する他システムとの境界を明確にし、どの範囲で要求事項を定義するのかを特定することが重要である。

その上で、システムレベルの要求事項は、システムを構成する個々のコンポーネントに対する要求に展開される。システムのコンポーネントとしてのソフトウェアへの要求事項は、製品に対するものと、それを開発する組織やプロセスに対するものに分類され、製品に対する要求事項は、ソフトウェア固有の特徴と割り当てられた特徴（価格、配付日など）に分類される。ソフトウェア固有の特徴は、ソフトウェアの機能に対するものと品質に対するものに分類

される。機能的な特徴に対する要求事項は、入力を出力に変換するソフトウェアの動作に関するものであり、品質面の特徴に対する要求事項は、その機能に付随する特質（feature）に関するものである。

通常、ソフトウェアの仕様化に関して、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。まずは、ISO/IEC 25030 を参考に、S&S の品質要求事項の仕様化を確実に実施することが肝要である。

システムを構成するソフトウェアの品質要求事項の仕様化に際しては、ソフトウェア自体に求められる内部（静的な）品質要求事項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）品質要求事項を、ISO/IEC 25010 の S&S 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、S&S が利用者を与える影響に関しては、ISO/IEC 25010 の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。更に、S&S の処理対象となるデータに関しては、ISO/IEC 25012（JIS X 25012）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 2 に示すように、異なる品質モデルの品質（副）特性の関係性を考慮して整合を図ることが望まれる。例えば、利用時の品質モデルで、S&S の利用者がそれを用いることで作業の効率がどの程度向上するかを扱う“効率性”を重視するならば、S&S 製品品質モデルでは、S&S の動作の効率を

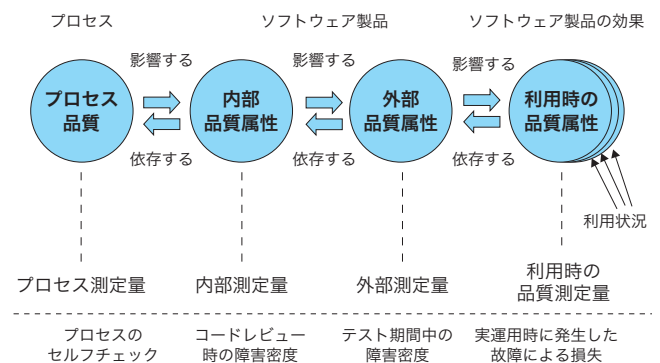


図 2 ライフサイクルでの品質（JIS X 25010）

表 1 S&S に対する要求事項の構造（JIS X 25030）

システム要求事項	ソフトウェア要求事項	ソフトウェア製品要求事項	固有の特質への要求	機能的な要求事項			
				ソフトウェア開発要求事項	ソフトウェア開発組織要求事項	ソフトウェア品質要求事項	利用時の品質要求事項
						ソフトウェア品質要求事項	外部品質要求事項
						ソフトウェア品質要求事項	内部品質要求事項
その他のシステム要求事項	例えば、コンピュータハードウェア、データ、機械部品及び人手による業務プロセスへの要求事項を含む	割り当てられた特徴の要求事項	例えば、価格、配付日、製品の先行き及び製品の供給者に対する要求事項を含め、管理面での要求事項				

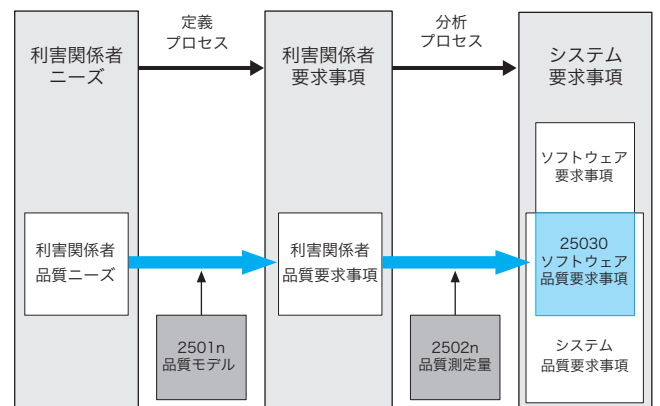


図 3 S&S の品質要求の定義と分析（JIS X 25030）

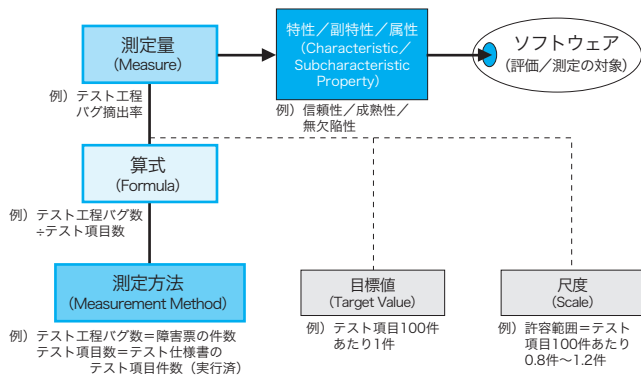


図4 品質要求事項定量化の考え方

扱う“性能効率性”やS&Sの操作のしやすさを扱う“使用性”を重視すべきであろう。

図3に品質モデルと品質測定量を利用したS&Sの品質要求の定義と分析の流れを示す。品質要求事項の定義においては、まずはS&Sの利用、またはS&S自体の機能や品質に対するニーズを有する利害関係者を識別することが必要である。利害関係者は、S&Sの開発に関与する開発者と発注者だけでなく、S&Sの直接、間接の利用者を含め、共通の属性を有するグループに分けて識別するとよい。これら利害関係者の個々のグループが、S&S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事項として仕様化するとよい。この際、相反する品質要求が識別された場合、あるいはS&S開発における費用、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、利害関係者間で協議を行い、調整を行う必要がある。

仕様化され、合意された利害関係者の品質要求事項に対しては、ISO/IEC 2502nの品質測定量などを参考にして、その充足の度合いをどのような品質測定量に照らして評価するかを明確にしておくこととよい。個々の品質測定量に対しては、図4に示すように、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とよい。目標値や尺度の設定においては、過去の実績データの収集・分析によるのが望ましい。しかしながら、利用可能なデータが欠如している場合には、データ白書や事例報告などに示されたものをまずは参考値として設定し、データの蓄積に伴って見直すのも一考である。

3 SQaREにおける品質の制御

SQaREにおけるS&Sの品質評価プロセスを図5に示す。

以下、品質評価プロセスの各アクティビティ実践の要点を解説する。

① 評価要求事項の確立

ここではまず、評価の目的を明確にする。典型的なS&S評価の目的には次のようなものがある。

- 開発した成果物が要求を満足しているかを確認する
- 候補製品を比較選定する

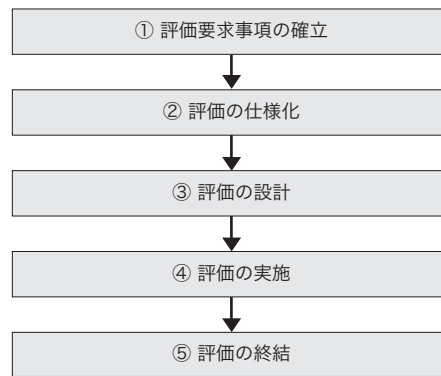


図5 SQaREの品質評価プロセス (JIS X 25040)

— 第三者（品質保証部門、試験機関など）が利用者の立場で客観評価を行う

その上で、仕様化された品質要求事項を取得する。ここで評価目的に照らして、個々の品質要求事項の充足を評価する対象を特定する。既製S&S製品の比較選定であれば、評価の対象は出来上がったS&S製品となるであろうが、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には、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S&Sのコンポーネント（サブシステム、モジュールなど）も評価対象になるであろう。これらS&Sの構成要素の評価に関して、共通的に考慮すべき品質特性とその品質要求事項もあれば、通信系のモジュール、ユーザインターフェイス系のモジュールなど、構成要素の種類に応じて考慮すべきものもある。このような点を加味して、評価対象ごとの評価の視点を明確にする。

個々の評価対象の品質評価では、品質特性の重要度と評価にかかる工数や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必要がある。

② 評価の仕様化

ここでは、仕様化された品質要求事項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。①で検討した評価目的、評価対象、評価の厳密さに照らして、品質要求事項の定義で設定した品質測定量と評価基準がそのままS&S及びその構成要素の評価仕様として利用可能かを検討し、必要に応じて見直しを行う必要がある。

また、個々の品質測定量の良否判定に加え、加重平均などの考え方をういて結果を集約し、工程終了、出荷、受領などの可否判定に用いる総合的な評価基準を設定する必要がある。典型的には、品質（副）特性ごとの評価基準、品質の総合的な評価基準、品質以外の管理ファクタ（納期、コストなど）を加味した総合評価基準を設定する。

③ 評価の設計

ここでは、実施する品質評価の計画を立案する。SQaREでは、品質評価計画のテンプレートをISO/IEC 25001で規定しており、以下の事項を記載することを推奨している。

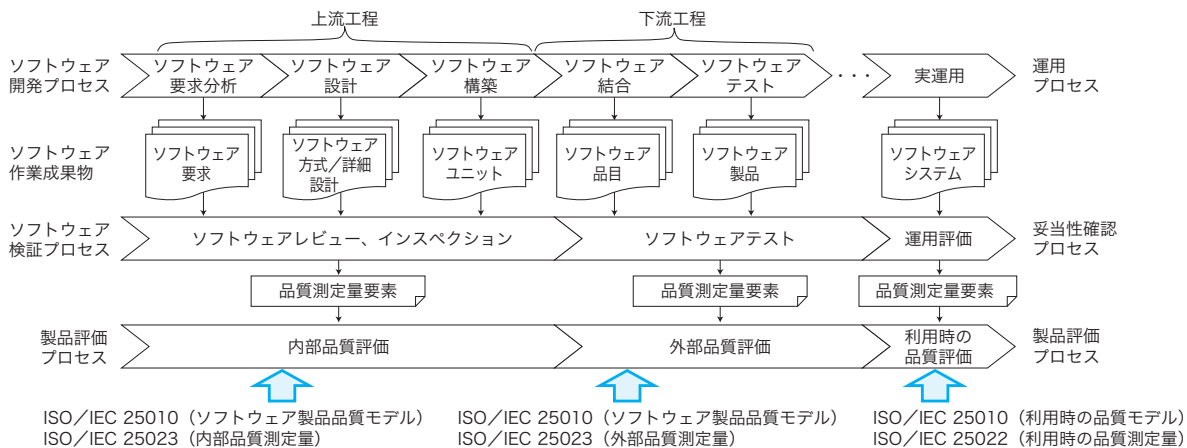


図6 ソフトウェアのライフサイクルと品質評価

- 評価の目的
- 評価に関連する組織
- 利害関係者の責任と権限
- 評価の予算
- 評価のスケジュール
- 評価の環境
- 評価の方法とツール
- 評価の基準

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が評価の目的である場合には、図6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とよい。評価の方法とツールに関しては、SQuBOK^{※2}、SWEBOK^{※3}などの知識体系を参照するとよい。

④ 評価の実施

ここでは、③の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。しかしながら、測定値単独では、その値の良否判定はできない。目標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（図7参照）。更に、品質（副）特性別の評価、品質全体の評価、

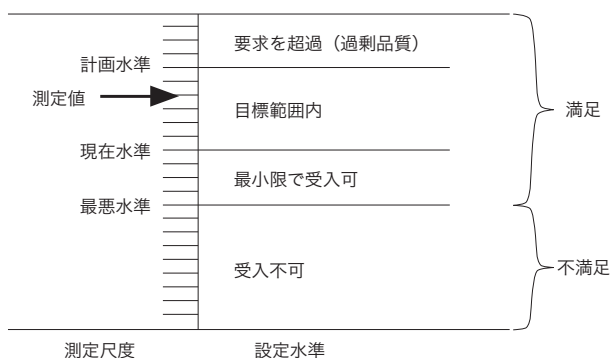


図7 品質測定量の評価イメージ (JIS X 0133-1)

品質及びその他管理ファクタの総合評価を行うための集計ルールにそって測定値を集計し、それらに対して設定された評価基準（目標値や尺度）に照らして評価を行い、判定を下すとよい。

⑤ 評価の終結

一連の計画や結果は評価報告書に記録し、少なくとも評価の依頼者と評価者の間で共有し、共同で審査を行うとよい。

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S&Sの修正などの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を記載して、測定方法や評価基準の見直しを行う。

収集したデータは、機密性を考慮した上で、保管、破棄など扱いを明確にし、然るべき処理を行う。

4 品質の実装

品質の仕様化で、S&S及びその構成要素で重視する品質特性と具体的な品質要求事項を定義したなら、それらを開発する過程でいかにして求められる品質を実装するかを、プロセスデザインの視点から検討する必要がある。

システム及びそれを構成する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。そ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。

重視する品質特性と具体的な品質要求事項が定義された

【脚注】

- ※2 Software Quality Body Of Knowledge
- ※3 SoftWare Engineering Body Of Knowledge

なら、これらのライフサイクルプロセスに照らして、重視すべきプロセス、アクティビティ、タスクを特定すべきである。例えば、S&S 製品品質モデルで、求められる機能の充足を扱う“機能完全性”を重視するのであれば、構成管理プロセス並びにテクニカルプロセスに埋め込まれた追跡可能性に関連するタスクの実装を重視すべきであろう。また、成果物に欠陥を含まないことを扱う“成熟性”を重視するのであれば、テクニカルプロセスやソフトウェア固有プロセスに含まれるレビュー、検証、妥当性確認、適格性確認などのプロセスの実装を重視すべきであろう。

ただし、これらのライフサイクルモデルでは、S&S の企画、開発、運用・保守の過程で“何を（What）”実施するかを規定しているが、それを“いかに（How）”実施するかまでは規定していない。品質の実装の観点からは、重視するプロセス、アクティビティ、タスクで、どのような技法やツールを活用するかを特定して、より実践的な品質作り込みの手段を特定することが重要である。

現段階では、品質特性格の標準的なプロセスデザインを規定したものはないが、How 発想でのプロセスデザインには、SWEBOK、SQuBOK などの知識体系が有用である。例えば SQuBOK では、表 2 に示す通り、使用性、セーフティ（SQuaRE では、利用時の品質モデルの“リスク回避性”に対応）、セキュリティに対応した技法を掲載している。

表 2 品質特性の実装に有効な技法

品質特性	使用性	セーフティ	セキュリティ
技法	<ul style="list-style-type: none"> ●ユーザビリティテスト ●インタラクティブシステムの人間中心設計プロセス 	<ul style="list-style-type: none"> ●セーフティ実現のためのリスク低減技法 ●セーフティクリティカルシステムのテスト 	<ul style="list-style-type: none"> ●セキュリティ要求分析 ●セキュリティ設計 ●セキュリティパターン ●セキュアコーディング ●セキュリティテスト

5 SQuaRE における品質要求定義と評価のプロセス改善

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。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。

SQuaRE では、この技術管理の活動を、ISO/IEC 25001 に規定している。図 8 に、組織とプロジェクトが連携して遂行する SQuaRE の技術管理活動の流れを示す。図 8 に示された個々の評価プロジェクトが遂行する活動は、図 5 に示された ISO/IEC 25040 の評価プロセスに該当する。

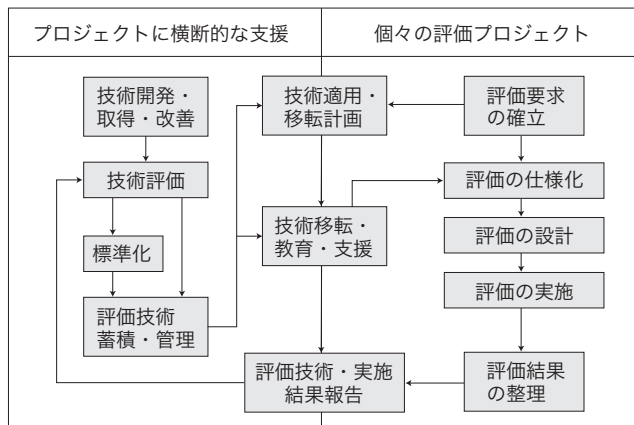


図 8 SQuaRE の技術管理のフロー (JIS X 0133-1 の解説)

SQuaRE の考え方に基づいて品質要求定義や品質評価を実施する場合、開発や評価を行う S&S の特徴に合った品質測定量を設定し、その測定方法並びに良否判定の基準を実務レベルで設定する必要がある。このような取り組みはプロジェクト個別に行うより、組織に集約するのが効率的である。

品質測定量の設定は、ISO/IEC 2502n から取捨選択し、不足分を独自開発するなどで行う。製品の規模、欠陥数など、その算出に必要な品質測定量要素のカウント方法を、プロジェクトによってぶれが出ないように明示的に定義する。また、それらを組み合わせると算出された品質測定量の値の適否を判断するための目標値や尺度を、これまでの実績データをプロジェクト横断的に分析して組織としての基準を設定する。また、プロジェクト側の評価プロセスで実践した結果を組織側にフィードバックし、測定方法や基準の見直し、陳腐化した技術のリプレイスなど、評価技術の改善を検討する。これらは、S&S の品質要求定義と品質評価の組織知として標準化し、組織側からプロジェクトへの教育や支援などによって適用や技術移転を促進する。

6 おわりに

本稿では、S&S の品質の仕様化、実装、制御及びその改善について、プロセス視点からの解説を行った。SQuaRE の提唱する S&S の品質要求定義と品質評価の考え方は、国際的に広く受け入れられている。2015 年中には、S&S とデータの品質測定量を規定した ISO/IEC 25022,23,24 が発行される予定である。更に、S&S 製品を利用した IT サービスの品質モデルと品質測定量の制定も進みつつあり、今後 SQuaRE がますます普及していくと考えられる。

今後も更に進化し、高度化して行くであろう S&S 製品並びにそれを活用したサービスを、安全・安心に利用するためのキーファクタは品質である。SQuaRE を活用して、それらに求められる品質を多角的かつ定量的に定義し、評価することによって、品質の確保、向上を図ることが一層重要になる。SQuaRE の制定に携わる立場から、その普及にも尽力する所存である。

