2018年8月8日発行
2020年8月1日改訂
第14巻 第1号（通巻56号）
ISSN 1349-8622

SEC journal 53

巻頭言
松本 隆明 IPA 顧問

論文
車載システムを想定した機能安全機能とセキュリティ機能の統合要件定義手法
左近 透 兵庫県立大学／中本 幸一 兵庫県立大学

日本のソフトウェア技術者の生産性と労働条件の決まり方：アメリカ，中国，フランス，ドイツとの比較を交えて
中田 喜文 同志社大学

企業固有スキルのレベルを判断するための社内独自試験実施の取り組みとその効果
清谷 佳史 NTTデータシステム技術株式会社

国際規格に基づく総合的なソフトウェア品質評価の枠組みとその実製品への適用による品質ベンチマーク
鷲崎 弘宜 早稲田大学 国立情報学研究所 株式会社システム情報 株式会社エクスモーション／
津田 直彦 早稲田大学／本田 澄 早稲田大学／中井 秀矩 早稲田大学／深澤 良彰 早稲田大学／
東 基衛 早稲田大学／込山 俊博 日本電気株式会社／中野 正 一般社団法人コンピュータソフトウェア協会／
鈴木 啓紹 一般社団法人コンピュータソフトウェア協会

FRAM（機能共鳴分析手法）による成功学に基づく安全工学
野本 秀樹 有人宇宙システム株式会社／道浦 康貴 有人宇宙システム株式会社／
石濱 直樹 研究開発法人宇宙航空研究開発機構／片平 真史 研究開発法人宇宙航空研究開発機構

ソフトウェア品質技術が品質特性に与える効果の見える化とその検証
小島 嘉津江 富士通株式会社／森田 純恵 株式会社富士通ゼネラル／廣瀬 竹男 富士通株式会社／
若本 雅晶 株式会社富士通研究所／菊池 慎司 株式会社富士通研究所／
楾 晃歓 株式会社富士通ビー・エス・シー／鷲崎 弘宜 早稲田大学

ソフトウェア技術者エントリ層教育コースによる人材育成とその評価
福間 和人 三菱電機株式会社／田村 直樹 三菱電機株式会社／藤岡 卓 三菱電機株式会社／

特集
SEC 2017年度活動概要

寄稿
SAPの成功：ドイツの制度環境からの一考察
山内 麻理 同志社大学 客員教授 フランス国立労働経済社会研究所(LEST) ドイツ日本研究所(DIJ) 客員研究員

IEC 62853と「つながる世界の開発指針」
中川 雅通 DEOS協会 技術部会 パナソニック／山浦 一郎 DEOS協会 技術部会 富士ゼロックス／
森田 直 DEOS協会 標準化部会 株式会社ソニーコンピュータサイエンス研究所／
武山 誠 DEOS協会 標準化部会 神奈川大学／木下 佳樹 DEOS協会 標準化部会 神奈川大学

IPA 独立行政法人情報処理推進機構
Information-technology Promotion Agency, Japan

SEC journal No.53
Vol.14 No.1 Aug. 2018

1 巻頭言
SEC journalを振り返る
松本 隆明 IPA 顧問

2 論文
車載システムを想定した機能安全機能とセキュリティ機能の統合要件定義手法
左近 透 兵庫県立大学／中本 幸一 兵庫県立大学

日本のソフトウェア技術者の生産性と労働条件の決まり方：
アメリカ，中国，フランス，ドイツとの比較を交えて
中田 喜文 同志社大学

企業固有スキルのレベルを判断するための社内独自試験実施の
取り組みとその効果
清谷 佳史 NTTデータシステム技術株式会社

国際規格に基づく総合的なソフトウェア品質評価の枠組みと
その実製品への適用による品質ベンチマーク
鷲崎 弘宜 早稲田大学 国立情報学研究所 株式会社システム情報 株式会社エクスモーション／津田 直彦 早稲田大学／
本田 澄 早稲田大学／中井 秀矩 早稲田大学／深澤 良彰 早稲田大学／東 基衛 早稲田大学／
込山 俊博 日本電気株式会社／中野 正 一般社団法人コンピュータソフトウェア協会／
鈴木 啓紹 一般社団法人コンピュータソフトウェア協会

FRAM（機能共鳴分析手法）による成功学に基づく安全工学
野本 秀樹 有人宇宙システム株式会社／道浦 康貴 有人宇宙システム株式会社／石濱 直樹 研究開発法人宇宙航空研究開発機構／
片平 真史 研究開発法人宇宙航空研究開発機構

ソフトウェア品質技術が品質特性に与える効果の見える化とその検証
小島 嘉津江 富士通株式会社／森田 純恵 株式会社富士通ゼネラル／廣瀬 竹男 富士通株式会社／若本 雅晶 株式会社富士通研究所／
菊池 慎司 株式会社富士通研究所／椋 晃歓 株式会社富士通ビー・エス・シー／鷲崎 弘宜 早稲田大学

ソフトウェア技術者エントリ層教育コースによる人材育成とその評価
福間 和人 三菱電機株式会社／田村 直樹 三菱電機株式会社／藤岡 卓 三菱電機株式会社／

66 特集：SEC 2017年度活動概要
IoT時代の安全安心に向けて
「つながる世界の開発指針」の展開状況
システムズエンジニアリングの推進
先進的な設計・検証技術の適用事例収集・公開
重要インフラ分野等システム/製品の障害対策
定量的管理による信頼性・生産性向上
システム構築能力の強化
上流工程の課題解決に向けて
システム理論に基づく安全性解析手法STAMP/STPAの普及促進
制御システム向けのセーフティ及びセキュリティ対策
コーディング作法ガイド（ESCR）の整備
ソフトウェア工学分野の先導的研究支援事業について
プロモーション活動

88 寄稿
SAPの成功：ドイツの制度環境からの一考察
山内 麻理 同志社大学 客員教授 フランス国立労働経済社会研究所(LEST) ドイツ日本研究所(DIJ) 客員研究員

IEC 62853と「つながる世界の開発指針」
中川 雅通 DEOS協会 技術部会 パナソニック／山浦 一郎 DEOS協会 技術部会 富士ゼロックス／
森田 直 DEOS協会 標準化部会 株式会社ソニーコンピュータサイエンス研究所／武山 誠 DEOS協会 標準化部会 神奈川大学／
木下 佳樹 DEOS協会 標準化部会 神奈川大学