ユーザーエクスペリエンスを指向するサービスの開発と運用に向けて



小樽商科大学商学部社会情報学科 教授
ビジネス創造センター UX 研究部門長

平沢 尚毅

近年のIoTなどの技術環境を応用し、利用者や社会に有益なサービスを構築するには、新たなユーザーエクスペリエンスを構想するアプローチが有効である。更に、このサービスを安定して運用するためには、利用者側から見たサービス品質をマネジメントすることが欠かせない。

1 ユーザーエクスペリエンスとは何か

本学のビジネス創造センターの下にユーザーエクスペリエンス（以下UX）の向上をねらいとして、製品あるいはシステム開発を支援するために、UX研究部門を設置したのが、2007年であった。当初に比べると、このUXという言葉は徐々に浸透し、事業組織名にする企業も見られるようになった。

現在、このUXを定義している規格が、ISO9241-210[1]である。ここでは、「製品、システム、サービスの利用の結果もたらされる人間の認識と反応」と定義している。更に、3つの注意書きがある。まず、UXが、利用者の情動などの人間の内面に生じた経験から、行動のような外面的な活動などすべてを含むものであること。次に、UXが、システムの利用の結果であること。そして、ユーザビリティの基準が、UXのある側面の評価に応用できること、などが記載されている。この規格には、ユーザビリティとUXの違いについては明記さ

れていないが、ユーザビリティに比べて、UXが利用者のより広範な行動を対象としていることがわかる。UXの定義は、国際会議[2]の場でも統一した見解に達することができなかったように、現在も発展段階にあると言える。

そもそもUXという考え方は、HCI（Human Computer Interaction）の世界では、1990年代にNormanらによって提案された概念であったと言われている。当初は、デザインが引き出す情緒的な側面を明確にするために提案されたものであったとされているが、現在は業界によって様々な使われ方がなされているのが実情である。

更に、経営学の分野でもUXにかかわる概念が広く利用されている。Schmitt[3]は、従来のマーケティングの領域を超える方法として経験価値マーケティングを展開した。また、Pine[4]は、経験経済という考え方を通して、製品のコモディティ化を防ぐために、経験を価値として利用する考え方を提案している。

このように、デザインにおける設計概念であったUX

は、経営学などの他領域と連動して現代社会にとって重要な概念ともなっているともいえる。UX という考え方は、技術やサービスでのイノベーションを実現するための、基本的なマインドセットを形成する上での重要な役割を担っていると言える。

2 UX がもたらした結果

2.1 iPod の出現

近年、UX を意識させた事件は、Apple 社の iPod^{※1} の発売ではなかったと思われる。デジタル機器だけではなく、ソフトウェアを利用して曲単位の音楽デジタルコンテンツを購入できるという新たなビジネスモデルも驚くべき発想であった。iPod が発売された当時は、ユーザビリティに優れた製品という見方があったが、それだけでは製品の特徴を隈取りできないものであった。斬新なユーザインターフェースに目がゆきがちであったが、実際は、もっと広い意味で、イノベーションを起こしていたことが理解できる。Apple 社が、それに続いて、iPad^{※1}、iPhone^{※1} といった画期的な製品シリーズを打ち出すことができた背景には、新たな UX を創出することができる基盤があったことが推測できる。

2.2 Web サイト構築

Amazon 社の書籍販売サイトには、ショッピングカート機能が設定されていた。これは、ビジネスモデル特許の文脈で議論されることがあったが、この機能は、UX の成果と見るべきものである。その後も、ウィッシュリストなど、Web サイトを通じて様々な UX を提供する機能が提案されている。

Web サイトの業界では、Web2.0 というコンセプトに代表されるように、様々なサービスが提案されていった。これらは、まさしく UX を広く認知させていった。とくに、IA (Information Architect) と呼ばれる技術者は、一貫して UX の観点からサイト開発を推進してゆくことを可能にした。Garrett の『ウェブ戦略としての「ユーザーエクスペリエンス」』[5] は、その代表的な著作である。

2.3 情報サービスにおける変化

情報サービスの利用者は、日々の生活で使いやすい

Web サイトの利用に慣れ親しんでいくうちに、社内システムのユーザインターフェースへの要求は、自然と高くなる。多様な国民を対象とした、電子政府関連のシステムの場合は、より高いユーザビリティが求められるのは、尚更である。

一方、各国の電子政府のあり方は劇的に変化している。2000 年当初は、ユーザビリティの課題が指摘されていたが、最近では、UX が電子政府の重要な課題として取り上げられている。例えば、英国政府が開発した GOV.UK[6] の UX の変革は劇的なものである。電子政府の指針を Web サイトによる行政手続きから、より広範なデジタルサービスを提供するように変化させている。そして、千を超える手続きを一つのサイトに集約している。こういった変革を担う UX 専門組織も整備されている。一方、我が国では、2014 年に今後の IT 国家戦略の方針である『世界最先端 IT 国家創造宣言』[7] が閣議決定されたが、UX に関する記述は見あたらない。今後、マイナンバー制度の導入が予定されており、国民視点から UX の指針を明確にする必要があるはずである。

2.4 新たなサービスの出現への期待

ICT の進展によって、様々な革新的なサービスが構想できる可能性が広がっている。

医療分野であれば、IoT を利用して、身体に装着したデバイスから、自動的に計測データを送信し、それを分析することによって健康状態をモニターできる。このデータを利用できれば、適確な診断に役立てることもできる。

また、移動手段についても大きな変革が起こっている。自動車の『移動する』という機能に特化したカーシェアリングサービスが広がりつつある。更に、UX を進展させると、自動車をシェアするのではなく、自動車の空き席をシェアするサービスも考えられる。実際、国外のリゾートには、そのようなサービスも出現している。

UX を中心に技術を展開していくと、このように新たなサービスアイデアが生まれる可能性が高まることを期待できる。

【脚注】

※1 Apple 社のデジタル製品の商標名

3 UXアプローチの現況

3.1 人間中心設計プロセス

新たなUXを指向するサービスを開発する方法は、人間中心設計が基本である。人間中心設計の基本的な考え方及びプロセスは、前述のISO9241-210に整理されているが、次の4つのプロセスが柱となっている(図1)。

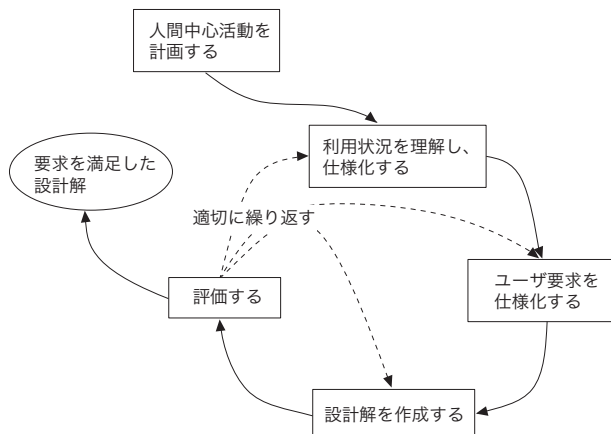


図1 人間中心設計プロセスの相互関係 (ISO9241-210より)

- ① (サービスの) 利用状況を理解し、仕様化する。
- ② ユーザ要求を仕様化する。
- ③ 設計解を作成する。
- ④ (設計解をユーザ要求に基づいて) 評価する。

これらのプロセスは、必ずしも、この手順で進めるものではない。そもそも、単独で人間中心設計プロセスは存在するものではなく、現行の開発プロセスへ統合し、UX構想及び開発を強化することを意図しているものである。例えば、組込みシステムであれば、IPA/SECが開発したESPRモデル[8]に統合して各事業の開発環境に応じてテーラリングして新しいプロセスを構築できる。

技術的に設計案が先行している場合でも、設計案の背景となる②のユーザ要求を想定することに着手することから始める。これを省略すると、④の評価(この場合は、妥当性確認)ができない。設計段階での妥当性確認が行われない場合は、実装するまで、ユーザの観点からの評価が行われず、実装後に、問題が見つかった場合の損失は大きい。

このように、人間中心設計は、常に①の利用状況の調

査から始めるとは限らない。最初に③の設計解のビジョンを構想して、②のユーザ要求を詳細化してゆくアプローチもある[9]。この方法では、利用者のシナリオを段階的に詳細化しながら、最終的にUI仕様を確定する。

欧州では、新たなUXを構想した、サービスを開発する方法について議論が行われている[10]。ここでの方法は、人間中心設計をベースにしながら、様々な手法を応用して、サービスの実装及び運用にも言及している。

3.2 ビジネスモデルとの連携

人間中心設計によって、効果的なUXサービスが技術的に可能であったとしても、当然のことながら事業性が低ければ実装できない。技術中心に事業性を検討する場合は、技術の応用からサービスアイデアを考え、その後、事業性を検証する。もし、UXを戦略的に企画・構想段階から検討することができれば、利用者像及び彼らに提供する価値、そのためのチャンネルについて検討することから始めることができる。それによって、開発後に事業を開始するまでのスピードを早めることを期待できる。前述のiPod開発の場合、新しいビジネスモデルを構想することから着手していたことが伺える[11]。

このように、新たなUXを提供するサービスを創造してゆくためには、技術開発とUXデザインに加えて、ビジネスモデルの構想が必要となる。これら3つを並行して進める開発方法論には、Socio-technicalアプローチと言われるものがある。今後、我が国において、新技術を実社会に定着させてゆくには、UX、技術、ビジネスを連携させてサービスを開発する方法と、そのための人材を育成する必要がある。とくに、人材においては、UXを担う人材と、サービス全体の構想及び運用管理ができるプロデューサの役割を担う人材が少ない。UXを担う人材は、NPO法人HCD-Net^{※2}の資格認定者を参照にすれば、日本国内に数百名程度しかいない。

3.3 安定したUXの運用

1990年代のIBMの改革で使われた「顧客中心のコ

【脚注】

- ※2 Human Centered Design Organization
人間中心設計推進機構

ンピュータ・ソリューションの提供」という考え方は、BtoB 事業領域ではソリューション・ビジネスとして進展しつつある。一方、BtoC 事業領域でも、インターネットの普及によって、製品単体ではなく、付加価値のあるサービスとして提供できる環境は整備されつつある。UX を中心にした開発を促進するならば、このサービス化は促進される。その結果、利用者には、長期的に利用を保証することも必要となる。すなわち運用における品質が重要になってくる。この運用における品質を保証する場合、少なくとも2つの観点が従来の製品保証とは異なってくると考えられる。

一つは、システムが安定して稼働し、システムトラブルがあった場合、最短時間で通常の動作に戻すことである。製品が壊れないことを保証するのではなく、安定した動作を保証することである。

一方は、利用者がやりたいことを実行し、当初意図した目的を達成することを保証することである。同様に、様々な使い方に対して、様々なリスクを排除することである。これらは、利用者がやりたいことを効果的に遂行できることを保証することである。そのためには、サービス提供側が、利用者の利用状況をよりよく把握する必要がある。

2つは同じようなことに見えるが、実際は意味合いが異なる。前者は、利用者側と契約した仕様通り開発して、その通り動作すれば良い。しかし、後者の場合は、想定される使い方、危険な使い方等を推定し、利用者の利用環境に合わせて、利用者の暗黙的な意図も想定した上で動作するように開発する。実際は、企画段階から、利用者の行動を、完全に予測することは不可能であるため、運用時の情報を可能な限り入手する仕組みを作り、この情報から改善してゆくしかない。UX を指向したサービスを安定して運用するためには、そのための品質マネジメントが求められるようになる。

4 安定した UX サービス社会に向けて

本報では、UX の考え方を基に、現代の UX 的な事象と、それを構想するための方法論に触れてきた。数年前は、システムのユーザビリティの向上について議論されてき

たが、ICT の急激な発展に伴い、様々なサービスを構想できるようになった今、ユーザビリティを包含する形で、新たに UX をどのように構築するか、という課題と向き合う必要性が出てきた。

かつて、人間中心設計の基盤を作った、Brian Shackel は、人間中心設計は先端技術を人間が利用できるように型をつけるものだ、と言っている。まさしく、UX を構想することは、そのような使命を持つものである。そして、実際に、型をつけるには、技術と UX、そして、ビジネスモデルの設計を連動させながら展開する必要がある。これが可能な組織作り、人材育成は急務である。

同時に、UX を指向するサービスは、運用が重要なものであり、そのための品質マネジメントが欠かせないものとなると考えられる。この品質は、利用者側から見た品質である必要がある。

以上のような条件が整備されてゆく中で、新たな UX を提供するサービスが安定しゆく社会となるように期待したい。

5 謝辞

本論を作成するにあたり、本学商学部商学科マーケティング論担当教員である、鈴木和宏准教授からご教示いただいた。この場を通じて、謝意を記したい。

【参考文献】

- [1] ISO 9241-210:2010 Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems, 2010
- [2] 黒須正明：人間中心設計の基礎 (HCD ライブラリー (第1巻)) 近代科学社, 2013
- [3] Schmitt, B.H. (著), 嶋村和恵 (訳)：経験価値マーケティング—消費者が「何か」を感じるプラスαの魅力, ダイヤモンド社, 2000
- [4] Pine, B.J. (著), 岡本慶一他 (訳)：経験経済,, ダイヤモンド社, 2005
- [5] Garrett, J.J. (著), ソシオメディア (訳)：ウェブ戦略としての「ユーザーエクスペリエンス」—5つの段階で考えるユーザー中心デザイン, 毎日コミュニケーションズ, 2005
- [6] <https://www.gov.uk/>
- [7] IT 総合戦略本部： <http://www.kantei.go.jp/jp/singi/it2/decision.html>
- [8] IPA/SEC：組込みソフトウェア向け開発プロセスガイド, 翔泳社, 2007
- [9] 山崎和彦 他：エクスペリエンス・ビジョン：ユーザーを見つめてうれしい体験を企画するビジョン提案型デザイン手法, 丸善出版, 2012
- [10] Stickdorn, M. 他 (著), 長谷川敦士 (監訳) 他：THIS IS SERVICE DESIGN THINKING. Basics - Tools - Cases — 領域横断的アプローチによるビジネスモデルの設計, ビー・エヌ・エヌ新社, 2013
- [11] Levy, S. (著), 上浦倫人 (訳)：iPodは何を変えたのか?, ソフトバンククリエイティブ, 2007

ソフトウェア品質プロフェッショナルに求められる専門性と倫理性



早稲田大学 グローバルソフトウェアエンジニアリング研究所 所長

鷺崎 弘宜

ソフトウェア品質のプロフェッショナルは、知識、スキル、コンピテンシから成る高い専門性と、高い倫理性の両方を備える必要がある。本稿では、自身や他者が品質のプロであることを客観的に示すために必要な代表的な知識体系やスキル・コンピテンシの枠組み、及び、それらを用いる上での留意点を説明する。また職業として真にプロフェッションと認められることに向けて倫理の重要性を説明する。

1 はじめに

ソフトウェア品質のプロフェッショナルとは、顧客や社会から要求されるソフトウェア品質を、与えられた制約下で効率的に組み入れ、及び、保証する専門家である。開発・運用する人々の能力はプロセス品質に影響し、プロセス品質はプロダクト品質に影響する。品質のプロフェッショナルというためには人的能力として、妥当かつ十分な知識やスキル (Skill) 及びコンピテンシ (Competency) を備えている必要がある。

情報技術における知識、スキル、コンピテンシの定義は様々にあるが、おおむね以下が共通する。

- 知識とは、特定のタスクや業務を遂行する上で把握している必要のある事柄を認知・把握した結果である。
- スキルは、特定のタスクや業務を遂行する上で必要な専門的あるいは一般的能力である。
- コンピテンシは、知識やスキルを用いて、特定のタスクや業務を遂行する上で必要な総合的な能力である。

更に社会に対し実質的な価値を提供し「プロフェッショナル」と認められるためには、高い専門性を形成する人的能力のみならず、高い倫理性を持ち望ましい行動規範に従う姿勢が重要である。

本稿では最初に、品質のプロフェッショナルが備えるべ

き知識を整理した幾つかの知識体系を説明する。続いて、品質のプロフェッショナル (やそれを目指す者) が参照可能な人的能力の枠組みを説明する。更に、ソフトウェア品質に携わる職業がプロフェッションであると認知されることに向けて必要な倫理や行動規範、更には、それらに基づくコミュニティにおけるプロフェッショナルの認定について解説する。

2 品質の知識体系

コミュニティによって妥当と認められた知識群について特定の専門領域を定義づけるように整理及び構造化した全体を知識体系 (Body Of Knowledge; BOK) と呼ぶ [鷺崎 2014]。

ソフトウェア品質に携わる者は、品質を扱う知識体系を参照することで、日々の学習や業務により得られる知識を構造化して整理し、周囲の知識の広がりや関係を把握し、深さを追求することができる。結果として、品質のプロフェッショナルに求められる妥当かつ十分な知識を定着させることにつながる。

以降において品質を直接に扱う代表的な知識体系として、ソフトウェア・エンジニアリング全般の SWEBOK、ソフトウェア品質に特化した SQuBOK、更には品質の側面に特化した知識体系を取り上げる。特徴を表 1 に示す。

表1 知識体系のスコープと品質の扱い

知識体系	スコープ	ソフトウェア品質の扱い
SWEBOK	ソフトウェア・エンジニアリング	【知識領域】ソフトウェア品質 / 【トピック分類】基礎概念、マネジメントプロセス、実践上の考慮事項、品質ツール 【知識領域】要求※、設計※、構築※ ※品質の作り込みに言及
SQuBOK	ソフトウェア品質活動	【カテゴリ】基礎概念 【カテゴリ】ソフトウェア品質マネジメント / 【サブカテゴリ】組織レベル、プロジェクトレベル（共通）、プロジェクトレベル（個別） 【カテゴリ】ソフトウェア品質技術 / 【サブカテゴリ】工程に共通、工程に個別、専門的品質特性

2.1 SWEBOK

Guide to the Software Engineering Body of Knowledge (SWEBOK Guide) は、大学卒業後4年以上の実務経験を持つソフトウェア・エンジニアリングの実務家が知るべき知識を体系化している [SWEBOK2014]。IEEE CS と ACM によりまとめられ、現在は IEEE CS が所管している。

最新の Version 3.0 は次の 15 の知識領域から構成される：要求、設計、構築、テスト、保守、構成管理、マネジメント、プロセス管理、モデル・手法、品質、プロフェッショナルプラクティス、経済学、計算基礎、数学基礎、エンジニアリング基礎。

要求、設計、構築の各知識領域において、品質要求を含む要求全般の定義と設計、実装という形で品質の作り込みについて言及している。

更に品質特有の扱いについては、ライフサイクルの全体にわたってかわるため、要求や設計といった開発工程から独立した知識領域の一つとしてソフトウェア品質を位置付けている。品質の知識領域においては最初に「ソフトウェア品質はソフトウェア・エンジニアリングのいたるところにかかわる問題」と明確に謳っている。

品質の知識領域の構成を図1に示し、内容を以下にまとめる。

- 基礎概念：ISO/IEC25010 に代表されるプロダクトの品質モデルに加えて、エンジニアリングプロセスの品質の基準や標準として ISO9001 や CMM についても幾らか言及している。また、品質へのコミットメントを共有することがソフトウェア技術者の文化として期待されることを明確に謳い、合わせて職業倫理 (Code of ethics) について言及している。更に、安全・安心な社会基盤としてソフトウェアが重要性を増す中で、ソフトウェアセーフティが追加されている。
- マネジメントプロセス：品質保証プロセスと検証・妥当性確認プロセス (Verification & Validation; V&V) を中心に、それらの共通点や相違、計画や実施の流れを整理している。
- 実践上の考慮事項：静的技法としてのレビューやソフトウェア測定・メトリクス、欠陥分類の技法などを取

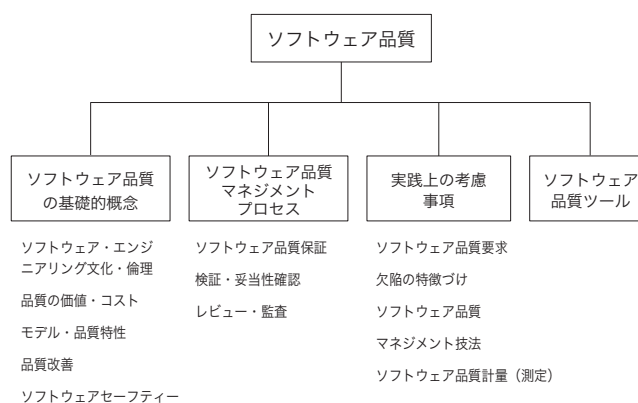


図1 SWEBOK v3.0 のソフトウェア品質知識領域

り上げている。動的な品質保証技法としてのテスト (動的テスト) は主にテストの知識領域で扱っている。

- 品質ツール：ソフトウェアの静的解析ツールについて幾らか説明している。動的解析ツールは構築、テスト、保守の知識領域で扱われている。

2.2 SQuBOK

ソフトウェア品質知識体系ガイド (Guide to the Software Quality Body of Knowledge; SQuBOK Guide) [SQuBOK2014] は、ソフトウェアの品質保証に携わるエンジニアやソフトウェア技術者・管理者を対象として、ソフトウェア品質の基本概念、マネジメント、品質技術の知識を体系化している。日本科学技術連盟ソフトウェア品質管理 (SQiP) 研究会と品質管理学会によりまとめられ、現在は SQiP 研究会が所管している。

内容をカテゴリごとに以下にまとめる。

- 基礎概念：品質の定義は時代背景や視点により変化するため、多数の品質の定義や規格を整理している。その中では、仕事や工程、人の質も含めて全体として品質と捉えることや、品質の向上により結果として納期やコストの要求を満足するという品質第一といった日本的な品質管理の考え方も含まれている。

- 品質マネジメント：構成を図2に示す。組織レベルの品質マネジメントプロセス、プロジェクト中で工程によらず扱うマネジメントプロセス（例えば構成管理）、及び要求分析やレビュー・テストといった工程に応じて扱う活動のマネジメントプロセスをそれぞれ分けて整理している。
- 品質技術：構成を図3に示す。メトリクスや品質計画に始まり、要求分析、更にはレビューという具合に主に品質の計画や確認の技術を工程に沿ってまとめている。

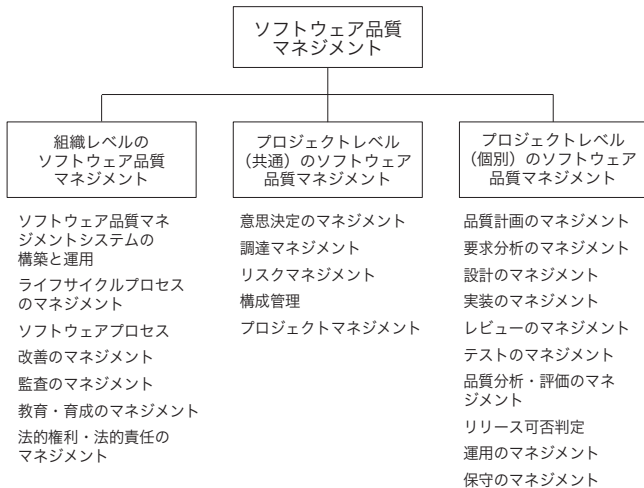


図2 SQUBOKのソフトウェア品質マネジメントの構成

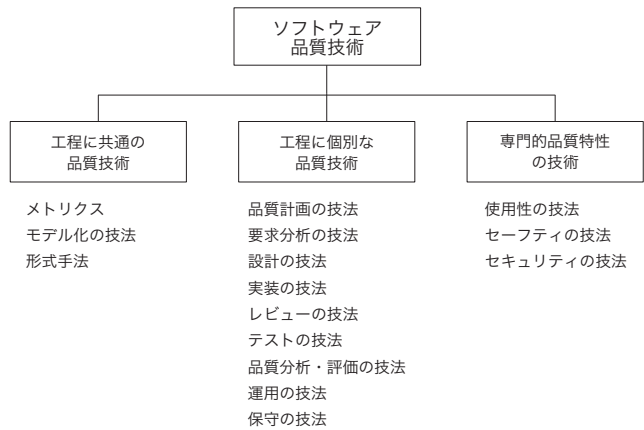


図3 SQUBOKのソフトウェア品質技術の構成

2.3 その他の品質の知識体系

ソフトウェアの品質評価や保証に特化した他の知識体系としては、品質測定における9つの論点（目標、コスト・リスク、コンテキストなど）をまとめたBody of Knowledge for Software Quality Measurement [Schneidewind2002]、テスト自動化に必要なスキルをまとめたTest Automation Body of Knowledge (TABOK) [TABOK2011] などがある。またテストについては、テスト技術者資格認定ISTQB^{※1}

のシラバス・用語集や、ISO/IEC/IEEE 29119-1 Software testing - Part 1: Concepts and definitions も知識体系を与えているといえる。

3 品質のスキル・コンピテンシの枠組み

前章で述べた知識に加えて、スキル及びコンピテンシを整理した枠組みを参照することで、品質のプロフェッショナルとして必要な人的能力を総合的に確認することができる。とくに品質は開発・運用のあらゆる段階にかかわるため、特定技術のみ詳しくればよいということではなく、プロであることを客観評価するために全体を整理した枠組みの参照が不可欠である。

以降において、エンジニアリング全般の国際的参考文献としてIEAプロファイル [IEA2013]、情報技術全般の標準の一種としてSFIA やCCSF/ITSS、iコンピテンシ・ディクショナリ、更にはソフトウェア・エンジニアリングにおける枠組みとして国際標準ISO/IEC 24773 [ISO24773] 及びIEEE CSのSWECOMを取り上げる。それらの特徴を表2にまとめる。

3.1 IEAプロファイル

国際エンジニアリング連合 (International Engineering Alliance; IEA) では、エンジニアリングの教育における卒業生に求められる属性、及び、プロフェッショナルエンジニアに求められるプロフェッショナルコンピテンシを定義しプロファイルとしてまとめている [IEA2013]。

同プロファイルにおいては、プロフェッショナルコンピテンシを13の観点（例えば知識の理解と適用、問題分析、倫理など）について、プロフェッショナルエンジニア、エンジニアリングテクノロジスト、エンジニアリングテクニシャンの3職位に沿って定義している。

同プロファイルはエンジニアリング全般を対象とするため具体的な知識やスキルの項目は含まず、専門領域における具体的な知識やスキルを別途参照して用いる。

3.2 SFIA

Skills Framework for the Information Age (SFIA) は、情報技術全般におけるスキルの枠組みを、7段階のレベルに沿ってまとめている [SFIA2011]。具体的には、次の6つのカテゴリごとに2～4のサブカテゴリ及び5～20程度のスキル項目が定義されている：戦略とアーキテクチャ、ビジネス変化、ソリューション開発と実装、サービス管理、調達とマネジメントサポート、クライアントインターフェース。これらのカテゴリ（及びサブカテゴリ）は、情報システムの開発と運用におけるプロセスにおおむね相当する。各スキル項目については、自律性、影響力、複雑性、ビジネススキルの4つの観点で必要なスキルが定義されている。

【脚注】

※1 <http://www.istqb.org/>

表2 スキル・コンピテンシの枠組みのスコープや品質の扱い

枠組み	スコープ	ベースとする知識体系	スキルのレベル	ソフトウェア品質の扱い
IEA プロファイル	エンジニアリング全般	指定なし	3段階 (3職位)	言及なし
SFIA	情報技術全般	指定なし	7段階	【カテゴリ】調達とマネジメントサポート / 【サブカテゴリ】品質と適合性 / 【スキル】品質マネジメント、品質保証、品質標準、適合性レビュー、セーフティアセスメント、技術監査
ITSS	情報技術全般	CCSF	7段階	【ほぼ全職種】 / 【スキル】品質計画、品質保証、管理 【職種】ソフトウェア開発 / 【スキル】品質検査
ISO/IEC 24773	ソフトウェア・エンジニアリング	SWEBOK	指定なし	言及なし
SWECOM	ソフトウェア・エンジニアリング	SWEBOK	5段階	【領域】品質 / 【スキル】ソフトウェア品質マネジメント、レビュー、監査、統計的管理 【領域】セキュリティ 【領域】セーフティ 【領域】測定 (及び他の領域で品質の作り込みの扱い)

品質についてはカテゴリ「調達とマネジメントサポート」中にサブカテゴリ「品質と適合性」の中で関係するスキル項目を挙げている(表2)。

SFIAは情報技術の専門能力がプロフェッショナル(専門)スキル、行動スキル、及び知識から構成されるものと定義し、それを経験や技術者認定・資格認証が裏付けとしている。

SFIAは、情報技術全般のスキルの枠組みとして事実上の世界標準である。世界各国の異なる情報技術者認定・資格認証間の相互参照を目的としたInternational Professional Practice Partnership(IP3^{※2})において共通の参照モデルとして採用され、IEEE CSにおいて策定中のIT Body of Knowledge(ITBoK^{※3})においてもSFIAの利用が明言されている。ヨーロッパにおける各種のスキル標準の共通枠組みe-Competence Framework^{※4}においても、SFIAは対応付けて用いる一例として説明されている。

更に日本国内で広く普及しているITスキル標準(ITSS)は、もともとSFIAを参考として策定されており、スキル項目やレベルの概念など相当の部分で共通している。ただしSFIAがビジネスプロセスごとにスキルを定義していることに対して、ITSSは職種ごとに定義している点異なる[IPA2011]。

3.3 CCSF/ITSS及びiコンピテンシ・ディクショナリ

IPAではスキルを「知識を活用して成果を生み出す能力」と定義し、IT人材に必要なITスキルと知識を共通キャリアスキルフレームワーク(Common Career Skill Framework; CCSF[CCSF2012])にまとめている。CCSFは、IPAが策定あるいは実施するスキル標準や情報処理技術者試験から共通に参照可能な標準及び知識体系として位置付けられている。

ITスキル標準(ITSS[ITSS2011])は、情報技術のエンジニアに求められるスキルを、職種ごとに7段階のレベルに

沿ってまとめている。個々のスキル項目の配下に知識項目があり、知識を適用する能力としてスキルを捉えることができる。またコンピテンシは、ヒューマンスキルやコンセプチュアルスキルから成るものとして別途定義している。

ITSSにおいてソフトウェア品質に関するスキルは全職種、及び、ソフトウェア開発の職種において定められている(表2)。

更にIPAは、ITSS等のスキル標準やCCSF、SWEBOK、SQuBOKに代表される各種の知識体系におけるスキル・知識項目と、典型的なタスクをそれぞれ整理し、更にその間の関係を含めてiコンピテンシ・ディクショナリ[iCD2014]にまとめている。

3.4 ISO/IEC 24773

ISO/IEC 24773:2008は、ソフトウェア・エンジニアリングの専門家の認定や資格認証の参照モデルであり、異なる認定・資格認証間の比較枠組みとして機能する[ISO24773]。

24773はスキルを、特定のタスクを遂行するために知識を適用する能力と定義している。また、ISO/CASCO 17024-1:2012を参照する形でコンピテンシを、知識やスキルを適用するものとして示された能力と定めている。

24773は枠組みであるため特定の知識やスキル項目を定義していないが、24773に照らして扱う認定や資格認証において特定の知識体系に基づいていること、及び、体系中の知識項目をSWEBOKに対応付けることを求めている。

【脚注】

※2 <http://ipthree.org/>

※3 <http://www.computer.org/web/education/itbok>

※4 <http://www.ecompetences.eu/>

3.5 SWECOM

A Software Engineering Competency Model (SWECOM) は、ソフトウェアエンジニアのコンピテンシとして、認知スキル、行動属性・スキル、及び、テクニカルスキルを定めている。

テクニカルスキルは次の13の領域から構成され、各領域においてスキル項目及び各スキル項目における活動を、5段階のレベルに沿って定めている：ソフトウェア要求、設計、構築、テスト、維持、プロセス・ライフサイクル、ソフトウェアシステムズエンジニアリング、品質、セキュリティ、セーフティ、構成管理、測定、ヒューマン・コンピュータインタラクション。

背景として求める知識及びスキル領域の構成はおおむねSWEBOOKに基づいているが、専門性の高い品質特性であるセキュリティとセーフティ、及び、横断的なスキルとして測定を独立して扱っている点が異なる。

ソフトウェア品質に関しては上述の3つの領域に加えて、SWEBOOKと同様に品質の作り込みについて他の領域で扱い、更に品質保証関係について品質の領域で扱っている(表2)。

4 知識体系やスキル・コンピテンシ枠組みの使いこなし

ここまで解説した知識体系やスキル・コンピテンシの枠組みを個人や組織におけるプロフェッショナル育成に活用するにあたり留意すべき事柄を以下に説明する。

4.1. 一般的な留意事項

ソフトウェア品質に限らず以下に留意すべきである。

- 体系化の観点の差異：知識体系やスキルの枠組みはそれぞれ異なる観点でまとめられており、目的に応じて選択すべきである。例えばスキルは、職種別にまとめられている場合と、プロセス別にまとめられている場合がある。
- スキルやコンピテンシの定義の差異：スキルとコンピテンシ及び知識を含む三者の関係は、体系や枠組みによって異なって定義されている。例えば「コンピテンシ」というとき、知識やスキルを適用する能力を指す場合と、テクニカル・ITスキル以外のヒューマンスキルを指す場合がある。とくに広範な領域の扱いを目的として、異なる複数の体系や枠組みを参照して知識やスキルの項目単位で対応付けを試みる場合、それらの定義の差異に留意すべきである。

4.2. ソフトウェア品質特有の留意事項

更に以下に示す品質特有の性質に留意すべきである。

- ライフサイクルに対する横断性：ソフトウェアの品質は、要求定義から設計を経て実装に至る品質の組み入

れ活動と、レビューやテストを経て運用に至る品質の確認・保証・管理活動のあらゆる時点で関係する。従って知識体系やスキルの枠組みを用いるにあたり、品質にフォーカスした領域のみではなく、他の領域についても品質との関係に着目して参照すべきである。逆に言うと、レビューやテストがソフトウェア品質の確保にあたり重要な位置を占めていることは間違いないが、レビューやテストの知識やスキルさえあれば品質を確保できると誤解してはならない。このようなライフサイクルに対する横断性は結果として、ソフトウェア品質は、特定の工程を担う専門職種に限らずあらゆる職種で関係することを意味する。

- 品質特性の多面性：ソフトウェアの開発と運用においてたいいてい、異なる複数のソフトウェア品質特性の達成が求められる。体系や枠組みにおける知識項目(及びスキル項目)が主として扱う品質特性を確認すべきである。例えば表3は、SQuBOK中の知識項目のうちで経験のある幾つかを、主に関係する品質特性について筆者が整理した結果である。このような知識やスキルと品質特性のマップを用意することで、要求される品質特性に応じて、個人や組織において保有する(あるいはこれから獲得する)知識・スキルが十分であるかどうかを判断できる。
- 専門的な品質特性：上述のようなマップは、特定の品質特性に着目して必要な知識やスキルを確認することに役立つ。とくに近年は、ソフトウェアシステムのインフラ化と多様化を背景として、セキュリティ、セーフティ、ユーザビリティの確保が重要となりつつある。これらの品質特性は実現にあたりしばしば、他の汎用的な品質知識・スキルとは異なった特有のものを求めるため、品質特性の側から必要な知識やスキルを深掘りできることは重要である。SQuBOKやSWECOMはその便宜を最初から図るため、これらの専門性の高い品質特性を項目や領域として独立に扱う。
- 品質特性のトレードオフ：品質特性の多面性は同時に、しばしば品質の実現においてトレードオフの関係をもたらす。体系や枠組みにおける各知識・スキル項目が、複数の品質特性を矛盾なく扱えるかどうか、品質特性間のトレードオフを扱えるかどうかについて確認するとよい。例えば表3中の品質駆動設計・評価は、多くの品質特性を扱い可能であり、評価においてトレードオフの分析を可能とする。
- プラットフォーム特有の品質：クラウドやIoTなどの比較的最近のプラットフォームやドメインにおいては、主に注視すべき品質特性若しくはその実現における知識やスキルが、従来のものとは異なる可能性がある。例えばクラウドにおいて一般に高い運用性やセキュリティを期待できる一方で、提供業者の撤退や情報漏洩といった頻度は低くとも致命的となりかねない問題を持つ可能性がある。知識体系やスキルの枠組みは大

抵、個別のドメインやプラットフォームから独立して汎用の形で定義されており、個々のドメインやプラットフォームに応じた知識やスキル項目の具体化と整理を別途検討すべきである。