98 編集後記
［C言語版］ESCR Ver.3.0書籍化/国家試験 エンベデッドシステムスペシャリスト試験のご案内

巻頭言

SEC journalを振り返る

松本 隆明
IPA顧問

IPA/SEC は、当時から IT システムの中核を担うようになってきたソフトウェアの高品質化、高生産性化を推進するために、2004 年 10 月にソフトウェア・エンジニアリング・センターとして IPA に設立されました。その後、IT システムを取り巻く環境は大きく変化し、高信頼、高生産性を保つことはもちろんであるが、更にいかにして IT によりイノベーションを創出するかという、いわゆる守りの IT から攻めの IT へのシフトが叫ばれるようになってきました。攻めの IT へのシフトにあたっては、最近話題となっている IoT、AI、ビッグデータといったイノベーションにつながる新技術の開発だけでなく、そうした新技術をいかにして社会実装につなげていくかといった、技術以外の側面も含めた進展がなければ実現できません。こうした状況に対応すべく、2018 年度から新たにスタートした IPA の第 4 期中期事業計画において、組織や役割の一部見直しが行われています。

SEC journal は、IPA/SEC の諸活動の一環として、設立わずか 3 カ月後の 2005 年 1 月に創刊号を発刊しています。本号で 53 号を迎える SEC journal 創設の目的は、新たに設立した IPA/SEC の活動を広く知ってもらうという広報誌的な意味合いよりも、当時はソフトウェア・エンジニアリングに関して学術的な論文発表の場はあっても、実践的な論文に関する公開の場が少なかったため、現場での応用に力点を置いた情報発信の機関誌を作ろうという思いがあったようです。

ソフトウェアは、人間の知的創造物であるが故に、同じ機能を持つソフトウェアでも作る人によって出来上がるソフトウェアは千差万別となります。従って、教科書的な指導書に従って画一的な手法で開発すれば、だれでも高品質なソフトウェアを効率良く作れるというわけではありません。そこで必要となるのが、実際の開発現場での実践例です。プラクティカルでエンピリカルな情報があってこそ、色々なエンジニアリング手法が実践的な方法論として開発現場で活用できることになります。その意味で、SEC journal がこれまで果たしてきた役割は極めて大きかったと思います。

あらためて SEC journal の掲載論文や解説記事を振り返ってみると、ソフトウェア開発に関するトレンドの変遷がよくわかります。刊行当初は、ソフトウェア開発者のスキルセットをどのように定めて人材の育成を行っていくかという議論が多数を占めていました。IT エンジニアの不足が大きな問題として注目され始めていた時期だと思います。とくに、組込みソフトウェアの分野では急激なニーズの高まりもあって、人材の育成は喫緊の課題として捉えられていました。

その後は、設計の上流工程の品質をどのように上げていくかといった議論が目に付きます。ソフトウェアの品質確保には上流工程の品質を上げることが最も重要であるという認識が高まってきたこともあると思いますが、当時ユーザ企業とベンダ企業間で開発プロジェクトの失敗をめぐる訴訟問題などが数多く起きていたことも起因していたのではないかと思います。設計の早い段階でユーザとベンダで抜け漏れなく開発内容を合意しておかないと後々大きな問題につながるという危機感が高まっていたということでしょうか。

最近はシステム思考に関する議論が増えてきました。IoT の普及により様々なモノがつながってサービスを実現するような時代になってくると、これまでのようなハードウェアとソフトウェアの設計、あるいは、ソフトウェアも組込みとエンタプライズの設計を独立に行っていたのでは全体として正しい設計ができなくなります。システム全体を俯瞰的に捉え、全体最適で系統的に設計していかないと立ち行かなくなってきたため、システム思考のやり方がより重要になってきました。IPA/SEC も組織としてシステム思考に積極的に取り組んできたこともあってそうした議論が増えてきました。

SEC journal 刊行から 13 年がたち、IT システム開発は当時に比べれば比較にならないくらいエンジニアリング的な方法で開発されるようになってきたと思います。産官学のメンバが一緒になって議論できる場として、SEC journal は極めて貴重な機関誌としてその役割を果たしてきたとあらためて思います。また、所長対談では、それぞれの分野における第一人者の方々にご登場いただいて、最新のトレンドや技術動向を分かりやすく解説いただき、読者の方々には極めて有意義な情報をお届けできたのではないかと自負しています。最後に、これまで SEC journal の継続的な刊行にご尽力いただいた数多くの関係者、ご協力いただいた方々に感謝申し上げると共に、今後とも読者の方々をはじめ皆様の更なるご支援をよろしくお願い申し上げます。

車載システムを想定した機能安全機能とセキュリティ機能の統合要件定義手法

左近 透[※1]　　中本 幸一[※1]

車両ソフトウェア開発における，ISO 26262 と自動車セキュリティ分析ガイド JASO TP15002 をベースとした機能安全機能とセキュリティ機能の要件定義手法を提案する．機能安全とセキュリティ双方の要件定義を統合するための動作モデルと開発プロセスモデルを定義した．これは安全に対する要求を詳細化し，要件を定義する過程で，セキュリティ脅威を詳細化しセキュリティ要件を特定するものである．また，セキュリティリスク評価を，機能安全要件の定義段階に応じて行う．まず, ASIL 導出時に利用する過酷度と回避可能性に対応する対処の規準値となるリスクスコアを定める．更に具体的な脅威の特定が可能になった時点でリスクスコアリングを実施し，そのスコアから対応すべき脅威を選択し，対処すべきセキュリティ機能を決定する．

A requirement definition method to integrate functional safety and security for vehicle software development

Toru Sakon[※1], Yukikazu Nakamoto[※1]

We propose an integrated requirement definition method the functional safety based on ISO 26262 and the security functions based on the risk assessment guide JASO TP-15002 in vehicle software development. We define a behavior model and a development process model. The security risk assessment is performed depending on the development phases in the functional safety. The threshold of risk scores are determined corresponding to the severity and evasiveness to be used when ASIL is derived. The risk scoring is carried out when the specific threats can be derived. With the score, the threats to be dealt with are identified and the security functions for the threats are obtained.

1　はじめに

自動車や製造機械，発電所，鉄道や医療用器械に代表される制御機器の設計に際しては，故障や動作異常の発生により生命や設備を危険にさらすことの阻止が優先度の高い目標である．安全には，危険そのものを発生させない本質安全と，危険につながる事象が発生した場合に，それに対処する機能により危険回避を行う機能安全がある．ISO 26262[1] は自動車の機能安全に関する規格である．

一方，制御機器の安全動作に対する脅威として，機器の故障などに加えて，サイバー攻撃の懸念が高まりつつある．また，研究者による自動車のセキュリティホールを利用した攻

※1　兵庫県立大学(University of Hyogo)

撃の発表や，プラントに対するサイバー攻撃の実例も発生した。ICS-CERT によると 2009 年以降，インシデント報告は増え続け，2013 年の ICS-CERT 報告書[2] では 250 件超の報告がされている．

このような制御機器に対するサイバー攻撃に対する対応として，電気・電子・プログラマブル電子機能安全システムに対する機能安全規格，IEC 61508 に対応したセキュリティ規格，ISA/IEC 62443[3] が制定されている．自動車産業においても，米国 SAE が開発した SAE J3061 -Cybersecurity Guidebook for Cyber-Physical Vehicle Systems[4] などサイバー攻撃に対するガイドラインや規格が提案されている．