表3 知識（技術）と品質特性のマッピングの例

技術	主に扱う品質	機能適合性	信頼性	使用性	性能効率性	保守性	移植性	セキュリティ
離散系モデリング・UML		X				X	X	
形式検証		X	X		X			X
品質シナリオ		X	X	X	X	X	X	X
ゴール指向要求分析		X	X	X	X	X	X	X
品質駆動設計・評価		X	X	X	X	X	X	X
アーキテクチャ、デザイン、セキュリティパターン		X				X	X	X
継続的統合CIほか		X	X			X		
クラウド		X	X		X	X	X	

5 倫理性とプロフェッショナリズム

倫理綱領や行動規範を自ら定め、社会にもたらす価値に責任を持ち、自ら律していく堅固なコミュニティをもって初めて、ソフトウェア品質に携わるといことが社会的に重要な専門職業（プロフェッション）と認められる。

国力・産業競争力を左右してしまうソフトウェア産業に携わる人々は、プロフェッションと呼ばれてしかるべき職業に就いていることを自覚すべきである [飯塚 2012]。しかしながら、ソフトウェア品質に携わる職業が、法的に高い倫理性が求められる医師や弁護士と並んでプロフェッションと認められているとは現状では言い難い。

そこで地位向上に向けて、国内外で知識体系やスキルの枠組みの整備のみならず、倫理綱領を定めてあわせて評価しプロフェッショナルとして認定する動きが活発である。

例えば SWEBOK の最新版 v3.0 では、プロフェッショナルプラクティスの知識領域を追加して、プロフェッショナルがコミュニティの倫理綱領や行動規範を遵守することの重要性を説明している。IEEE CS が実施していたエンジニアの認定制度である Certified Software Development Professional (CSDP) においては、SWEBOK に基づき知識・スキルを確認すると同時に、IEEE CS と ACM が定めた倫理綱領及び行動規範^{※5}に従うことを被認定者に求めていた^{※6}。

国内については例えば IPA の情報処理技術者試験は、CCSF に基づき情報処理技術者に求められる知識・スキル項目を確認するが、倫理性は求めないためプロフェッションとは言い難い。この場合、2014 年に開始された情報処理学会の認定情報技術者制度^{※7}を併用することで、情報処理学会の倫理綱領^{※8}に従うこととなり認定情報技術者がプロ

フェッションを形成すると認知されうることとなる。

IEEE CS/ACM の倫理綱領・行動規範と情報処理学会のものとは異なるが、「公益への供与」「依頼者との契約の尊重」を含む点では共通している。とくにソフトウェアの品質が社会の品質を左右する時代において、個別の利益のみならず広く公益への影響に責任を持つ姿勢が重要である。

6 おわりに

品質のプロフェッショナルが備えるべき知識の体系やスキル・コンピテンシの枠組みが様々な定義され、それぞれに基づいた技術者認定や教育が進められているのが現状である。そのような状況においては、自身の目的に沿って、普及している体系や枠組みをまずは参照し、広がりを見るにあたっては、個々の違いに留意しつつ i コンピテンシ・ディクショナリや 24773 等を活用し関係を追うことを推奨したい。

さらに地位向上や社会への責任を考えると、品質のプロフェッショナルの立場が法的に定められることが望ましいが、そうではない現状ではコミュニティに属し倫理綱領・行動規範を遵守することが重要である。

【脚注】

- ※ 5 <http://www.acm.org/about/se-code>
- ※ 6 IEEE CS は CSDP の運営を終了し、新たな認定制度を計画している。
- ※ 7 http://www.ipsj.or.jp/annai/committee/education/it_shikaku.html
- ※ 8 <https://www.ipsj.or.jp/ipsjcode.html>

【参考文献】

- [飯塚 2012] 飯塚悦功：“プロフェッションとしての自覚”，SQiP 研究会ソフトウェア品質のホンネ，2012，http://www.juse-sqip.jp/wp3/honne/backnumber_045/
- [鷺崎 2014] 鷺崎弘直：“ソフトウェア品質の知識体系”，情報処理，Vol.55，No.1，pp.65-71，2014
- [SWEBOK2014] IEEE CS：“Guide to the Software Engineering Body of Knowledge, SWEBOK Version 3.0,” <http://www.swebok.org>
- [SQuBOK2014] SQuBOK 策定部会：“ソフトウェア品質知識体系ガイド 第 2 版—SQuBOK Guide V2”，オーム社，2014
- [Schneidewind2002] Norman F. Schneidewind：“Body of Knowledge for Software Quality Measurement,” Computer, Vol.35, No.2, pp.77-83, 2002
- [TABOK2011] Automated Testing Institute：“Test Automation Body of Knowledge (TABOK) Guidebook,” 2011, <http://www.automatedtestinginstitute.com/>
- [IEA2013] IEA：“Graduate Attributes and Professional Competencies,” Version 3, 2013, <http://www.ieagrements.org/>
- [SFIA2011] The SFIA Foundation：“Skills Framework for the Information Age, Version 5 (SFIA 5 framework reference),” <http://www.sfia-online.org/>
- [IPA2011] IPA：“欧州 IT スキル標準の概要”，<https://www.ipa.go.jp/files/000010370.pdf>
- [CCSF2012] IPA：“共通キャリア・スキルフレームワーク（第一版・追補版）”，<http://www.ipa.go.jp/jinzai/itss/csfv1.html>
- [ITSS2011] IPA：“IT スキル標準 V3”，2011, <http://www.ipa.go.jp/jinzai/itss/>
- [iCD2014] IPA：“i コンピテンシ・ディクショナリ”，2014, https://www.ipa.go.jp/jinzai/hrd/i_competency_dictionary/
- [ISO24773] ISO/IEC 24773:2008 Certification of software engineering professionals - Comparison framework

システム及びソフトウェアの品質測定量とその測定方法



株式会社シンフォーム
谷津 行穂

1 はじめに

システム及びソフトウェアの品質の向上については、社会の変化やICTの活用場面の多様化に応じて世界中でその重要性を増してきている。システムを作る人、提供する人、使う人、利用の影響を受ける人、更にはそれらの総体としてもたらされる価値を享受する人、などシステム及びソフトウェアに直接・間接を問わずかかわる方々の多様化も急激に増加している。IoT (Internet of Things) という言葉に代表されるように世界中の物がコンピュータ及び通信によって相互連携することにより、新しい価値を提供するようになる。

これらの環境の変化に対応するためにも、利害関係者がお互いに誤解を生じないように同じ言葉で「品質」の

重要性・仕様の定義・測定・評価・向上について会話できることが重要である。そこで大きな役割を占めるのが、品質の測定量や測定方法といった表現方法・手段である。本稿では、システム及びソフトウェアを主に対象とした品質モデル (ISO/IEC 25010) とそれを構成する各種品質特性に関して、利用時や開発時の視点での品質測定量及び測定方法 (ISO/IEC 25020/21/22/23) について言及する。またシステムを構成するもう一つの重要な要素である、データの品質モデル (ISO/IEC 25012) 及びそれを構成する各種品質特性に関して、品質測定量及び測定方法 (ISO/IEC 25024) についても言及する。

2 システム及びソフトウェアの品質モデルとその構造

システム及びソフトウェア製品の品質は、その品質ライフサイクルのどのフェーズにあるかによって要求者・開発者・使用者などの視点が異なるため、フェーズ間の品質要求事項の変換が必要となる。利用時の品質要求、外部品質要求、内部品質要求などのように多段階変換を経て作成へと進む。また、ソフトウェアの静的特性、システムの行動、利用者その他利害関係者への影響、など測定対象の違いによっても品質要求事項の視点が異なるので、要求事項の変換が重要となる。これを表したものが品質ライフサイクルと呼ばれるものである (図1)。

これらの品質にかかわる要求事項の仕様化や評価といった枠組みを提供する特性の集合などを表現したものが「品質モデル」であり、ISO/IEC 25000 では、「品質

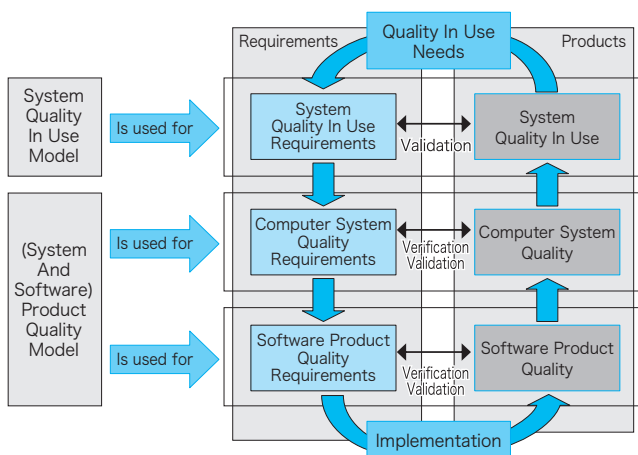


図1 システム及びソフトウェアの品質ライフサイクルモデル (ISO/IEC 25000 に基づく)

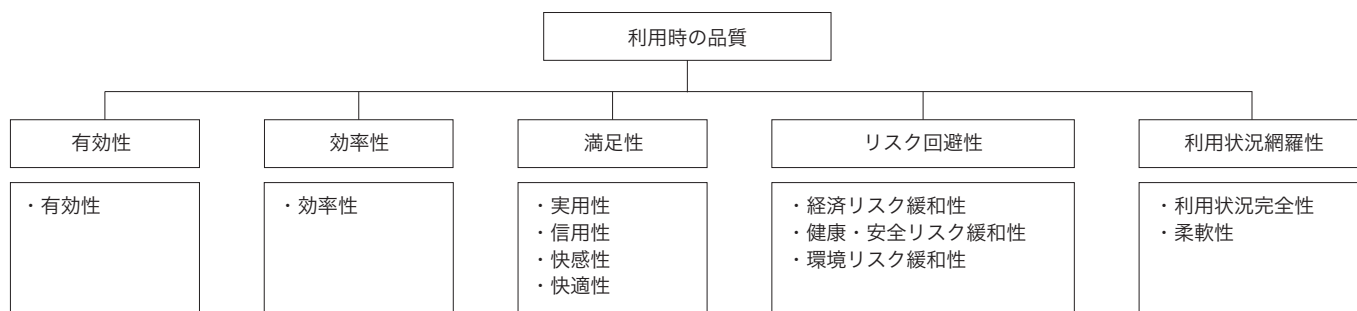


図2 利用時の品質モデル (JIS X 25010 に基づく)

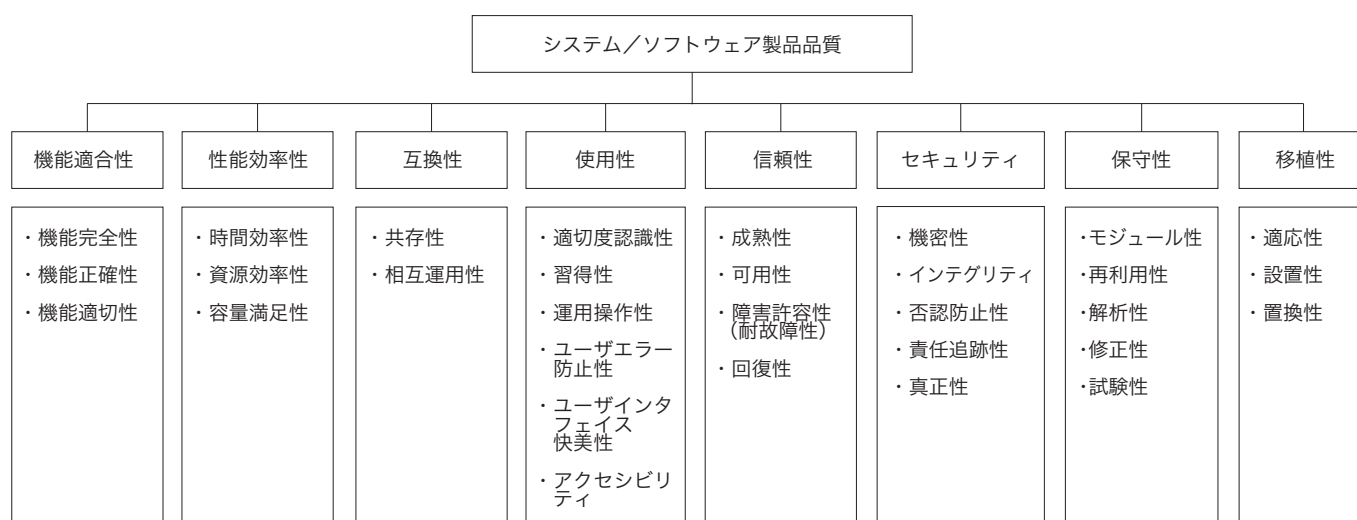


図3 システム・ソフトウェア製品の品質モデル (JIS X 25010 に基づく)

特性	データ品質	
	固有	システム依存
正確性 (Accuracy)	○	
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)	○	○
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)		○
移植性 (Portability)		○
回復性 (Recoverability)		○

図4 データの品質モデル (JIS X 25012 に基づく)

要求事項の仕様化及び品質評価に対する枠組みを提供する特性の定義された集合及び特性間関係の集合」と定義されている。具体的には ISO/IEC25010 System and software quality models に、利用時の品質 (Quality in

use model—図2) 及び製品品質 (Product quality model—図3) として提示されている。

なお、近年その重要性が認識されてきたデータ品質についても同様な考え方で、要求事項の仕様化や評価といった枠組みを構成する品質特性が提供されている。システム及びソフトウェアが生成、蓄積、提供、更新及び抹消する情報データすべてに対して、その属性を測定する (図4)。

3 SQaRE における品質測定法の段階的設計

システムやソフトウェアの品質を測定する際に、よく用いられるメジャーは開発後半となるテスト段階やシステム統合段階などに問題 (バグ) 件数を測定し、品質の評価をするなどである。これでは品質向上を促進することは難しく、もっと早期の開発初期や利用者の要望・要求を論じるときに、要求分析をして要求と測定法を明らかにしてきちんと仕様化することが必要である。このことが品質向上を促進する。

顧客や開発者と利用者がお互いに合意した品質要求事項及び達成を確認できる測定法について、準備しておくことが重要である。それに基づいて、品質要求を詳細化していき、各段階での達成度合を評価できるようにしておく。その際にISO/IEC 25010の品質モデルが役立ち、品質を分類した品質特性・品質副特性ごとに測定法を設計していき、順次適用する。

測定法の段階的設計では、次のような流れで実施する(図5)。

- 最初に要求事項を明確にし、測定・評価する品質特性(副特性)を選択する
- 最終製品(システム及びソフトウェア)や中間成果物などの測定対象物を定める
- 品質特性(副特性)に影響を及ぼす特徴(属性)を選定し、定量化する測定方法を定める
- 測定の範囲、時期や段階、粒度などを定める
- 実際の測定の詳細を詰める(レビューの種類、どのようなテストか)
- 以上の品質測定法の要素を確定し、組み合わせて計算し測定結果を出す

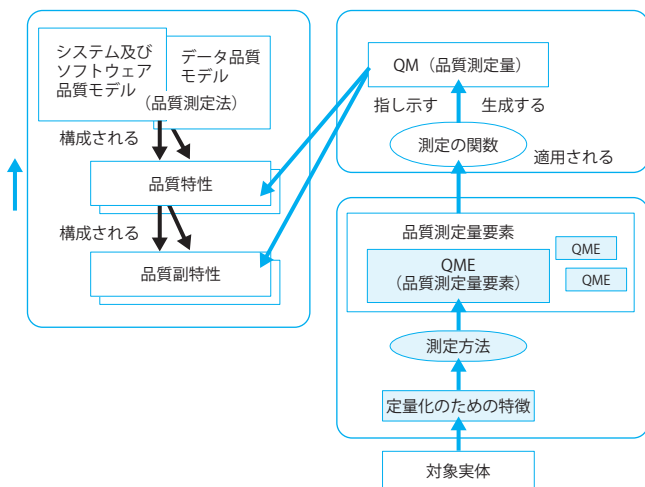


図5 システム及びソフトウェアの品質測定法の構成 (JIS X 25021 に基づく)

このような構成で段階的に品質測定法を定義したものを「品質測定量 (Quality Measure)」と呼び、品質特性・品質副特性そのものは直接定量化することはできないが、このように計算した品質測定量を、品質特性・品質副特性を定量的に表現する指標 (Indicator) として用いることができる。本稿では品質測定量及びその測定方法を合わせて品質測定法と記述する。(JIS X0133-1 4.20 参照)

4 SQuaRE における品質測定量とその測定法の実際

ISO/IEC 25000 SQuaRE シリーズは、2500n: 品質管理部門、2501n: 品質モデル部門、2502n: 品質測定部門、2503n: 品質要求部門、及び 2504n: 品質評価部門の5つの部門、及び 2505n~2509n までの拡張部門から構成される。その中でも品質測定量と測定法に密接に関係するのが、2501n: 品質モデル部門と 2502n: 品質測定部門である。2501n 品質モデル部門では、25010 品質モデルで利用時の品質モデルとシステム及びソフトウェア製品品質モデルを示している。また、25012 データ品質モデルでは、システムがもたらす情報の品質について重要な構成要素であるデータの品質モデルについて 25010 を補完している。

2502n 品質測定部門では、25020 で品質測定量の参照モデルとその使用の手引きを提供し、25021 で基礎となる品質測定量要素の開発ガイドと要素の例を提供し、

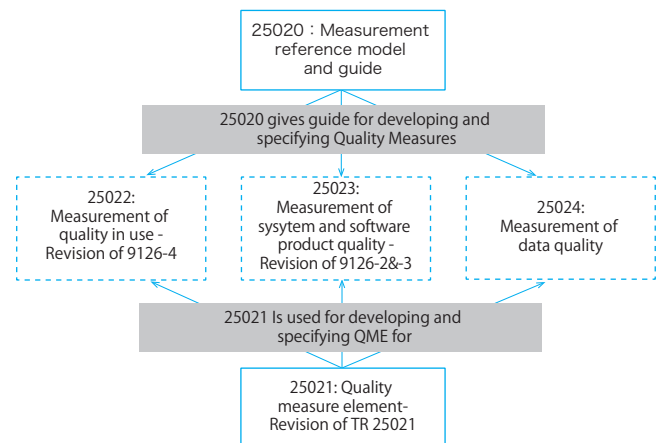


図6 SQuaRE 品質測定部門の構造 (JIS X 25021 に基づく)

25022 ではシステム・ソフトウェア製品を実際に利用した際に利用者が感じる品質を利用時の品質 (Quality in use) としてその測定量及び測定方法を提供し、25023 ではシステム・ソフトウェア製品品質を外部品質 (実行・テスト時) と内部品質 (設計・実装時) の観点から測定量及び測定方法を提供し、25024 ではデータ品質の測定量及び測定方法を提供している (図6)。

このことから、ISO/IEC 25010 の品質モデルを用いると次のように測定法を設計できることがわかる、

- 利用時の品質特性の測定法：利用者がシステム及びソフトウェアを利用した時に及ぼす影響について効率や満足度及び危険性などに分類した品質特性また

は品質副特性について測定する

- 外部品質の特性の測定法：本格利用前のテスト段階などで実現できるシステムやソフトウェアの挙動や使用性などの動的な品質特性または品質副特性について測定する
- 内部品質の特性の測定法：システムやソフトウェアの内部構造などの静的な品質特性または品質副特性について測定する

それぞれの品質特性についての測定量及び測定方法について、以下に説明する。また、測定した結果について評価プロセスでの扱い方や、評価の考え方については評価プロセスの稿にゆずる。

利用時の品質の特性の測定

ISO/IEC 25010 の利用時の品質モデルで、**有効性、効率性、満足性、リスク回避性、利用状況網羅性**の5つの品質特性から構成される。ISO/IEC 25022 ではその品質特性・品質副特性ごとに品質測定量が定義されている。利用時の品質は、実際に利用した利用者の視点からの品質測定であり、開発者の側から測定した使用性(Usability)とは異なる視点である。有効性、効率性、満足性については利用者の利用場面によって影響を受けるので、利用者層の違いや、タスク・環境の違いなどをはっきりと識別することが求められる。例としては、満足性での「実用性」があげられる。システムが備えたある機能についてその実用性を利用の場面で測定する場合、既述したようにそのユーザ種別や使用環境、利用タスクなどが各種ある場面での評価としては、チェックリストや質問による満足度合いの回答を得る、という方法がとられることが多い。その回答のマイナスからプラスの点数の集計などで評価する場面がこれにあたる。

次に個々の品質特性について簡単に解説する。

- 有効性：利用者が目標を達成する時の正確さや完全さの度合い
- 効率性：利用者が目標を達成するための正確さや完全さに関連して、使用した資源の度合い。その場合の資源とは費やした労力、材料や資金などがそれにあたる
- 満足性：ある利用状況でシステム及びソフトウェアが使用されるとき、利用者ニーズが満足される度合いであり、副特性は実用性、信用性、快感性、快適性である
- リスク回避性：システム・ソフトウェアを使用する

時の、潜在的なリスクを緩和する度合いであり、副特性は経済リスク緩和性、健康・安全リスク緩和性、環境リスク緩和性である

(例) 経済性リスク緩和性の品質測定量の例として、
$$\text{REc-3-G 業務成果 (利益率向上度合)} = \frac{\text{実利益率}}{\text{目標利益率}}$$

というように品質測定番号をつけて利用・参照しやすいように提示されている

- 利用状況網羅性：想定された利用状況及び想定を超えた利用状況の両方においてシステムまたはソフトウェアが利用できる度合いであり、副特性は利用状況完全性、柔軟性である

外部品質の特性の測定

製品品質モデルの品質特性としては、**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**という8つの品質特性から構成される。ISO/IEC 25023 ではその品質特性・品質副特性ごとに品質測定量が定義されている。外部品質特性は、システム及びソフトウェア製品の品質特性として、外部から動的に測定するもので構成される。要求分析から展開した外部仕様やテスト仕様を準備しておき、実際のテストでの動作から仕様との乖離や合致度を測定する、ということが代表的な使用場面である。例としては、性能効率性での「時間効率性」などでは代表的な品質測定量として応答時間やスループットが知られている。

各品質特性はそれぞれ品質副特性に展開されている。次に個々の品質特性について簡単に解説する。

- 機能適合性：ある状況で使用するとき、明示・暗黙にかかわらずニーズを満足させる機能を、製品またはシステムが提供する度合いであり、副特性は機能完全性、機能正確性、機能適切性である。機能仕様ではなくニーズであることが重要である
- 性能効率性：ある条件で使用する資源の量に関する性能の度合いであり、副特性は時間効率性、資源効率性、容量満足性である。理解しやすい品質測定量である
- 互換性：同じ環境を共有する間、製品、システムなどが他の製品、システムなどとの情報交換できる度合い、またはその機能を実行できる度合いであり、副特性は共存性、相互運用性である
- 使用性：有効性、効率性及び満足性をもって目標を達成するために、利用者が製品またはシステムを利

用することができる度合いであり、副特性は適切度、認識性、習得性、運用操作性、ユーザエラー防止性、ユーザインターフェース快美性、アクセシビリティである

—信頼性：明示された条件で、システム、製品などが明示された機能を実行できる度合いで、副特性は成熟性、可用性、障害許容性（耐故障性）、回復性である。

耐故障性や可用性などはなじみのある品質副特性であり、MTBF や平均回復時間などもよく目にする品質測定量である

—セキュリティ：人間、製品若しくはシステムが、認められた権限に応じたデータアクセスができ、情報及びデータを保護する度合いであり、副特性は機密性、インテグリティ、否認防止性、責任追跡性、真正性である。近年非常にその重要性を増している。SQuaRE シリーズでも新たに大幅な改訂が加えられた品質特性である

—保守性：製品やシステムを修正することができる有効性や効率性の度合いであり、副特性はモジュール性、再利用性、解析性、修正性、試験性である

—移植性：ある運用環境または利用環境からその他の環境に、システム、製品などを移すことができる有効性や効率性の度合いであり、副特性は適応性、設置性、置換性である

内部品質の特性の測定

製品品質モデルの品質特性としては、**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**という8つの品質特性から構成されることは外部品質特性と同じである。ISO/IEC 25023 ではその品質特性・品質副特性ごとに品質測定量が定義されている。

システム及びソフトウェア製品の品質特性として、内部から展開した特性を「開発者視点で測定する」と言える。品質特性としては外部の品質特性及び品質副特性と同じ分類であるが、品質測定量や測定法に内部品質の測定用に解釈を加えた部分がある。それはソフトウェアなどの内部構造などを対象とすることによるものである。システムやソフトウェアのアーキテクチャや論理設計、プログラム設計、コンポーネント設計、ソースコードなどが備えるべき特徴を製品品質特性に対応させて定量化して表す、ということにあらわれている。品質測定の活動も、デザイン・レビュー、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じて測定する。この品質特性・品質副特性の記述は外部品質の特性を参

照されたい。

データ品質の特性の測定

データ品質の特性という点では、データ自身が保有している品質特性と、影響を受け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る。それらの特性に加えて、データの特徴としてそのライフサイクルに合わせて、データの作成時、蓄積・保存時、加工時、移送時、利用時、抹消時、などそれぞれの局面でデータ品質特性を表すことができる。ISO/IEC 25024 では、データ品質特性をデータ固有品質特性群とシステム依存品質特性群とに分類でき、それぞれの品質特性ごとに品質測定量と測定方法で定義されている。

—データ固有品質特性（12 品質特性）

正確性、完全性、一貫性、信ぴょう性、最新性、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

—システム依存品質特性（10 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、可用性、移植性、回復性

以上述べてきたように、図1で示した品質ライフサイクルに沿った考え方として、利用時の品質ニーズから始まり実行時の品質、更には設計・実装時の品質にわたる各段階で品質測定を実施することにより、システム及びソフトウェア製品の品質を適宜確認していくことができる。その各段階で品質要求を定義・確定するときに品質測定量を含む品質測定法を用いる。このような考え方を適用することにより、冒頭述べたように利害関係者同士がお互いに共通の言葉や考え方で品質を定義し測定していくことができるようになり、効果的な品質向上に資することができる。

5 おわりに

今後、ますますシステム及び社会が複雑化していく中において、システム及びソフトウェア製品の品質を的確・適切に定義し測定していくことは、様々なリスクを認識し解決していくためにも重要な第一歩と位置付けられる。本稿で述べた、SQuaRE を中心とした品質ライフサイクル、品質モデル、品質測定量とその測定方法を活用して、プロセス設計と品質評価とを連動させることが、一層重要になる。品質測定量を含む品質測定法を更に活かすものにしていくために、具体的に成果をあげているような事例に更に目を向けて普及させていきたい。

小規模開発組織と品質課題

— VSE 国際規格の課題に即して —



東海大学 情報教育センター 講師
情報規格調査会 SC7/WG24 主査

伏見 諭

1 ソフトウェア品質と小規模開発組織の役割

我々の生活と産業を支えているソフトウェア品質の重要性はいうまでもないが、それがどのように確保されるかという点では、考えておくべきことがある。意識されることが少ないように見えるが、ソフトウェアの開発を担っている企業や開発組織の多くが実は比較的の小規模なものであることが重要である。この「小規模である」ということは、小規模な企業の役割が大きいということ、及び開発チームが小規模であることという2つの視点からなっている。

まず、小規模企業の比重がどのようなかを経済産業省の特定サービス産業実態調査の結果で点検してみると、表1のような状況となっている。すなわち、30人以下の企業が8割強、100人以下の企業が9.5割といった割合を占め、非常に大きな企業数割合を占めていると言える。

表1 ソフトウェア業規模別の企業数及び売上高
(経済産業省 平成25年度特定サービス業実態調査結果から集計)

企業規模	企業数	売上高 (億円)
30人未満	24,154 (82%)	22,631 (16%)
30～99人	4,083 (14%)	31,438 (23%)
100～499人	1,037 (3.6%)	35,613 (26%)
500人以上	297 (0.5%)	49,600 (36%)
合計	29,433 (100%)	139,282 (100%)

また、小規模な開発の比重がどのようなを示すデータは、IPAのデータ白書などで報告されている(2012-2013年度のデータ白書では5,000人時以下のプロジェクトが4割近くを占める)。このデータ白書のデータの母集団の範囲が大規模な企業である点や業界での多層発注構造の存在を考えれば、実際の開発現場では統計的なデータより更に小規模な開発が多いとも見られよう。

さて、企業の売り上げ規模や、総工数の観点から見ると、「小規模」の比重は、上記のような企業数分布、プロジェクト件数分布での比較より比重が下がるわけだが、問題は、実際の開発における「管理の主体」は、それぞれの企業であり、またよく見られるようにそれぞれの部署や開発チームがプロジェクト/プロセス管理の内容を決めているという点である。結論的に言えば、ソフトウェアの開発品質は、実はこれらの小規模企業や小規模開発チームの実態が決めていくということになる。

小規模企業や小規模開発チームでの開発品質をどのようにコントロールできるか。そのアプローチには、ソフトウェアのサプライチェーンでの品質統制を高度化する(例えば、品質要件の厳密化や受け入れ試験の厳密化)という方向性と、開発現場それ自体の能力を上げるという方向性が考えられよう。ここでは、後者の方向でのアプローチを紹介する。

2 VSE 国際規格

ソフトウェアの品質に関する標準的な観点は、ハンドブック類やいわゆるBOK (Body of Knowledge) なども存在するが、国際規格というものが議論しやすい。製品

の品質は、SQuaRE 規格で、開発プロセスの品質は SLCP やそのアセスメント方法である SPICE 規格 (ISO/IEC 15504 とその発展形である 33K シリーズ規格) に代表される国際規格が定められている。

しかし、前節で述べたような小規模企業、小規模開発組織にとっては、それらの大きな規格は実際的には馴染みが薄く、また取り入れようとしても対応リソースや適切なテラリング実施能力などの面で無理があるとの見方がある。そうした観点から、より馴染みやすく、リソース制約や特定の状況の元でも採用が容易なソフトウェア/システムエンジニアリングの規格を用意すべきであるとの提案がなされた。これは、2004 年の ISO/IEC JTC1/SC7 会合での提案である。

さて、グローバル経済下でのソフトウェア開発でも、品質の確保は重要な問題であり、米国や EU から新興国などへのソフトウェア発注では、国際規格や米国の CMMI を用いた開発企業への要求がなされてきた経緯がある。しかし、それらが大きなコスト要因となることから、新興国のソフトウェア開発企業ではリーズナブルな負担での国際水準の確保が望ましいという声も背景としてあった。

SC7 会合で新たに提案された規格は略称で VSE 規格シリーズと言われる。VSE とは、Very Small Entities の略であり、従来の中小企業 (SME) と言われる企業層よりも更に小規模な企業層/開発チームという位置づけの用語である。数字的には 25 名以下程度と特徴づけられている。

VSE 規格シリーズは、実は色々な種類の VSE を対象とする方針で開発されており、複数のプロセス (および成果物) 規格を用意する。それらの基本となるものから制定を始め、その最初の規格を「基本プロファイル」と言っている。基本プロファイルは、大きなプロセス規格である SLCP (ISO/IEC 12207) のプロセス群から必須と言うべきプロセスとその要素を少数選択し、またそれらの実施に伴う作業成果物群を標準的に必要なものとしてまとめ直したものである (図 1 参照)。

本稿のテーマであるソフトウェア品質達成ニーズの視点で見ると、VSE 規格は、製品開発プロセスとその作業成果物 (文書など) を軽く標準化することにより、ごく基本的な開発作業上の指針を提供しているものと評価できる。

図 2 に、現在開発されている「プロファイル」群と今後の開発方針を示したが、小規模 IT 関連組織の関与するビジネス領域を幅広く取り扱おうとしていることがうかがえよう。

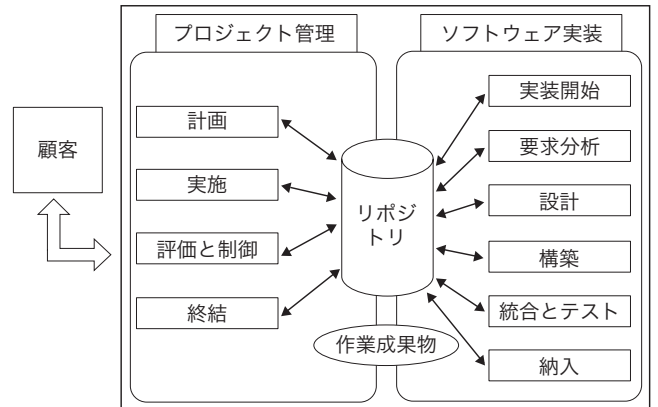


図 1 VSE 基本プロファイルの概要

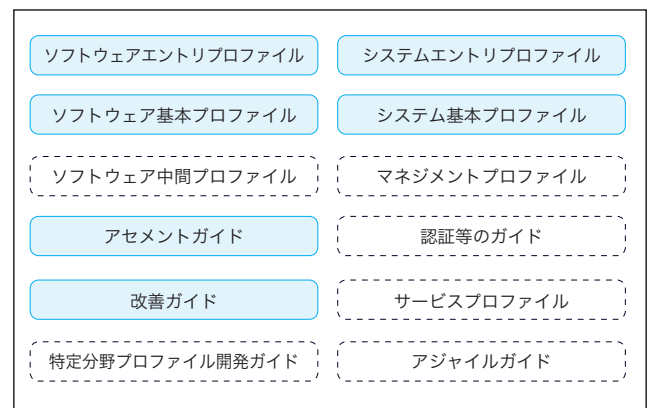


図 2 VSE プロファイル群と提起されている拡張方向 (実線は既存、破線は開発中のもの)

3 小規模開発の品質問題

小規模企業、小規模開発チームの製品品質と開発作業品質をどのように確保するかという問題は多くの困難な点をはらんだ問題である。「品質要求を完璧なものとする」という議論もありうるだろうが、現場実態としては、例えばアジャイル開発の発想に見られるように、現場の知恵による創意工夫と発想力に支えられて品質が確保されているのも重要な事実であり、その現場力というものを向上させることは重要なアプローチである。

VSE プロファイルは、直接には、開発などのプロセスの品質確保を目指すものであるが、検証と妥当性確認 (いわゆる V&V) に重点的な目を向けるなど、製品品質の課題にも貢献したいというスタンスがある。また、実質的にも、要求事項の着実な実装という面では、プロセス品質と製品品質は密接な関係がある。

小規模企業、小規模開発チーム自身が、開発プロセスの品質向上に取り組むモチベーションと確信を持ち、品質要求としてあらかじめ明示されたものも、はっきり

とは明示されていないものも含めて、品質目標や利用者の品質期待の本質的な実現を目指す必要がある。おそらく、VSE 規格自身もそのような観点から見直しと強化を行っていく必要がある。

VSE 組織に発注を行う立場のシステム利用者・システム運用者にも VSE 規格の導入のメリットがある。VSE 組織が一定の品質で開発作業を行い、またその製品にかかわる品質トレーサビリティも確保できることがメリットにあげられる。そのような規格導入は VSE 組織自身の自律性で行うのが望ましいが、発注条件への組み入れといった形で用いられることもあり得る。とくにグローバル発注においては、この規格が国際規格である点を活用すれば相手方に納得させることが容易になると思われる。

4 サプライチェーンでの品質問題

近年の品質にかかわる特徴ある視点として、セキュリティや安全性の問題がある。また人間工学的な問題もある。これらの品質要求の多くは、製品企画の段階で基本的な品質方針が設定されるべきであり、その意味では、VSE 企業が関与しない領域で多くが決められるという場面も多々ある。

そうした場合には、VSE 企業は製品全体の品質コンセプトや、個別の品質要求群を発注者から受け取る立場であると考えられる。その際の問題点は、そうして伝達される品質要求群が正しく理解され、適切な程度に実装され、また不足する要求（や発注側からは暗黙の仮定となっている要求）が VSE によって適切に補足されるかどうかという問題である。

そのように見ると、VSE が品質目標を理解する能力、それを開発作業で活かし得る能力、開発の見積りなどで品質対応がどのような作業負荷となるかを明確化し得る能力などが、求められることになると思われる。

現在の VSE 規格には、上記のような高度な要求視点は無いが、VSE 規格の応用方向として、一部で検討が始まっているところである。

既存の VSE プロファイル群は、クリティカルでない開発分野を対象としている。しかし、日本から提案した「特定分野プロファイル開発ガイド」は、VSE 規格を、より多様な領域に拡張適用できるようにするものである。例えば医療分野・自動車分野などのクリティカルソフトの領域、セキュリティが問題となる領域も含め、そのサプライチェーンの中で、VSE 規格を活用して必要十分と言える現実性の高いプロセスを確保すること、最終的な製品の品質確保に貢献することを目指している。