しかし，機能安全とセキュリティは安全動作保護の観点から密接に関係すると考えられるが，開発プロセスや相互の機能の関係などは明確でなく，議論が継続されている．例えば，SAE J3061 はセキュリティ機能開発プロセスを中心に規定して記述されており，機能安全機能との関係性は，対応する機能安全開発プロセスとの情報交換が例示されているのみである．同種の規格，ガイドラインである EVITA[5] などでも同様である．

本論文では，車両開発を想定した場合の機能安全機能とセキュリティ機能の統合要件定義手法を提案する．まず，開発対象の定義を行うアイテム定義では，セキュリティ攻撃にさらされる点とそこで発生し得る脅威の組み合わせである Exposure Control Point（以下，ECP と略す）を導入する．次にハザード分析とリスクアセスメントでは，セキュリティ分析手法に，JASO TP15002 で記載されている車載システム向けセキュリティ分析手法を適用する．ECP をハザード要因とし，対応するハザード事象及び回避可能性を求め，これらを元にした対策基準指標を決定する．次に機能安全規格の機能安全コンセプトとシステム設計に相当したセキュリティ仕様の導出において，ECP を段階的に詳細化して脅威を具体化しつつ対応する対策を定める．この各段階において，機能安全における故障モード対策と脅威対策との対応を確認させ，対応しない部分について技術セキュリティ要求の定義を行う．

2 関連研究・標準

自動車におけるセキュリティ機能開発に関連する標準は幾つか提案されている．この中で，明示的に機能安全開発とセキュリティ開発の関連性について述べたものに，SAE J3061[4] がある．しかし，概念的な説明にとどまり，手法や統合モデルの提案には至っていない．一方，研究レベルでは，安全とセキュリティに関するリスクを統合して扱うリスクアセスメント手法[6][7][8] が幾つか検討されている．しかし，統合された手法による機能安全機能とセキュリティ機能の要件定義プロセスの検討は行われていない．

本研究と同じく要件定義に対する研究では Automotive Spice をベースにした Macher らの研究[9] がある．Macher らは，研究[10] で提案されたリスクアセスメント手法を利用する．この手法は，機能安全のハザード分析を実施するタイミングで脅威分析を行い，ハザードの大きさや脅威の起こしやすさなどから対処のレベル分けを決定する．脅威への対策要件は，車両ネットワークの階層から対策を検討する静的手法，更に機能間の呼び出し階層から対策を検討する動的手法で検討される TrustZone と呼ばれる境界面を定義し，その境界面からの攻撃を防御する方策として対抗策が検討される．この境界面上のハードウェアとソフトウェア間のインターフェース（HSI）にレベル付けされた脅威対抗策が配置される．Macher らは仮想ステアリングシステムを対象に要件分析を行った．

Macher らの研究では，機能安全のハザード分析と同じタイミングでリスクアセスメントを行うため，機能安全と並行してセキュリティ要件分析のプロセスが定義できる．しかし，脅威の起こしやすさなどは対象システムの詳細な情報を必要とする場合が多く，ハザード分析段階ではこれらの情報は必ずしも入手できない．そのため，リスクアセスメント時のリスクスコアリング値が本来のリスクを正しく反映していない可能性がある．また，ISO 26262 の各プロセス段階に相当する作業は定義されているが，それら作業及び出力と各プロセスの入力情報が関連付けられていない．また，脅威から機能を保護する機構の要件分析であり，脅威発生時の車両の安全な振る舞いの検討は行われていない．

3 ISO 26262とJASO TP15002

自動車制御システムの機能安全規格である ISO 26262 と，自動車のセキュリティ分析を目的としたセキュリティ分析ガイド JASO TP15002[11] の概要を述べる．

3.1 ISO 26262

ISO 26262 は，自動車の機能安全の実現のための安全管理，要件定義，設計，製造，評価 / 検証，生産と運用，それを支える要件定義や開発上必要支援プロセス，並びにガイドラインなどから成る．自動車の機能安全とは，通常動作状態で機器の故障が発生し故障モードに陥った場合，機能安全機能により事故が発生しないような安全状態への移行を実現することである．開発プロセスは，ウォーターフォール型プロセス（V 字プロセス）を基本として定義されている．ここでは，本論文の検討対象である要件定義からシステム設計に至るプロセスを述べる．

(1) アイテム定義（開発対象定義）

アイテムとは，車両観点で見た機能を実現する開発対象である．形式的には，センサなどの外部入力，演算などの内部処理，アクチュエータへの出力から成る要素を組み合わせたものである．例えば，パワーステアリング機能は，ハンドルにかかるトルク，車速と舵角を入力とし，それに対応したトルク出力を計算し，アクチュエータにトルク調整出力を出す（図 1）．

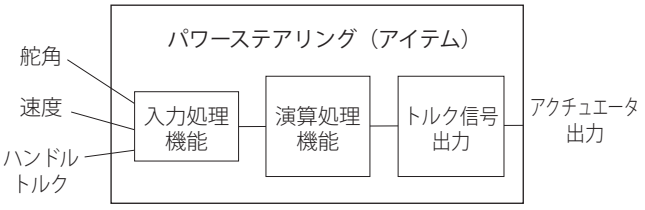


図 1　パワーステアリングアイテムの構成

(2) ハザード分析とリスクアセスメント

ISO 26262 ではリスクを「危害発生確率とその危害の過酷度との組み合わせ」で定義している．

そのために，まず，ハザード（危機事象）とハザードが発生する状況を特定し，帰納的 / 演繹的手法で発生可能性の分析を行う．一般には，HAZOP（Hazard and Operability Study）手法で不作動，勝手な動作，過大，過小などのガイドワードで制御出力の期待値からのズレを想定し，ハザードを導出する．次に FTA（Fault Tree Analysis）でハザードの原因となる事象を洗い出す．最後に FMEA（Fault Mode and Effect Analysis）を適用し，個別の故障要因から FTA の正しさを検証する．

次に特定されたハザードと状況に対してリスクアセスメントを行う．ISO 26262 では不具合発生時にその状況や操作者の回避可能性を考慮した，達成すべき安全度の指標としてのリスク低減目標 ASIL（Automotive Safety Integrity Level）を定める．ASIL はハザードの過酷度，発生頻度，回避可能性から，定められた分類表に従って 4 段階の ASIL と一般品質保証レベルに分類される．開発では，ASIL のレベルに応じた開発手法や検証，システム構成を取ることが要求される．