5 IPA の貢献

IPA では、旧ソフトウェア・エンジニアリングセンターのプロジェクトとして、プロセス改善 WG の活動から、SPINA³CH 自律改善手法（スピナッチキューブ）を 2011 年に公表し、2013 年に改訂版を公表した。これはソフトウェア開発現場のプロセス改善を促進するためのモデルであり、その性格から、VSE の現場でも適用しやすいものと考えられる。そのため、この手法の応用を VSE 国際規格の一部とするのが有効ではないかと考え、そのような応用手法が国際提案された。結果、2014 年秋にその規格（VSE 規格シリーズの一部として自律改善手法を推進する Technical Report）が最終的に承認され、2014 年末現在で国際規格発行待ちの状態となっている。

この新規格は、VSE の本体部分から見ると、プロセス自体の標準化というより、プロセス改善というやや異なる視角のアプローチを提供するものである。小規模企業、小規模開発チームの品質能力向上という意味では、国際貢献として大きなものがあると期待される。

VSE 規格は当初から、いわゆるプロセスアセスメント手法を用いたプロセス改善の切り口を持っていたが、この IPA からの貢献は、具体的な開発上のトラブルや問題意識から出発してプロセスを自律的に確立・向上しようとするものであり、様々な現場状況に合わせてプロセス強化策を探ろうとする現在のトレンドにも合致するものと言えよう。

6 今後の展望

VSE 規格は、日本工業規格（JIS）としては、JIS X 0165-2 として、シリーズの一部の規格が制定されている。日本国内では、国際的なシーンでの新興国／開発途上国と同様のニーズ（国内ソフトウェア産業の認証的な制度による底上げ）は見られないが、ソフトウェア品質の向上という VSE 規格に込められたニーズは共通すると考えられる。日本では、サプライチェーンの項でふれたような、VSE 規格の応用的な試みと、SPINA³CH 自律改善手法との組み合わせといった試みがなされており、それらが全体として成果を上げることが期待される。

また、小規模開発チームという視点ではアジャイル開発というキーワードもあるが、VSE 規格とアジャイル開発との融合を目指す動きもあり、そうした様々な試みが広い意味での品質向上の成果にも結びついていくことが期待される。

Embedded Technology 2014 (ET2014)

出展報告

SEC 企画グループ 主任

荒川 明夫

IPA/SEC は、2014 年 11 月 19 日から 21 日にかけてパシフィコ横浜で開催された Embedded Technology 2014 (ET2014) に出展した。同会場内アネックスホールでは、ET2014 の主催者である一般社団法人組込みシステム技術協会 (JASA) と共催し、IPA セミナーを 2 日間開催した。

1. 展示会概要

Embedded Technology (ET) とは、一般社団法人組込みシステム技術協会 (JASA) が主催する組込みシステム技術に特化した国内最大規模の専門展であり、組込みシステム開発にかかわる技術者や開発者向けに最新技術などの情報を発信している。

2. 出展概要

IPA/SEC は、事業成果の普及・啓発を目的として、本展示会に出展しており、今回で 10 回目となる。本年は、6 コーナーを設け、IPA/SEC で取り組んでいる信頼性の高いソフトウェアを開発するための取り組みのみならず、セキュリティや人材育成などの取り組みについても幅広く紹介した。

また、パネル展示やデモの実施に加えて、ブースプレ

ゼンを行い、IPA 職員や SEC の事業成果普及に貢献いただいている連携委員、地域連携団体・組織からそれぞれの取り組みなどについて発表した。

3. IPA 展示ブース

展示ブースでは、以下のコーナーを設置し、該当する事業成果のパネル展示や関連資料の配布を行った。

- ・ 開発技術
- ・ システム障害事例・対策
- ・ 品質 / プロジェクトの見える化
- ・ 人材育成
- ・ セキュリティ
- ・ 地域連携

『開発技術コーナー』では、「ソフトウェア高信頼化への取り組み」や「高信頼化を確保するための手法」に関するパネル展示のほか、「アジャイル開発」や「共通フレーム 2013」など、ソフトウェア開発において参考にしていただきたい情報を紹介した。

『システム障害事例・対策コーナー』では、JASA と共同開発を行っている事業であり、本展示会初日に新製品発表会場で記者発表を行った「モデルベースアプローチに基づく障害原因診断手法」について、デモ機を用いて紹介した。

『品質 / プロジェクトの見える化コーナー』では、ET-West2014 で記者発表を行った組込み系開発プロジェクトに特化した「組込みソフトウェア開発データ白書」の



紹介のほか、ソフトウェア品質の説明力を強化するための様々な取り組みをまとめ、来場者に説明した。

『人材育成コーナー』では、情報処理技術者試験、i コンピテンシ・ディクショナリ など、IT 人材を育成するための事業を紹介した。

『セキュリティコーナー』では、組込みシステム開発に深く関連する自動車や医療機器の情報セキュリティや、「ファジング」「Android アプリの脆弱性の学習・点検ツール」など、セキュリティに関する事業を紹介した。

『地域連携コーナー』では、IPA/SEC と連携している全国各地の業界団体の中から 14 団体の事業を紹介した。

また、各コーナーでのパネル展示のほかに、関連資料の配布、「SEC10 年の歩み」パネルの展示、ブースプレゼン(39 セッション / 3 日間)、SEC 先端技術入門ゼミ(15 セッション / 3 日間) を実施した。

4. SEC 先端技術入門ゼミ

本出展からの新企画として、ソフトウェア開発などの先端技術や各業界での取り組みを解説する初心者向けのチュートリアルである「SEC 先端技術入門ゼミ」を実施した。

セミナーやカンファレンスと異なり、受講者が直接、講師に質問をすることができるこの企画は、受講者だけではなく講師にも好評であった。受講者からは、今後も取り上げて欲しいテーマについて「ソフトウェア開発にしばったシステムエンジニアリング入門」や「形式手法によるモデリングの事例解説」などの要望をいただいた。講師からは「学校のような講義形式であったため、受講者の反応がよくわかった」や「受講者の理解度に応じて講義を展開していった」という意見が寄せられた。

5. IPA セミナー

展示会場に隣接されたアネックスホールでは、本年も 11 月 20 日、21 日の 2 日間、IPA セミナーを 8 部構成で開催し、延べ 700 名の参加があった。

今回の IPA セミナーは、1 日目に「セーフティ&セキュリティ設計の見える化」や、「製品・サービスに必要なソフトウェア品質」など、IoT 時代に対応するソフトウェ

ア開発について講演したほか、超上流工程を支える新たな方法論である「コンセプトエンジニアリング」、「先進的な設計・検証事例にみる開発技術のトレンド」について、紹介した。

2 日目は、「環境変化への適応から考えるサイバーセキュリティ対策」や「障害解析から再発防止・未然防止につなげる活動事例と教訓を共有する仕組み」、「ソフトウェア開発データ白書」の取り組みを紹介し、記者発表を行った「モデルベースアプローチに基づく障害原因診断手法」についても講演した。



6. ET2014 を振り返って

今回の出展は、過去 10 回の出展の中で、最も多くの方に IPA 展示ブース及び IPA セミナーに足をお運びいただいた。アンケートや展示会場で来場者の方からいただいた多くのご意見を次回出展の参考とし、今後の IPA 事業活動に反映していきたい。今後も例年どおりの出展方法ではなく、今回実施した「SEC 先端技術入門ゼミ」のように、新たな企画にも挑戦し、来場者・参加者の理解度・満足度向上に努めていく。

ET2014 IPA/SEC ウェブサイト

<http://www.ipa.go.jp/sec/events/20141119.html>

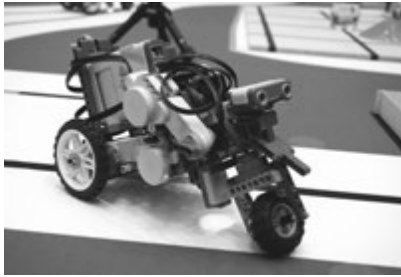
- IPA セミナー・IPA ブースプレゼン・SEC 先端技術入門ゼミの講演資料がダウンロードできます
- IPA セミナーの動画を公開しています

ETロボコン 2014 観戦記

SEC 研究員

杉崎 眞弘

毎年恒例となった一般社団法人組込みシステム技術協会（JASA）主催の全国規模のロボットコンテスト「ETロボコン」。今年も若い技術者の方々による熱戦が繰り上げられました。その熱戦の様態を紙面をお借りして紹介します。



ETロボコンとは？

主催である一般社団法人組込みシステム技術協会（JASA）の案内によると、組込みシステム分野における技術教育、人材育成をテーマとしたロボットコンテストで、コンテストの目的は、「5年後10年後に世界をリードするエンジニアの育成を目指し、若手及び初級エンジニア向けに、分析・設計モデリング開発、製品サービスの企画開発にチャレンジする機会を提供する。」とあります。

実際このETロボコンの特色には、

- ・走行体（ハードウェア）は、指定された同一の規格・部品を組み立てたもの（教育用レゴ®）を使用し、各チームは独自で設計・製作した搭載ソフトウェアの性能のみで競い合えるようになっています。
- ・競技内容は、所定のコースでの走行競技と事前の設計モデル審査とで総合的に競うデベロッパー部門と、自由な発想で走行体を使った製品化、サービス化の企画力を競うアーキテクト部門に大別されています。更に、デベロッパー部門は、ワンメイクの2輪走行体でソフトウェアの設計力を競う入門向けプライマリークラスと、新型3輪走行体で高度な仕様変更の対応性が要求されるアドバンスクラスに分かれており、それぞれのクラスでの審査・評価となります。
- ・もともとUMLロボコンとして2002年に開始された

ことから、事前のモデル審査に加えて、競技後に優秀な走行体の設計モデルを講師陣の講評のもと、参加者全員で吟味する有意義なモデル・ワークショップが各大会ごとに用意されています。

という点があります。

このことからETロボコンはソフトウェア技術者の修練の場として、「動くものを作るおもしろさ」に加えて「競い合う」「優れたものを直に見れる」と技術者心をくすぐり続けて、今年で13年目になる歴史ある全国規模のコンテストです。

ETロボコンへのIPAのかかわり

IPAは後援団体としてETロボコンを支援しています。

今年は、IPAから理事・SEC所長をはじめ、職員が北は北海道から南は沖縄まで全国11の全地区大会にお伺いし、特別賞であるIPA賞のプレゼンターを務めさせていただきました。そして最終決戦の横浜でのチャンピオンシップ大会では、IPA理事長も出席して、熱戦を観戦させていただきました。そもそもIPAがETロボコンの後援をしている理由は、この大会を組込みソフトウェアの設計・実装・実行の出来栄を評価する総合的なソフトウェア開発競技会と捉え、若手ソフトウェア開発技術者の育成には絶好の修練の場であると考えているからです。

そのことより、特別賞としてIPA賞を設けており、競技の優劣より、設計における要件の明確化、モデリング

図法の的確さ、設計展開の十分さ、表現の仕方などを評価して、これを授与しています。



立石 IPA 理事
(北海道・東北)



松本 IPA/SEC 所長
(関西)

地区大会 2014 の模様

今年の各地区大会とチャンピオンシップ大会のスケジュールは次の通りで、9月13日の南関東・中四国大会を皮切りに、以降毎週各地で大会が開催され、各大会で勝ち残ったチームで競い合う横浜でのチャンピオンシップ大会まで、1カ月間でこなすハードスケジュールでした。各大会でお世話になった大会運営スタッフの方々の熱意と規律正しさには敬服いたしました。

ET ロボコン 2014 全国地区大会・チャンピオンシップ大会スケジュール

地区大会	開催日	会場
南関東	9/13 (土)	関東学院大学横浜・金沢八景キャンパス [神奈川県横浜市]
	9/14 (日)	
中四国	9/13 (土)	福山大学 宮地茂記念館 [広島県福山市]
北陸	9/14 (日)	金沢工業大学扇が丘キャンパス [石川県野々口市]
関西	9/14 (日)	京都コンピュータ学院京都駅前校 [京都府京都市]
	9/15 (月・祝)	
東北	9/20 (土)	いわて県民情報交流センター (アイーナ) [岩手県盛岡市]
九州	9/20 (土)	九州産業大学 [福岡県福岡市]
	9/21 (日)	
沖縄	9/20 (土)	沖縄産業支援センター [沖縄県那覇市]
東海	9/27 (土)	浜松市勤労会館ホール / 静岡大学情報学部 [静岡県浜松市]
	9/28 (日)	
東京 (2大会)	10/11 (土)	早稲田大学西早稲田キャンパス [東京都新宿区]
	10/12 (日)	
北海道	10/12 (日)	北海道情報大学 [北海道江別市]
北関東	10/12 (日)	小山市立生涯学習センター [栃木県小山市]
チャンピオンシップ大会	11/19 (水)	パシフィコ横浜 [神奈川県横浜市]
	11/20 (木)	

ここでは、筆者を含めて全大会をカバーした IPA 職員の感想を、写真を交えてご紹介します。

まずは、各チーム自慢の走行体の走りっぷりですが、ごつい体型に似合わずセンシティブな面があり、とくにコースを読み取る光センサは、会場の明るさ、不用意なフラッシュ撮影の影響を受けやすく、走行前のキャリブレーションには、みなさん真剣そのものの表情でした。

デベロッパー部門の競技コースのレイアウトは、全大会すべて同一仕様で作られていて公平を保っています。

コース上には、クラス別にルックアップ・ゲート、駐車場、スピン台、ジャンプ台、モーグルなど難所と呼ばれる腕の見せ所の加点部分に加わり、更にアドバンストクラスでは大会前日の 21 時まで詳細が公開されず、直前でプログラム変更が要求される仕様未確定エリアが用



プライマリークラス
2輪倒立振り子型



アドバンストクラス
3輪 NXTrike



なにはなくともキャリブレーション・・・





意されています。それらをチーム一丸となって技を繰り出してクリアすると高得点が加算され、総合得点で競うルールになっています。

同じ走行体にもかかわらずプログラミングの違いでこれほど走りと技に差がつくのかと思えるほど、余裕で走破するチームから、スタートで逆走、転倒してしまうチームなど、悲喜こもごもの情景でした。

次に、お楽しみのアーキテクト部門は、各チーム趣向を凝らしたショータイムの様相を呈していて、技術を突き詰めて走行体の可能性に挑戦したもの、家庭や社会にこんなロボットがいたら助かるかもと思えるコント付のデモンストレーション、更にキャラクターが登場する演劇風のものまであり、大変楽しませてくれました。

そんな中、筆者が最も驚いたのは、2輪走行体を玉乗りのように丸い筒の上に乗せ、自走しながら、前方に迫る障害物を音波レーダで捉え、筒の上を水平移動しながら避けて通過するというものでした。離れ業に見える動きに感動して、終了後早速そのチームにどうしたらそういう動きができるのか話を聞きに行ってきました。左右の車輪の回転数を変えながら前後の動きを繰り返して水平移動を可能にしているそうで、さすが自動車メーカのチームだけあって、車輪の軌跡について熱っぽく語ってくれました。(写真がうまく撮れなかったのが残念)

熱戦を終えて

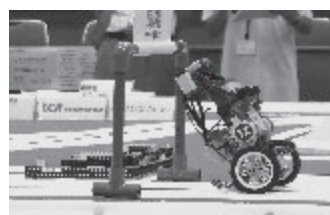
大会中、熱戦を終えた多くのチームに色々話を聞かせてもらいました。最も聞きたかったのは、ETロボコンへの参加動機です。企業チームのほとんどは新人研修としてプログラミング学習のため半年から1年かけてこのプロジェクトに業務として参画しているとのこと。うまくいかず人知れず枕を濡らすこともあったとか。高校、高

専、専門学校、大学からの学生チームは、学科の研究テーマとして技術を先輩から受け継ぎながら研さんをつんでいるとのこと。また学生チーム・社会人チームを問わず女性の活躍が目立ち、日本の将来は明るいと思いました。

多くの大会では、競技日の翌日がモデル・ワークショップになっており、これがこの大会の最も貴重で有益な時間ではないかと思えます。実際の開発現場での経験豊富な講師陣が、競技で優秀な成績を修めたチームの設計仕様書(モデル分析・設計書)の良いところ、改善点などを解説してくれる内容で、なかなかほかでは聞けない実践的な学習の時間になっており、全チームが参加して熱心に講義を受けていました。また、大会会場に貼りだされている参加チームそれぞれの設計仕様書には、いつも若い人で人だかりができており、自分たちの悩みを優秀チームにぶつける姿は頼もしいと思いました。

今回IPA賞を受賞された各大会それぞれのチームは、大会審査員による事前モデル審査の段階から設計書の仕様記述内容で高い評価を受けており、競技でも良い走りにつながっていました。まさに「練られた設計は、良い結果に結びつく」が実証されていると思います。今回筆者は初めて観戦して最も感銘を受けたのは、一昔前まで日本では先進的な技術者の間でのみ取りざたされていたUMLによる設計表記が、今や若い高校生チームにまで普及し習得されていることです。自分の設計を表現するのに世界共通の標記手法が用いられているのを見ると、隔世の感を覚えると共に、これからの若い技術者が世界に飛躍することを期待せずにはいられません。これからも修練に励まれることを熱望しています。

また、このような有意義な場が、ますます発展されることを願っています。



米国の NIST、MIT、SEI の 3 組織を訪問して

SEC ソフトウェアグループ リーダー

中尾 昌善

1. はじめに

IPA/SEC では、国際連携活動の一環として、米国の有力なソフトウェア技術拠点である NIST（米国商務省国立標準技術研究所）、MIT（マサチューセッツ工科大学）、SEI（カーネギーメロン大学ソフトウェア工学研究所）と定期的に意見交換を行っている。今年度も 12 月 8 日から 12 日にかけて 3 組織を訪問したので、その模様を報告する。

2. NIST

(1) CWE（Common Weakness Enumeration：ソフトウェアにおけるセキュリティ上の脆弱性の種類を識別するための共通の基準）

NIST 側からソフトウェアの脆弱性情報を取りまとめた CWE、CVE^{※1}、CAPEC^{※2}に関する取り組みの紹介があった。色々な組織から集めている情報のため定義があいまいで、3つのデータベース間の整合性がないなどの問題



写真1 厳重な NIST の正面入り口：ここからシャトルバスに乗って目的の建物に行くくらい構内面積が広い。



写真2 NIST メンバとの議論模様：この後、NIST 職員
のクリスマスパーティにも飛び入り参加させて
いただいた。

もあるが、ベンチマーク用の情報として活用されているようである。今後は、CWEの項目間の関係からソフトウェアの欠陥パターンを導出することや、形式記述での表現の仕方などについて検討を進めるとのことである。IPA/SECからは、コーディング作法ガイド（ESCR）へのセキュリティ項目の取り込みの活動を紹介し、その内容について確認してもらうこととした。

(2) SAMATE（Software Assurance Metrics And Tool Evaluation）

産業界や政府など様々な機関から提出してもらったソフトウェアの Assurance 情報を集めた SARD（Software Assurance Reference Dataset）というデータベースの紹介があった。これもベンチマーク用に活用されており、3,000ダウンロード/月程度のアクセスがあるとのことである。

【脚注】

※1 Common Vulnerabilities and Exposures

※2 Common Attack Pattern Enumeration and Classification

(3) 医療機器のインタオペラビリティ

IoT時代における医療機器のインタオペラビリティに関する取り組みの説明（接続テストやデータログの取得など）があった。IHE（Integrating the Healthcare Enterprise）など色々な機関との関係についても解説があり、NISTが各機関のハブとなって、この取り組みを推進していることがわかった。IPA/SECからは、ソフトウェアサプライチェーンと品質に関する取り組み、及び障害事例に基づく教訓集作成の取り組みを紹介した。教訓集の取り組みにはNISTも関心があるようで、今後意見交換を継続していく必要があると感じた。

いずれに関しても活発な質疑応答があったため、予定時間をオーバーしての終了となった。

3. MIT

MIT訪問の主要目的は、ナンシー・レブソン教授にお会いし、彼女が創始者であるSTAMP技術に関する情報交換を行うことであった。STAMPとはSystems Theoretic Accident Model and Processesの略であり、システム理論に基づく事故モデルの構築手法のことである。JAXA「こうのとり」の事前ハザード分析や中国高速鉄道の事故分析に適用されたことで知られている。

IPA/SECもSTAMP技術の有用性に着目し、その導入推進に力を入れていこうと考えており、その活動を開始している。

当方から直前に多くの質問事項を送付したにもかかわらず、教授はそれらの一つ一つへの回答資料を数10ペー



写真3 雪の舞うMIT校内移動：後ろに見える建物でミーティングを実施した。

ジにわたって用意して下さっており、その誠実さが窺えた。主な質問は、STAMPの実際の適用域に関する事項、具体的な技術内容の確認、今後の活動予定等であった。

まず、STAMP技術が既に幅広い分野で利用されていることを具体的に示していただいた。更に、大規模システムは色々な専門家が個々にいるほど複雑なので、すべてを理解して見通すのは難しいのではないかという質問にも、まずはステップ1で共通的な事項に取り組み、ステップ2で領域毎の技術に取り組んでいくとの回答であった。色々な自動化ツールを開発中であり、その1つは既に1,000ダウンロード以上に及び、またSTAMPのワークショップには25カ国からの参加があり、関心の高さと適用活動の多さを感じ取れた。

STAMPの新たな研究動向として、自動化ツールやコンセプトレベルでの安全要件・モデル作成の手法、STPA-Secと呼ぶサイバーセキュリティへの活用などのトピックを紹介いただいた。とくに、セキュリティに対しても因果関係分析のSTAMPモデルが適用できるとの見解は興味深いものであった。

お忙しい身でありながら、30分以上も超過して我々の更なる矢継ぎ早の質問に答えて下さったレブソン教授に感謝を申し上げたい。



写真4 レブソン教授とのミーティング：左奥の女性がレブソン教授

4. SEI

SEIのあるピッツバーグは氷点下であった。そんな寒い朝の8時半に始まったミーティングは、途中SEI所長であるニールセン氏との昼食を挟み、夕方4時半まで続いた。

(1) Assurance

SEIの主な活動分野は、Software Engineering, Assurance, Specific Capabilities, Cyber Securityの4つであるが、最近ではAssuranceに関する要求が高まっているとのことである。サプライチェーンにおけるSoftware Assuranceでは、プロダクト自身のセキュリティ設計だけでなく、どのような状況で使われるのかといったSystem Assuranceの考え方が重要とのことである。IPA/SECからは、ソフトウェアサプライチェーンに関する取り組みと、障害事例に基づく教訓集作成の取り組みを説明した。教訓集の米国における活用の是非について聞いたところ、適用環境が異なるとなかなか難しい面はあるが、マインド醸成には有効で、学校等で教えるのが良いかもしれないとの意見だった。

(2) Software Quality & SME

SEIでは以前CMMI（ソフトウェア開発プロセスの改善モデルと評価手法）をSME（中小企業）へ適用する検討を行ったが、SMEは品質への関心が薄く、リソースも限られることから、なかなかうまくいかなかったため、現在はとくにSME向けの活動は行っていないとのことである。INCOSE（システムエンジニアのための国際団体）では、VSME（Very Small and Micro Entities）のWGでシステムズエンジニアリングの観点での取り組みが行われているようである。

(3) Agile in Government

SEIがかかわっている国防総省の開発プロジェクトを短縮化する目的で、アジャイル開発の適用を進めており、現在では約30%近くをアジャイルで開発しているとのことである。費用が固まらないと政府系では適用が難しいのではないかと質問したところ、要件リストを作り、総量で管理するようにすれば可能であるとのことだった。

(4) TSP（Team Software Process：ソフトウェア開発工程の分析 / 改善手法の一つ）

TSPの適用は250以上のプロジェクトに広がり、ソフトウェアの欠陥を中心として分析を始めているとのことである。IPA/SECで発行しているソフトウェア開発データ白書が対象としているプロジェクトと規模的にも似て



写真5 SEI入り口：カーネギーメロン大学の一面にある。



写真6 先方のプレゼン模様：それぞれが持ち時間をオーバーして説明するほど熱の入ったものであった。

いるため、相互にデータを比較して日米の違いを明らかにすべく、秘密保持契約を結んで今後共同で検討を進めることとした。

5. おわりに

訪問した3組織とも、国際的な意見交換や共同作業の重要性を強調しており、今後も引続き連携を進めていくことで一致した。また、いずれにおいてもセーフティとセキュリティは一体であるとの認識であり、IPAもSECとISEC（セキュリティセンター）で一体となって対応していく予定である。

IPA グローバルシンポジウム 2014 開催報告

～ソフトウェア高信頼化セッションより～

SEC システムグループ 主任

八嶋 俊介

1. はじめに

2014年10月22日、IPAは「豊かな社会を創るIT、豊かな社会を拓げるIT」をテーマとして、IPAグローバルシンポジウム2014を開催した。本シンポジウムは、国内外の有識者による講演・パネルディスカッションによって構成されており、国際的な視点に立ったITの最新情報を発信することで、我が国全般のIT力向上に資することを目的に開催しているイベントである。

本稿では、ソフトウェア及びシステムの高信頼化をテーマとして実施した3つの講演について、それぞれ概要を報告する。



写真1 会場内の様子

2. 今求められるユーザの開発力とは

最初に、注文を受けてから2ミリ秒での応答を実現した次世代株式売買システム「arrowhead」の開発を背景に、システム開発において求められるユーザ企業の役割について、株式会社日本取引所グループ 専務執行役 CIO 鈴木義伯氏にご講演いただいた。

arrowhead の取り組み

arrowhead の開発において目標とし、実現したものは以下の3点である。

- ・レイテンシー（遅延時間）の低減（高速性要件：10ミリ秒以下の応答時間→2ミリ秒を実現）
コンピュータを使った多くの発注がなされるようになってきていることから、旧システム（応答時間2～3秒）から大幅な処理の高速化が目標とされた。
- ・信頼性の向上（信頼性要件：99.999%の可用性）
2005年から2006年にかけて、システム障害やシステムの性能不足による取引停止が多く発生しており、市場からの信頼を取り戻すため「止まらないシステム」の整備が急務となっていた。
- ・リニアな拡張性の実現（拡張性要件：5年後、10年後までの拡張性確保）
発注者側のコンピュータの処理性能が上がり続けている（発注量が増え続けている）背景から、それに耐え得る拡張性・柔軟性が目標とされた。

arrowhead 開発成功の鍵

前述のような高い信頼性を誇る arrowhead の開発を成功させた鍵のひとつに、「発注者責任の明確化」が挙げられる。

日本取引所グループでは、処理業務の重要度でシステムを分類しており、それぞれについて発注工程や品質レベルを定め、ITベンダとのかかわりを規定している。とくに、arrowheadのような取引所の中核業務であるマーケットを支える基幹システムは、重要度が非常に高く位置付けられており、開発は取引所自らが取り組むこととされている。具体的には、企画・要件定義・外部設計までを取引所が責任を持って進めることとしており、このルールを徹底することで、発注者責任を明確化し、取引所の責任で要件定義書を作成した。

ふたつめの鍵として、「フィードバック型V字モデル」の採用が挙げられる。これは、開発工程において前工程の品質を、次工程で確保しながら開発を進める手法であ



写真2 鈴木義伯氏

る。通常のウォーターフォールモデルであれば、V字の後半からテストフェーズに入るが、この手法を採用し、とくに単体テストに入る前のコーディングの段階で詳細設計のバグを出すことで、全体のバグの5～6割を出すことができた。先行工程で品質を高めることで手戻りを減らし、生産性を高めることに成功した。

求められるユーザの開発力

ユーザのIT部門には、顧客のニーズに応えながら世界の中で存在感を保ち、求められるビジネスモデルの変化をキャッチアップし続け、業務を継続することが求められる。業務部門と一体となり、新しいビジネスモデルに挑戦し、実業に適用する開発力が必要となる。このためには、モデル/システムを企画し、要求仕様をまとめ、要件に落とし込める能力がIT部門に備わっていないといけない。日本取引所グループでは、新入社員は何年かのうちに必ずIT部門に属することをルール化するなど、人材育成の面からも開発力の強化を進めている。

人材育成という観点では、ITベンダがユーザを育てるという発想も重要だと言える。ユーザのIT力を育成することもITベンダの役割であると考え、これが実現できれば、結果的にITベンダ自身も有意義にプロジェクトを進めることができ、お互いに良い関係を構築することができるだろう。

3. ヒューマンエラーとシステム安全 ～英国ヒューマンファクター第一人者によるR&D最前線～

2番目に、イギリスの鉄道業界におけるヒューマン

ファクターの考え方や、事故を未然に防ぐための報告の仕組みについて、英国RSSB（鉄道安全標準化機構）シニアヒューマンファクターズスペシャリストヒュー・ギブソン氏にご講演いただいた。

RSSBとは

RSSBは、総延長距離20,000km、年間15億人の乗客を運んでいるイギリスの鉄道業界において、主に安全性やパフォーマンスの観点から業界のサポートを行っている組織である。イギリスの鉄道業界は日本と異なり、線路や信号を管理するインフラ事業者（2社）と、実際に車両を運行し乗客を運ぶ旅客会社（約30社）がそれぞれ別に存在している。RSSBは、これらの加盟会社から資金を受けて運営しており、政府からは独立した活動を行っている。活動の主な内容として、以下の4点が挙げられる。

- リスクの理解……車両の脱線や、乗客のホームからの転落などのリスクを分析する。リスク対策を行うに当たり、どこに焦点を当てればよいかの指針（ガイドライン）を出している。
- 様々な会社との協業……国内外の様々な会社（日本の会社も参画）と持続可能な開発を行い、ITを利用した列車制御、信号制御も含めた、エンジニアリングの向上に努めている。
- 標準化……利用者が安全に鉄道を利用できるように、鉄道会社間でのインターフェースを確立する。（ルールの統一化）ホーム上の白線の位置や幅、車両のライトの明るさなども標準化の項目に含まれる。
- 研究開発……研究成果の技術移転や、これからの鉄道の提案などを行っている。日本の鉄道総合技術研究所とも協業しており、例えば、世界的にはとても厳しいルールである指さし確認の有効性を評価し、イギリスの鉄道業界でも取り入れるなどの検討を行っている。

ヒューマンファクターとは何か

1999年、ロンドンのパディントン駅で発生した大きな鉄道事故（ラドブローク・グローブ事故）は、車掌によるヒューマンエラーが原因とされた。これは、ヒューマンエラーを防止するため、ヒューマンファクターの考

え方を鉄道業界に取り入れていかななくてはならないという大きな転換点となった。

ヒューマンファクターは、個人、職場、組織の3つのレベルから成る。個人である車掌や運転士が働きやすいように、職場、あるいは組織から、適切なサポートを受けているかどうかを検討する。

RSSBは、ヒューマンファクターを法律に組み込んだり、国としての標準を作るような取り組みを行っている。具体的には、車掌のライセンスの提供方法の検討や、運転士から見えやすいような信号の設置場所の標準化などが挙げられる。

このような取り組みの中で重要なのは、システムの開発や運用において、バグや障害から学ぶことが多いのと同じように「事故から学ぶ」ことである。RSSBでは、安全に関する情報を収集して分類する必要があると考えており、様々な会社がお互いに情報を共有するような取り組みを行っている。

また、ITを活用した方法では、SMIS (Safety Management Information System) と呼ばれる国が運用する事故データベースから事例を抽出し、根本原因を検討し、IFCS (Incident Factor Classification System) と呼ばれるデータベースを作成する取り組みを行っている。乗客の転倒や信号の見落としなど、年間約75,000件の新規事例が格納されるSMISのデータの中から、およそ300件を抽出し、コミュニケーション、設備、環境などの10の根本原因に分類している。この結果、信号に関する事故の56%は、コミュニケーションが原因で発生していることが分かった。



写真3 ヒュー・ギブソン氏

機密保持報告システム CIRAS (サイラス)

CIRAS (Confidential Incident Reporting and Analysis System) は、前線で働く車掌や信号係のためのもので、問題を発見／報告しても会社に対応しない場合、直接CIRASに報告できる仕組みである。(銀行でいう内部告発のようなもの) 1999年のラドブローク・グローブ事故以来、関心が高まっており、鉄道分野以外にも、海洋、航空、高速道路などの各分野へ多角化が図られている。

例えば、信号が木に隠れていて見えづらい状況について、運転士が会社に報告しても改善が図られない場合、運転士はCIRASに電話をかける。(Webサイトでの登録も可能) 連絡を受けたCIRASは原因究明の調査を行い、機密を保持した上で、会社の代表者に調査結果を伝える。それを受けて会社は調査を行い、結果についてCIRASから運転士にフィードバックを行うという仕組みである。年間3,000件の電話が寄せられており、有用な情報については業界全体にフィードバックが行われる。

CIRASによる効果として、会社がモニタリングを積極的に行うようになったなどの安全面の直接的な改善だけでなく、国のルールが変わったといった、インパクトの大きな変化も起きている。CIRASの活動成果については、RSSBが発行しているニュースレター「The CIRAS」やWebサイトで誰もが事例を参照できるようになっている。

4. 社会とテクノロジーの融合をデザインする ～社会システムとテクニカルシステムのコ・デザイン～

最後に、現代社会の解決しがたい様々な問題を解決するために、俯瞰的に物事を捉えることの必要性について、慶應義塾大学大学院 システムデザイン・マネジメント研究科 (SDM) 准教授 白坂成功氏にご講演いただいた。

システムズエンジニアリングの最新動向

システム工学は古くからある学問であるが、1974年から1994年にかけて大きな変化があった。以前はEngineering Managementということで、科学的にマネジメントを行うOperations Researchや最適化を中心とした学問であったが、1994年以降は設計を中心とした「Systems Engineering Process」にシフトしてきている。

2000年以降は更に範囲が広がっており、以下のような考え方が出てきている。

- System of Systems Engineering (2008) [1]
例えば「デジカメ」と「プリンター」をつないで印刷する場合のように、それぞれ独立して存在するシステムが、つながったときに1つのシステムになるという考え方。「System of Systems=巨大なシステム」と誤解されることが多いが、システムの規模は問題ではない。
- Enterprise Systems Engineering (2010) [2]
これまでのシステムズエンジニアリングのスコープを拡張し、単にモノを作るだけでなく、モノを作る側の考え方を変えていく必要がある。より高い視点から全体像を見渡した上で、具体的にプロダクトに落とし込むという考え方。
- Engineering Systems (2011) [3][4]
例えば電気自動車を社会に投入するときは、モノだけでなく、充電設備などのインフラ、対応する政策(法律)などが必要となる。これからは必要なものを同時にシステムとしてデザインしていく必要があるという考え方。
システムの構成要素は、これまで対象としてきた技術的観点である「人工技術的要素」、これから考慮する必要がある社会的観点である「人的要素」、人間によって作られていない自然の部分であり、これまで俗人的に考慮されていた「自然要素」の3点である。これらシステムの構成要素を体系的にとらえてデザインするために、「規模と範囲」「機能」「構



写真4 白坂成功氏

造」「時間的広がり」の4点を意識し、観察角度(Viewpoint)を変えながらシステムをデザインするという考え方。

Viewpoint について

例えば家を建てるときには、屋根の色や家具を決めるだけでは不十分で、水回りや配線、費用など、様々な観点で検討しなければならない。このような項目は、専門家に聞けば検討項目は十分に網羅できる。一方、そうでない者が初めてシステムを構築する際は、Viewpointをどのようにデザインするかを考える必要がある。

例として、過去に話題となった保険金不払いの問題について、再発防止の対応システムを設計することで実験した。システム設計と同じ手法(人工技術的要素)を用いて人間の行動(人的要素)を設計し、検討した結果、出来上がった再発防止の対応システムは、専門外の人間が設計したものにもかかわらず、金融の専門家が考えたものとほぼ同じものとなった。

社会とテクノロジーの融合をデザインするために

システムズエンジニアリングは、社会的観点まで広げることで、社会システムと技術システムを対象とできる。しかし、実現するためには、様々な人を巻き込んだ集合知、デザイン思考(マインドセット)が必要となる。したがって、多様な専門家が協働するための仕組み、方法論を検討していく必要がある。