この後，ハザード発生に対する安全機能を実現するための要件（セーフティゴール）を導出する．

(3) 機能安全コンセプトの導出

セーフティゴール実現のために，アイテムを構成する各要素またはアイテムの外部に，安全実現のための機能を割り当てた機能安全コンセプトの導出を行う．

まず，セーフティゴールを詳細化して，達成に必要な機能（機能安全機能，以下，安全機能と略す）を導出する．次に，機能安全要求の抽出を行う．機能安全要求とは，セーフティゴールに基づくアイテムの安全な振る舞いの仕様，及び実装に依存しない安全方策である．更に，作業時点で判明しているアイテムの具体的な構成に基づいて機能安全要求をアイテムの構成要素またはアイテム外部の実装に割り当て，機能安全コンセプトの導出が完了する．安全機能の追加により，アイテムの構成にも安全機能が追加される（図２）．

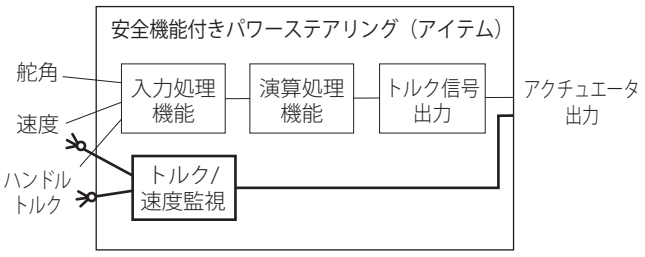


図２　安全機能付きパワーステアリングアイテムの構成

(4) 技術安全要求の導出とシステム設計

安全機能の設計対象であるアーキテクチャ（前提アーキテクチャ）に当てはめて，実装に関する要件を導出したものが技術安全要求である．これらをアイテムの構成要素に安全機能実装要件として割り当てたものが技術安全コンセプトである．システム設計では技術安全要求を元に，ハードウェア・ソフトウェア・ハードソフトインターフェースに機能要求を割り振り，実装仕様が決定される．

3.2 JASO TP15002 セキュリティ分析

JASO TP15002 は公益社団法人自動車技術会の開発した自動車システム向けのセキュリティ分析ガイドである．このガイドのセキュリティ分析は，分析対象の定義，脅威分析，リスクアセスメント，対策方針決定，セキュリティ要件の選択という 5 つのフェイズから成る．

分析対象の定義では，分析対象の構成要素及び構成要素間のデータフローのモデルを作成する（図 3）．更に構成要素ごとに，保護すべき機能とデータを保護資産として抽出する．

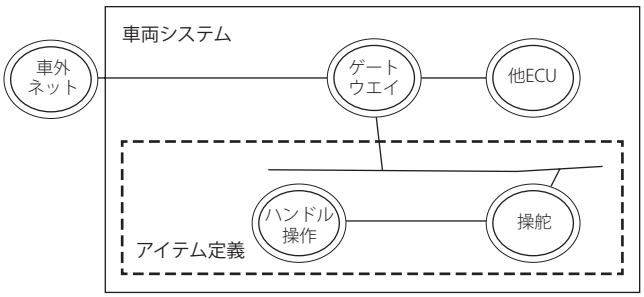


図３　分析対象モデル例

次に対象システムのライフサイクルで対象システムに関与可能な人・組織（ステークホルダ）を定める．これは，次の脅威分析に際して，対象システムの攻撃目標や攻撃傾向・動機の分類に利用する．

脅威分析では，まず分析対象システムの置かれた環境や条件を定義する．次に当該条件における保護資産に対する脅威を，どこ（Where）から，誰が（Who），いつ（When），どのような動機（Why）で，どのように引き起こされる脅威（What）かの形式で整理する．

リスクアセスメントでは，抽出された脅威のスコアリングを行う．保護資産の重要度と攻撃困難さを元にする CRSS(Common Risk Scoring System) と RSMA（Risk Scoring Methodology for Automotive system）が手法として示されている．ある基準以上の脅威は FTA などで原因を抽出，対策方針を決定する．

最後にセキュリティ機能要件とセキュリティ保証要件を定義する．前者は，ISO/IEC15408 CC（Common Criteria）Part2 で定義されるセキュリティ機能コンポーネントの実装を要件としたものである．後者は，CC Part3 の Evaluation Assurance Level で指定し，相当する保証要件を確認検証する．

4　機能安全･セキュリティ開発の問題点

機能安全・セキュリティ開発で解決すべき考える問題点を列挙する．

(1) 安全機能とセキュリティ機能の関係の明確化

セキュリティ脅威の一部は安全機能で対処可能な場合があ

る．例えば，車載ネットワークの Flooding によるサービス拒否攻撃では，当該ネットワークに接続された ECU 間の通信が途絶する．しかし，安全機能で，断線などによる ECU 間の通信途絶が想定される場合は安全機能により安全状態へ移行する．しかし Flooding による車載ネットワーク攻撃は複数の機能が同時に影響を受ける場合が想定され，安全機能の想定外の挙動をする可能性がある．また，セキュリティ機能の対応の結果，安全機能が想定しない故障モードを引き起こし，ハザードを引き起こす可能性もある．

したがって，機能安全とセキュリティそれぞれで想定している故障モードの内容と脅威内容を対比し，更に対処決定の際，対処内容の相互に与える影響を検討する手順の定義が必要である．

(2) 安全とセキュリティのリスク評価手法の相違

ISO 26262 では，アイテムが定義された次の段階で脅威分析とリスクアセスメントが実施される．ハザードが抽出されたのち，ハザードとその発生状況における過酷度，発生頻度，回復可能性からリスク軽減度の目標である ASIL を定める．

一方，JASO TP15002 おけるリスク評価は保護資産の特定，保護資産に対する脅威分析と，脅威の結果引き起こされる被害の評価で実施される．ところが，これら資産が，ISO 26262 のリスク評価段階で明確になっているとは限らない．例えば，ある資産は実装方式が決まった後に明らかになる．これは，ISO 26262 の機能安全コンセプト導出以降の作業に相当する．すなわち，安全とセキュリティで，プロセス上の同じタイミングでリスク評価が実施できないことを意味する．

(3) 安全とセキュリティの機能仕様導出方法の相違

ISO 26262 では，アイテム定義→ハザード・リスクアセスメント→機能安全コンセプト導出→技術安全要求導出→システム設計と段階が進むに従い，詳細化された各機能の故障に対応して安全機能が仕様化される．一方，セキュリティ機能設計では，データフローダイヤグラムによる抽象モデルの作成後，攻撃可能地点の決定，網羅的な脅威の抽出と，リスク評価実施後，特定された対応すべき脅威に対してセキュリティ機能が仕様化される．この詳細化の過程の相違により，要件の分析や仕様作成の各段階において安全機能と，セキュリティ機能との対応を取る事が複雑になる．