5. 終わりに

ITが我々の身の回りに幅広く浸透し、社会生活を営む上でなくてはならない存在となることで、その安全性に対する取り組みも極めて重要となる。

例えば、今回の3つの講演で共通していた人的要因、あるいは人に対する問題を、エンジニアリングの中にどのように取り込んでいくのかについては、今後の重要な課題となってくるだろう。

【参考文献】

- [1] Mo Jamshidi : 「System of Systems Engineering」, 2008年
- [2] George Rebovich Jr., Brian E. White : 「Enterprise Systems Engineering」, 2010年
- [3] Charles M. Vest, Olivier L. de Weck, Daniel Roos, Christopher L. Magee : 「Engineering Systems」, 2011年
- [4] 春山真一郎(監訳) : 「エンジニアリングシステムズ」, 2014年

欧州におけるソフトウェア高信頼化技術の適用状況

～欧州2カ国の著名教授による特別セミナー講演より～

SEC ソフトウェアグループ リーダー

中尾 昌善

1. はじめに

ソフトウェアの高信頼化技術は、我が国では先駆的な取り組みを行う大学や企業がある一方で、まだまだ浸透しているわけではない。その課題克服のために、欧州での導入推進の状況を参考とし、意見交換を行うことを目的として、2014年10月29日にSEC特別セミナーを開催した。冒頭で、経済産業省 情報処理振興課長の野口聡氏よりIT産業への期待感についてお話をいただいたのち、英国ニューキャッスル大学教授のJohn S. Fitzgerald氏とデンマークオーフス大学教授のPeter Gorm Larsen氏にご講演いただいた。更に、ご講演を踏まえ、両氏と我が国の産学官の有識者によるソフトウェアの高信頼化技術の導入推進に関するパネルディスカッションも行った。



写真1 野口聡情報処理振興課長

2. John S. Fitzgerald 教授の講演概要

同氏は、ニューキャッスル大学のソフトウェアリライアビリティセンター (CSR:Center for Software Reliability) の所長であると共に、EUのFP7 (第7次

研究枠組み計画) のシステムモデル化推進のプロジェクトを指揮した欧州における先導的な立場の方である。

ニューキャッスルは産業革命の中心地

英国北部に位置するニューキャッスルは、海に面した石炭の産地であり、蒸気機関の発明に代表される産業革命の中心地であった。そのような伝統のもとに、ニューキャッスル大学は、民間からの資金提供を受けながら、50年にわたるコンピュータサイエンス研究の歴史があり、CSRは1984年に設立された。

CSRの原則

CSRには3つの原則が存在し、それらは次の通りである。

- ① ソフトウェア単体だけでなく、システム全体として捉えること。
- ② システムの信頼性を考える際には、確かさを示す証拠が必要である。
- ③ 理論開発は、実践に結びつかなければならない。

EUから研究資金提供を受けていることから、とくに③を重要視しており、技術を受け入れる産業界の実務者の作業に貢献することに力点を置いている。

技術革新プロジェクトの紹介

- (1) 航空宇宙関連の組織と共同で進めたトラステッド・ゲートウェイの開発には、コストと品質の改善のために形式手法を適用した。これは、産業界に対して、より高度な手法を導入するためのプロモーションプロジェクトであった。
- (2) ボッシュと行った事例では、エンジンのアイドリング時の燃料効率性解析を形式手法 (Event-B) で実施し、要求事項のモデリングに要する負荷を40%

削減した。技術を産業レベルにスケールアップするに際して、形式モデルで理解を高め、課題事項をはっきりさせ、ツールで作業の改善を行うという効果が得られた。一方で、要求事項のモデリングによる抽象化を進めすぎると実装とのギャップが生じるので、そのギャップ管理が重要であること、及びツールは適用産業ごとのドメインツールが必要であるという認識も得られた。このプロジェクトが成功したのは、産学間のコミュニケーションにて互いの懸念事項を交換できたことが大きかった。エンジニアリングはチームワークだということを再認識する機会にもなった。

今後の課題

CSRのプロジェクトでは、形式手法などがアーキテクチャを構造化するモデルや確率的なモデルにも適用可能なかという今後の課題が見出せた。

IoTの時代には、Systems of Systems という概念での考察が必要で、システムの個々の構成要素が、その環境の中で、それぞれの挙動としてどうあるべきかを考えることが重要である。更に、そのシステムを支える周辺技術（ツールなど）にも配慮が必要である。

今、ニューキャッスルでは、大学だけでなく市民も参加可能なサイバーフィジカルシステムズラボラトリー（Cyber Physical Systems Laboratory）の設立が進行中であり、2017年には施設が完成し、ますます産学連携を進めていく予定である。



写真2 John S. Fitzgerald 教授

3. Peter Gorm Larsen 教授の講演概要

同氏は、オーフス大学のソフトウェアエンジニアリン

グ科の科長である。産業界で開発エンジニアとして活躍し、その後学問の世界に戻られたという経験から、形式手法を初めとした各種開発技法の産業界への導入推進に力を発揮されている。

抽象化の効能と注意点

システムをモデル化し、抽象度を上げることで、システム全体を理解しやすくすることとアイデアの正しさを確認することができる。ただし、同じ飛行機の解析モデルにおいても、風力解析モデルとコックピットにおけるパイロットのヒューマンファクターモデルは全く異なる。すなわち、モデル化するには、解析の目的に応じて、厳密化が必要な部分と本質的でないため削ぎ落とす部分を常に意識しなければならない。

米国における調査結果

形式手法に関して、2009年と2012年に100件近いプロジェクトに対して、その有効性に関する調査を行った。形式手法導入により、「時間がかかった」という回答が25%、「短縮できた」が20%、「変化なし」が約半数であった。コスト面での質問では半数以上が未回答だったが、回答があった中では4：1の割合で「コスト効率が上がった」という回答が「悪化した」という回答を上回った。また、「確実性が高まった」という回答が88%にも及んだ。導入上の障壁は何かという質問には、「教育」と「ツールの統合」がほとんどであった。

欧州におけるR & D戦略

EUにはHorizon 2020というプログラムがあり、約700億ユーロが研究投資されている。その目的は、将来の雇用創出であり、次の3つに力を注いでいる。①卓越した科学、②産業界でのリーダーシップ、③社会問題の解決。とくに、製造業において雇用が失われてきたという社会問題が深刻であるため、要素技術の開発により将来のイノベーションを強化し、中小企業の成長につなげることを目指している。そこでは、サイバーフィジカルシステムが重要であるという認識となっている。

サイバーフィジカルシステム（CPS: Cyber Physical System）の活用例

物理的なシステムに対して、ソフトウェアによるモデリングが役立つというのが、CPSのコンセプトの一つである。例えば、郵便において、紙を折って封筒に封入するシステムがあり、そこでは曲がった折り方にならない

ことをソフトウェアによるモデリングにより確認することで、事前に問題を発見した。また、セルフバランシングスクーター（セグウェイのようなもの）の開発では、仮想環境で動作をシミュレートすることに寄与した。ほかにも、川底を掘る掘削機の開発においては、見えない川底に対してオペレータが無茶な動作を強要して掘削部を壊してしまうという問題があるため、それを防止するアシストモードの開発にも貢献した。

将来への可能性

鉄道、自動車、農業、ビルディングオートメーション等、CPSの応用範囲は広い。設計空間探索、すなわちモデリングによる事前確認において、ソフトウェアの果たす役割は大きく、ますますの活用が望まれる。



写真3 Peter Gorm Larsen 教授

4. パネルディスカッション模様

両氏のほかに、産業界から一般社団法人組込みシステム技術協会（JASA）技術本部長の漆原憲博氏、学問分野からは形式手法に造詣の深い九州大学教授の荒木啓二郎氏、そして官の立場から SEC 所長の松本隆明の計5名で、先進的な信頼性向上技術の導入推進に関するパネルディスカッションを実施した。

ソフトウェアの開発技術は、少なくとも日本ではまだまだ浸透しておらず、例えばアジャイル開発についてのアンケートでは、78%は導入を試みておらず、5%は試みたがうまくいかなかったという報告がなされているとの課題認識が提示された。その課題に対して、次のような事項への改善を行っていけば、今後の導入推進と産業界への寄与が期待できるのではないかという議論がなされた。



写真4 パネルディスカッションの様子

- ・ 多重下請けの産業構造の得失はさておき、発注側の強力なリーダーシップがあれば、受注側においても先進的技術の導入推進が進むと想定される。
- ・ 技術そのものが難しい点については、軽くて導入しやすい手法から始めるのが良い。
- ・ 特定業界向けに特化した手法があれば、個々の分野での進展が期待できる。
- ・ 技術者育成のための産学連携をもっと進めていく必要がある、大学から見ればドメインエキスパートと思われる産業界の人と議論できる場が欲しい。
- ・ わかりやすい言葉で統一感を持って示していくことが必要で、ISO 26262のような標準規格準拠は統一感の醸成に有効である。また、インフォーマルとフォーマルをつなぐ橋渡しのような概念と噛み砕いた解説があれば望ましい。

ほかにも、プロジェクトマネージャのスタンスや契約書のあり方に関して着目していく必要があるという意見が提示された。

5. おわりに

欧州では、製品やシステムそのものに利用されるソフトウェアだけではなく、物理的なシステムを脳からコントロールしたり、開発前にシミュレーションする役割としてのソフトウェア、すなわち、サイバーフィジカルシステムが話題であることが窺えた。IPAは、今後も諸外国との情報交換を通じて、ソフトウェア・エンジニアリングの最新動向を発信していきたいと考えている。

安心して使えるソフトを教えて！

IPA 顧問

松田 晃一

知らない内にネットバンキングの自分の口座からお金が無くなっていた、パソコンに入れておいた大事な住所録がいつの間にかどこかへ漏れていた、普通に使っていたのにいつの間にかパソコンの大事な情報が消えていた、充電中のパソコンから火が出た、急に現金が入用になったのに銀行のATMが一日止まっていた……。

これらは、どれもソフトウェアの不具合が原因で、実際に起こった事故です。ITが我々の生活の隅々に入り込んで無くてはならないものになっている現代社会では、便利さの裏に数々のリスクもまた忍び寄っています。そのようなリスクをできるだけ避けるために、できるだけ安心できる製品やサービスを選んで使いたい、というのはもっともなことです。しかし一般のユーザがそのような選択が簡単にできるのでしょうか？「安心して使えるソフトはどれなの？」は、大多数のユーザにとって切実な問い掛けです。

このロボットの制御ソフトは安心ですか？

近い将来には、ロボットが家庭にどんどん入って家事を手伝ってくれる、空を飛び交うドローン（無人航空機）が自宅へ荷物を届けてくれる、自動運転の自動車が街を行き交う。そんな世の中が直ぐにもやって来そうです。歳をとって身体能力の衰えを感じ始めると（もちろん知的能力も、ですが）こんな世の中が切実に待ち遠しくなります。しかし、一方でこれらのシステムの安全性、中でも制御の中核となるソフトウェアの信頼性はとても気になることです。これまでのソフトウェア事故の大半は、経済的な損失にとどまっていたが、これからは人身への被害という、経済損失とは比較できない重大な問題を引き起こす可能性があります。ソフトウェアのバグによって人間に危害を加えることがあってはなりません。と言っても、ソフトウェアのバグを完全に0にすることも実際問題としては不可能です。どの程度までが許容できるか、社会的なコンセンサスを形成していくほかありません。

自動運転は人間の運転より危険ですか？

例えば、自動運転自動車です。昨年は、年間で4,100人余の方が自動車事故で亡くなっているとのこと。そして、そのほとんどすべてが運転者である人間のエラーが原因です。もし、自動運転システムのエラーが、人間のエラーに比べて圧倒的に少なく、例えば100分の一程度ならば、交通事故の死者数は数十人に激減する可能性があります。日本の現状では自動運転と言っても、運転はあくまで人間がするものであって、システムは人間を補助するものだという位置付けのようですが、人間の判断ミスや操作誤りの発生と自動運転システムの誤作動とはどちらの発生率が高いのでしょうか？たぶん、人間の漫然とした運転、脇見、不適切な操作などが、システムの誤動作より圧倒的に多いのではないのでしょうか？緊急時にパニックになった人間が下手に手を出すより、システムに任せた方が余程危険回避の可能性が高いように思います。自動操縦装置の動作と矛盾した操作をパイロットが行ったために多数の死者を出す大事故となった中華航空機事故を思い出します。無条件に機械を信奉するつもりはありませんが、自動運転（自律運転）自動車は受け入れるべく環境整備を進めては、と思います。

コンセンサス作りに向けたユーザ視点での品質説明を

いずれにしても、このような問題は自動運転システムの品質評価などの客観的なデータをもとに、専門家だけではなく一般市民を含む幅広い社会的な議論を通してコンセンサスを形成していくことが何より重要でしょう。

色々な品質指標の定義や測定したデータだけを専門家から示されても、それをどう解釈し、どう判断すればいいのか一般市民にとってはほとんど判りません。専門家は社会的な議論に耐えられるように、ユーザ視点からみた品質について分かりやすい形での情報提供とリスクを含めた丁寧な説明が最も求められることではないでしょうか。

編集後記

皆様、新しい年をどのように迎えられましたでしょうか。今年の元旦は拙宅の横浜に雪が降り、うっすら雪化粧の静かな年明けとなりました。さて、SEC journal 39号をお届けします。今号では、「品質について考える」をテーマに、IoT時代の品質とは何かを、国際規格の動向を軸にあらためて皆様と考えてみたいと思います。執筆陣の方々もご専門であるだけに、ボリュームのある仕上がりとなりました。その中で所長対談にご登場願った英国 RSSB のヒュー・ギブソン氏からの、鉄道のように大規模な人手によって運用されているシステムの安全問題にはヒューマンファクターによる分析が不可欠で有効という言葉に、「原点回帰」という言葉が浮かびました。今年も皆様にとって良い年になりますよう、願っております。

(編集長)

編集部より

次世代のソフトウェア・エンジニアリングに関して等、忌憚のないご意見をお待ちしております。下記の FAX またはメールにてお気軽にお寄せください。

SEC journal 編集部 FAX : 03-5978-7517 e-mail : sec-journal_customer@ipa.go.jp

SEC journal 編集委員会

編集委員長	杉崎真弘
編集委員 (50 音順)	荒川明夫
	石川智
	石橋正行
	日下保裕
	杉浦秀明
	中尾昌善
	長谷川佳奈子
	三原幸博
	室修治
	山下博之



元旦の井草八幡（東京・杉並区）の風景（撮影：K. Hasegawa）

SEC journal 第10巻第5号（通算42号） 2015年1月31日発行

© 独立行政法人情報処理推進機構 2015

編集兼発行人 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
所長 松本隆明
〒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 16階
Tel : 03-5978-7543 Fax : 03-5978-7517
URL : <http://www.ipa.go.jp/sec/>
e-mail : sec-journal_customer@ipa.go.jp

※本誌は「著作権法」によって、著作権等の権利が保護されている著作物です。

※本誌に掲載されている会社名・製品名は、一般に各社の商標または登録商標です。

SEC journal 論文募集

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では、下記の内容で論文を募集しています。

論文テーマ

- ・ソフトウェア開発現場のソフトウェア・エンジニアリングをメインテーマとした実証論文または先導的な論文
- ・ソフトウェアが経済社会にもたらす革新的効果に関する実証論文

論文分野

品質向上・高品質化技術、レビュー・インスペクション手法、コーディング手法、テスト/検証技術、要求獲得・分析技術、ユーザビリティ技術、プロジェクト・マネジメント技術、設計手法・設計言語、支援ツール・開発環境、技術者スキル標準、キャリア開発、技術者教育、人材育成、組織経営、イノベーション

応募要項

締切り：1月・4月・7月・11月 各月末日

査読結果：締切り後、約1カ月で通知。「採録」と判定された論文はSEC journalに掲載されます。

応募方法：投稿は随時受付けております。応募様式など詳しくはHPをご覧ください。

<http://www.ipa.go.jp/sec/secjournal/papers.html>

SEC journal 論文賞

毎年「採録」された論文を対象に審査し、優秀論文にはSECjournal論文賞として最優秀賞、優秀賞、所長賞を副賞と併せて贈呈します。

ITパスポート試験のご案内

ー ビジネスにITを活用する すべての社会人のための「国家試験」ー

- ビジネスにITを活用するためには、情報システム部門に限らず、利用する側の社員一人ひとりにも“IT力”が求められています。
- iパス（ITパスポート試験）は、セキュリティ、ネットワーク等のITに関する基礎知識をはじめ、企業活動、経営戦略、会計や法務、プロジェクトマネジメントなど、幅広い総合的知識を測る国家試験です。
- iパスを通じて、社員一人ひとりに“IT力”が備わることにより、組織全体の“IT力”が向上し、様々なメリットが期待されます。

iパスのメリット

ITを活用した業務効率化とビジネス拡大に！

iパスを通じて習得したITの基礎知識を活かすことで、業務にITを積極的に活用し、業務効率化につながります。また、ITに関する基礎知識は、社内の情報システム部門等との円滑なコミュニケーションにも役立ちます。営業職であれば、顧客に対して製品やサービスを具体的にわかりやすく説明できるようになり、顧客のニーズをより深く把握できるようになり、ビジネスチャンスの拡大にもつながります。

情報セキュリティ対策・コンプライアンス強化に！

社員一人ひとりが、情報セキュリティやモラルに関する正しい知識を身につけ、意識することで、情報セキュリティに関する被害を未然に防ぐことができ、「情報漏えい」などのリスク軽減、企業内のコンプライアンス向上・法令順守に貢献します。

経営全般に関する知識など幅広い知識がバランスよく習得できる！

iパスは、ITに関する知識にとどまらず、企業活動、経営戦略、会計や法令など、ITを活用する上で前提となる幅広い知識がバランスよく習得できます。そうした知識が身につくことにより、業務の課題把握と、ITを活用した課題解決力が備わり、組織全体の業務改善につながります。

詳しくは、iパス Web サイトをご覧ください。<https://www3.jitec.ipa.go.jp/JitesCbt/index.html>

※企業の活用事例、企業の声、合格者の声など魅力的なコンテンツがご覧になれます。

IPA Better Life with IT

SEC Journal No.39
第 10 卷第 5 号 (通巻 42 号)
2015 年 1 月 31 日発行

© 独立行政法人情報処理推進機構

ISSN 1349-8622



古紙パルプ配合率70%再生紙を使用