5　提案手法

本論文では，機能安全とセキュリティ開発を統合するモデルを設定する．モデルは機能安全・セキュリティ動作モデルと機能安全・セキュリティ要件定義統合プロセスから成る．

(1) 機能安全・セキュリティ動作モデル

本論文での機能安全・セキュリティ機能の動作モデルを図 4 に示す．

動作モデルは 6 要素で構成される．

①　車両本来機能：通常状態での車両機能部．制御機能（コントローラ）と車両のセンサ類やほかの ECU，運転車が相互に情報を交換して機能している．

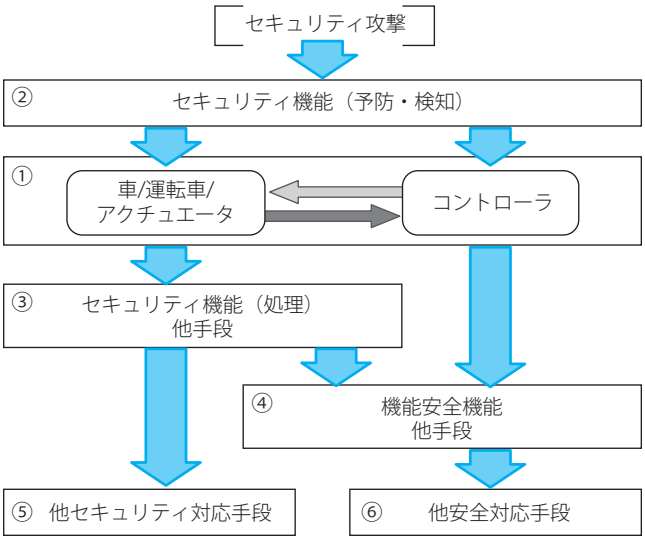


図 4　機能安全・セキュリティ動作モデル

②　セキュリティ機能（予防・検知）：車両本体機能及び安全機能をセキュリティ脅威から保護する．脅威の予防・検知機能を持つ．脅威発生の場合，セキュリティ処理機能を起動する．

③　セキュリティ機能（処理）セキュリティ脅威が発生した場合の処理機能．脅威に対する直接対処や，安全機能へ処理を移行し安全状態へ移行させる．対処困難な場合には，ほかのセキュリティ対抗手段（強制リセットなどを想定）へ移行する．

④　安全機能：車両本来機能の構成要素を監視し，故障の場合は，安全状態に移行する機能を持つ．もし安全機能で対処できない場合には，ほかの手段，例えばエンジンが停止しない場合，Kill スイッチでエンジンを強制停止させるなどの手段を取る．

⑤ ⑥ 他セキュリティ対応手段・他安全対応手段：セキュリティ機能や安全機能で対応できない事象への別手段による対応を行う．

通常状態では，コントローラと車両・ドライバーやアクチュエータは正常な相互作用を行なっている．セキュリティ脅威が発生した場合セキュリティ機能（処理）を起動し，必要な対処を行う．この対処は，セキュリティ侵害事象の中で，侵害事象の結果が安全機能で故障モードに類似しているために，枠組みで対処する事が妥当なものと，そうでないものに分割し，前者については安全機能を呼び出す事により，安全モードに動作を移行する．それ以外の事象については，セキュリティ機能内の対処もしくは他セキュリティ機能の対処とする．

この動作モデルは利点を持つ．

- 車両がセキュリティ攻撃を受けた際に，ハザード事象の発生を防ぐ為に安全状態に移行することが望ましい．このモデルではセキュリティ脅威発生と安全機能を関連付けており，脅威発生時の安全状態移行を実現する動作のモデル化している．
- 上の利点の実現のため安全機能とセキュリティ機能の関係を分離・単純化している（4 節，問題（1））．これにより，

相互の機能の対応を明確化することが可能となる．

- 安全機能とセキュリティ機能の対応を取る必要から，要件定義プロセスで脅威やハザード・リスクの詳細度で整合性が確保される．（4 節，問題（2）（3））．

(2) 機能安全・セキュリティ要件定義統合プロセス

この動作モデルに基づき，機能安全とセキュリティ機能の要件定義プロセスを定義する（図 5）．

このプロセスは，機能安全の要件定義プロセスとそれに並行するセキュリティ要件定義プロセスから成る．このプロセスでは，脅威，ハザード，リスクの詳細度を機能安全とセキュリティで整合的に取り扱う必要上，並行するプロセス間では，要件分析に使われる情報の詳細度は同程度でなければならない．そのために，a）セキュリティ脅威分析は機能安全の要件分析のレベルに合わせた詳細度とする．機能安全の要件分析が進み，その詳細化に合わせて脅威を記述する．すなわち，脅威は，脅威種別，対象及び脅威の明確になった段階で記述する．b）セキュリティ脅威により損なわれてはならないのは安全である．そのため，対処の徹底度は，安全のハザード分析とリスク分析の時点で導出される．しかし，対処すべき脅威が具体化されていないため，この時点ではリスク対処の基準を作るしかない．c）セキュリティ機能自身も安全機能の対象である．そのため，機能安全で求められるレベルでのプロセスで開発されなければならない．そのために ASIL レベルが導入される．d）実装要求になる辺りで攻撃がある程度具体化されるため，脅威分析とリスクスコアリングを実施する．このときに b）での脅威を利用する．

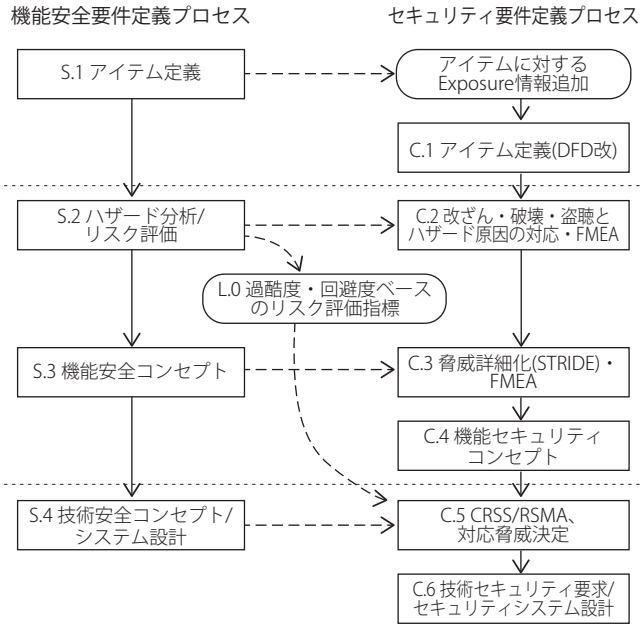


図５　機能安全・セキュリティ統合プロセス

以下，セキュリティ要件定義プロセスの各ステップを説明する．

5.1 拡張アイテム定義（図 5.C.1）

このプロセスでは，機能安全要件定義プロセスのアイテム定義（図 5 S.1）で定義されたアイテムを入力とする．プロセスの出力は，入力されたアイテムに，セキュリティ脅威の情報を追加した拡張アイテム定義である．車載セキュリティでは攻撃の開始地点は必ずしもアイテムに含まれない．例えば，車載ネットワーク上の機能上関係ない ECU からの攻撃が相当する．このような場合を含めて脅威を明示的に表記するために Exposure Control Point（ECP）を導入する（図 6）．

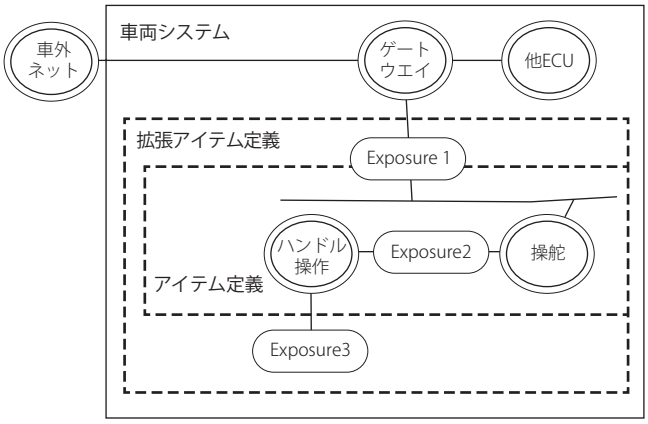


図６　拡張アイテム定義（丸四角が ECP）

この，ECP は情報セキュリティの脅威分析で Data Exposure Control Point として Fisher[12] により導入されたものを，データに対する脅威に加えて，サイドチャネル攻撃など物理的な脅威まで含むように概念拡張したものである．ECP はアイテムにおいて物理攻撃を含む脅威にさらされる点とそこで発生する脅威の組み合わせである．図 6 の場合，アイテム外のネットワーク，アイテム内部のネットワーク，操作機能への直接の脅威が ECP として表記されている．また，アイテム定義の段階の情報では，必ずしも脅威の具体化できないため抽象度の高い，改ざん / 破壊 / 開示の 3 つを脅威とする．このようにして，ECP が追加された拡張アイテム定義をプロセスの出力とする．

5.2 ハザードと脅威対応（図 5.C.2,L.0）

機能安全要件定義では，定義されたアイテムを元に，ハザードを導出し，その発生状況における過酷度，発生頻度，回避可能性から ASIL を導出する（図 5 S.2）．一方，本提案では，5.1 の拡張アイテム定義と機能安全でのハザード及びリスク分析の内容を入力とする．

初めに，各 ECP に割り当てた脅威からハザードを導出する．ハザードの導出には FMEA など帰納的手法を用いる．更に，機能安全ハザードとの対応付けを行う．もし，対応付けがない場合は，機能安全にハザードを追加し，脅威が原因のハザードに対する安全機能の要件が導出されるようにする．次に対応する機能安全ハザードの過酷度と回避可能性から，リスクスコアリングのパラメータ及び対処要否の境界値の決定を行う．ここでは具体的な数値及び決定方法は言及しない．また，セキュリティ機能の保護対象の機能の部品であると考え，保護対象機能の ASIL 値をセキュリティ機能の安全指標とする（図 5 L.0）．ここでの出力は，セキュリティを考慮したハザードが追加され詳細化された拡張アイテム定義，セキュリティ機能のリスク評価及び安全指標である．

5.3 脅威詳細化・機能セキュリティコンセプト（図 5.C.3,C.4）

ここでは，機能安全コンセプト（図 5 S.3）に対応する機能セキュリティコンセプトを導出する．この際，ECP の抽象的な脅威を詳細化する．機能安全コンセプト相当段階では，実装の前提となるアーキテクチャは定まっている．このアーキテクチャ及び拡張アイテム定義を入力とし，STRIDE 手法 [13] により脅威の詳細化を実施する．そして，先と同様に FMEA などの帰納的な手法により，新たなハザードの発生を確認する（図 5 C.3）．この場合も，既存の機能安全要件定義プロセスで分析したハザードにマッピングすることを原則とする．このことにより，セキュリティ事象とハザードに対応した安全状態を関連付ける．更に詳細化された脅威から，それぞれの脅威に対応したセキュリティ機能安全要求を導出する（図 5 C.4）．これらを前提としたアーキテクチャに割り当てた機能セキュリティコンセプトを出力とする．

5.4 リスク評価と対応，技術セキュリティ要求導出（図 5.C.5,6）

ここでは，技術安全要求に相当するセキュリティ要件の導出を行う．この段階では，技術安全要求導出段階で利用されるシステム設計の情報を元に導出された脅威を更に詳細化すると同時に，その対抗策を導出する．この詳細化された脅威に対してリスクスコアリングを行う．リスク対応を決める閾値やこれらで利用されるパラメータ値は，５．３で定められた値（図 5 L.0）を用いて行う．これにより，システム設計に対する具体的な脅威に対して，安全の観点から対処すべき脅威を選択する．最後に対処すべき脅威に対してセキュリティ機能を決定する．リスク評価の結果，対処の必要な脅威に対する直接防御（予防）及び攻撃検知，セキュリティ機能の通知と内部の安全機能と関連付けた処理の観点からのセキュリティ機能の実装要求（技術セキュリティ要求）と対応するセキュリティシステム設計を行う．

この段階で，ハザード，機能安全要求，技術安全要求，脅威，機能セキュリティコンセプト，技術セキュリティ要求の関連付けが完了する．

6 仮想的な開発ターゲットへの適用

ここでは仮想的な開発ターゲットを想定し，提案手法の適用検証を行う．ここでは，ハンドルと操舵装置の間を車載 LAN で接続した Fly-by-Wire 方式の操舵装置を例として検討する（図 6）．

6.1 拡張アイテム定義（図５S.1,C.1）

図 7 に対象システムを示す．ここでは，ハンドル操作機能と操舵機能は専用線通信で結ばれているが，速度はセンサなど外部との通信から得られる．

セキュリティ要件定義のアイテム定義では，セキュリティ脅威にさらされている部分を表現するため，データフローダイヤグラムに ECP を追加する（図８）．ECP の内容は，対象となるデータ（ここでは速度データ）や処理に対する破壊 / 改ざんとする．物理的制約条件として，通信 1 の通信路，操舵部分とハンドル操作部への直接攻撃は困難とする．この場合の ECP はセンサとの通信部分で暴露されている内容の破壊 / 改ざんとする．

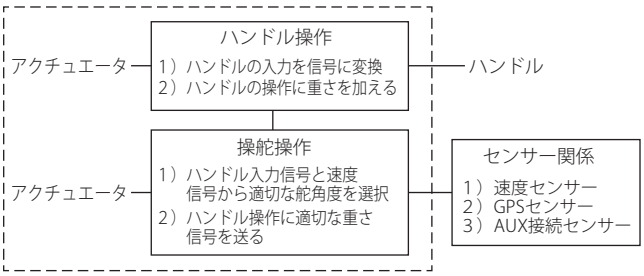


図７　対象アイテム定義（破線内）

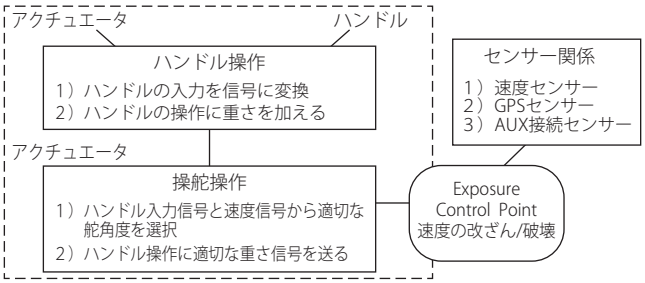


図８　拡張アイテム定義（破線内）

6.2 ハザードと脅威対応（図５S.2,L0）

次に，機能安全要件定義プロセスのハザード分析，リスク評価に対応するセキュリティ定義プロセスでの作業を行う．ここでは上のシステムで想定されるハザードのうち，速度に対応するハンドルが軽くなる場合を想定する．セキュリティ観点では通信路２には ECP としてデータ破壊及び改ざんの脅威が想定できる（表１）．

このハザードに対する過酷度及び回避可能性は機能安全側での HAZOP などを行って定められる．それに従い，後に使用されるリスクスコアリング手法でのリスク対処必要とされる閾値を定める．この決定方法はここでは議論しない．

表１　ハザード / ハザード要因と ECP

機能安全要件定義		セキュリティ要件定義
ハザード事象	ハザード要因	ECP
ハンドルの補助トルクが過大にかかって軽く動作し，ハンドルを大きく切って事故を起こす．	通信路がエラーを起こし，ハンドルと速度が連動しなくなる	破壊
		改ざん

6.3 脅威詳細化・機能セキュリティコンセプト（図 5．S3,C.3,4）

機能安全コンセプト相当段階では，ハザード要因に対して適切な機能安全コンセプトが導出されている．それに対応して脅威を割り付ける（表 2）． このとき，ECP を STRIDE 手法で詳細化する．表２では通信データ改ざんは，正常な場合と改ざんに

よる通信エラーデータが通信路上に流され，機能安全要求の通信エラー監視機能では対応できないため，空欄となっている．

表２　安全機能と脅威

機能安全要件定義		セキュリティ要件定義	
機能安全要求	安全機能	STIRDE による詳細化	ECP
通信２のエラーが多い場合には，操舵操作から出す重さ信号を最大に設定する．	通信状態を直接監視機能に入力・通信路異常を監視し，異常ならば重さ信号最大化	通信路上で，中間者によるデータ破壊	破壊
-	-	不正機器による偽データまたはデータ改ざん	改ざん

空欄部に対する機能安全要求を導出するため，FMEA でハザードを導出し，HAZOP を行う．ここではデータの改ざんによりハンドルが異常に軽くなり操作を誤ることをハザード事象とする．この場合，対策機能は正しい機器が発信したこと（発信元改ざん）及びそれが改ざんされていないこと（通信路改ざん）を保証する対策が必要となる．ここではこの２つを確認できるメッセージ認証機能（MAC 認証）を用いて対策機能とすることを想定する（表３）．

表３　機能セキュリティコンセプトと脅威（FMEA 後）

機能安全要件定義		セキュリティ要件定義	
機能安全要求	安全機能・機能セキュリティコンセプト	STIRDE による詳細化	ECP
通信２のエラーが多い場合には，操舵操作から出す重さ信号を最大に設定する．	通信状態を直接監視機能に入力・通信路異常を監視し，異常時は重さ信号最大化	通信路上で，中間者によるデータ破壊	通信データ破壊
正規と区別できない⇒メッセージ改ざん検出機能で検出したら重さ信号最大	MAC 認証（HMAC による MAC・鍵とシリアル値同期）	不正機器による偽データまたはデータ改ざん	通信データ改ざん

6.4　リスク評価と対応，技術セキュリティ要求導出（図 5.C.5,6）

前節で導出されたセキュリティに対する機能要求を元に，実装要件である技術セキュリティ要求の導出を行う．技術安全要求仕様相当段階では，機能安全コンセプトに対して，その実装要求（技術安全要求）が導出される．システム設計情報などを利用してセキュリティ脅威を更に詳細化し，これらの脅威に対して，JASO TP15002 に従い CRSS によるスコアリングを実施する（表 4）．

表４　詳細脅威とリスクスコアリング例

脅威 ID	対象に対する脅威	STRIDE による詳細化	スコア（判定）
T1	通信路上の設置デバイスで電気的に破壊する	通信路上データの破壊	8（要）
T2	中継デバイス（GW など）を改ざん / 破壊する		2（対処不用）
T3	送信側デバイスのアプリケーションを改ざん，不正メッセージを送出	システム改ざんによるデータ改ざん	2（対処不用））
T4	送信側デバイスを不正に入れ替え，不正メッセージ送出		6（要）
T5	通信路情で設置したデバイスでメッセージ入れ替え	データ改ざん	6（要）

この結果に対して，先に機能安全のハザード分析時に得られている評価指標（図 5 L.0）と比較し，スコアリング結果から対応すべき脅威を選別し，セキュリティ機能を検出予防と検出後措置の構成で設計する．表５では，表４で示された脅威に対する対処策を示している．ここでは，対処必要とされた脅威に対する対応策の導出手順を述べる．まず，T1 では，脅威の結果，現れるエラー通信の増加を検知指標とする．エラー通信数がある境界値を越えれば（検出１），表３で定めた検出後措置である，重さ信号送出（対処１）を実施する．次に T4,T5 では，不正デバイスからの送信検出に MAC 認証を採用する（検出２）．脅威が検出された場合，表３の対処方法である重さ信号送出（対処２）を実施する．

表５　技術安全要求と技術セキュリティ要求

技術安全要求と技術セキュリティ要求		セキュリティ ID	セキュリティ 対応
単位時間当たりの通信のエラー通信数を取得することで通信エラーを判定し，閾値を超えたときにハンドル重さを最大とする．	（検出 1）通信２の状態を直接監視機能に入力・エラー通信数をカウント．境界値より多ければ（対処 1）正規出力を停止し，監視機能から重さ信号送出	T1	要
		T2	非
デバイス側で対処		T3	非
MAC 認証を通らないメッセージを検出し，閾値を超えたら重さ信号最大とする	（検出 2）送信デバイスから送られてきたメッセージの MAC 値を検証し，（対処 2）不正データがあれば，対処 1 を実施	T4	要
		T5	要

これらの対策を図 4 のモデル図にマッピングしたものが図 9 である．セキュリティ脅威は予防検知の部分で検出される．機能安全設計で対処可能なセキュリティ脅威は，そのまま安全機能で対処される．安全機能で対処できないセキュリティ脅威は新たに実装されるセキュリティ機能（処理）で対処する．

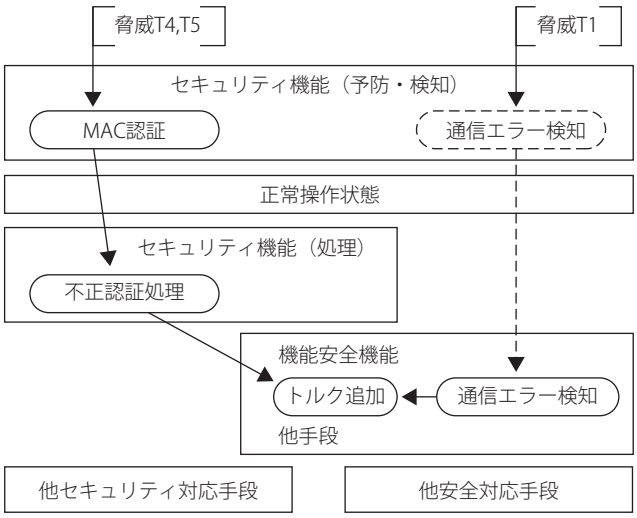


図９　セキュリティ対応対象システム

最終的に，セキュリティ攻撃の発生時に，機能安全要件定義で定義された安全状態への移行を設計に組み込み，発生が懸念されるハザードに対処する．

7　考察

本提案では，機能安全とセキュリティの開発を統合するため，機能安全での要件定義対象であるアイテムに ECP を導入し，機能安全と同じ対象でのセキュリティ要件定義を可能にした．また，脅威及び，セキュリティ機能の導出は，ISO 26262 で定義されている帰納的・演繹的手法を用いた対処抜け防止手順を流用し，セキュリティ対応手法の導入を最小限にとどめ，新たに発生するコストを抑えている．

本提案のリスク評価は，STRIDE 手法及び CRSS/RSMA を利用する．STRIDE は情報システムの分析では標準的な手法の一つであり，今回の分析も情報の動きに着目した分析を行うため，適合性が高いと考えられる．また，CRSS は情報処理システムの実績ある脆弱性評価方法である CVSS を，RSMA は情報セキュリティ管理とリスク管理プロセスにかかる作業を規格化したガイドラインで ISO/IEC 27005[14] での攻撃容易性と被害の概念を利用したリスクレベル決定表を用いており一般性があると考えられる．しかし，これらについては，結果の網羅性と実用性担保の観点から，更に検討が必要と考える．

次に，本研究と先行研究 [9] の比較を行う．本研究では，ECP から予想される被害からリスクスコアリングの基準を導出，システム設計段階で脅威の詳細度を高め，更にシステム設計情報と併せてリスクスコアリングを行う．先の基準に従い，対策実施するものと，残存脅威とするものの２つに分類する．これにより，対処しない詳細度脅威を管理できる一方，詳細度の高い脅威を対象として検討するため検討内容の詳細度が高く，対策の有効性や漏れの検証に有利である．更に，ステアリングシステムに対する適用結果を比較する．先行研究 [9] のセキュリティ対策は Trustzone の境界面の HSI に対する，ハザード分析段階でのリスク評価に基づきキーワードレベルの対策を定める．例を表６に示す．

表６　先行研究での対策例

信号名	ASIL	リスク／対策キーワード
車両速度シグナル	B/2	2／異常挙動検出，侵入検知，デバイス認証

表５と比較すると，同等ではあるが記述内容は本研究がより詳細である．また，本研究では，セキュリティ攻撃発生時の安全状態移行も対策化（図９）している．そのため，本提案の要件分析は，脅威発生時の安全動作も含む，より安全性の高いものである．

8　まとめ

本論文では，安全機能とセキュリティ機能の要件定義の手法を検討，提案した．本論文が，機能安全とセキュリティの統合に関連する議論の一助となれば幸いである．また，今後は実プロジェクトへの適用などを通じて有効性の検証と手法の洗練化を図りたい．

【参考文献】

[1] ISO 26262：2011. Road vehicles — Functional Safety —
[2] https://ics-cert.us-cert.gov/Year-Review-2013
[3] https://www.isa.org/isa99/
[4] SAE J3061：2016. Surface Vehicle Recommended Practice
[5] https://www.evita-project.com
[6] M.Steiner, et. al., Combination of Safety and Security Analysis - Finding Security Problems That Threaten The Safety of a System. In SAFECOMP 2013
[7] D.Ward, et. al., Threat Analysis and Risk Assessment in Automotive Cyber Security, SAE Int. J. Passeng. Cars – Electron. Electr. Syst. 6（2）:507–513, 2013
[8] K.Schmidt, et.al., Adapted Development Process for Security in Networked Automotive Systems," SAE Int. J. Passeng. Cars – Electron. Electr. Syst. 7（2）:516–526, 2014
[9] G. Macher, et.al., Integrated Safety and Security Development in the Automotive Domain, SAE Technical Paper 2017-01-1661, 2017
[10] G. Macher, et.al., SAHARA: a Security-Aware Hazard and Risk Analysis Method, in DATE, 2014
[11] 公益社団法人自動車技術会 :JASO テクニカルペーパー 自動車 - 情報セキュリティ分析ガイド , JASO TP15002, 2015.
[12] R. Baskerville, Information systems security design methods: implications for information systems development, ACM Computing Surveys, vol.25 no.4, pp.375-414, Dec. 1993
[13] A.Shostack, threat modeling. Wiley, 2014
[14] ISO/IEC 27005:2011 Information technology — Security techniques — Information security risk management

日本のソフトウェア技術者の生産性と労働条件の決まり方：
アメリカ，中国，フランス，ドイツとの比較を交えて

中田 喜文※

日本のソフトウェア技術者の生産性と労働条件の現状について主要 4 カ国と比較した同志社 (2016) によると，日本のソフトウェア技術者の自己評価生産性は低く，時給と労働条件で見た労働条件も 5 カ国中最も低いものであった．このような状況をもたらした原因の解明のため，生産性と労働条件決定のメカニズムを理論モデルとして提示し，そのモデルの妥当性の検証を，同志社 (2016) で用いたものと同じデータによって試みた．結果は，モデルの主たる説明要因である個人の能力，職務，職場環境，及び業務のマネジメントのありようの差異で，生産性については日本とほかの国の差異の大半が説明できた．この結果から，今後の日本の政策として，ソフトウェア技術者の能力の向上，職務の配置の適正化，職場環境の改善，そしてマネジメントの改善が生産性の向上には効果的であることが示唆された．

Determination of Software engineers' productivity and their working conditions:
with reference to those in US, China, France and Germany

Yoshifumi, Nakata※

In this paper we analyzed the determination of the perceived productivity and two working condition variables, hourly pay and working hour, of Japanese software engineers in comparison to those in US, China, France and Germany. We applied what we call the four factor model for this analysis. It has been shown that this model is quite effective not only in Japan but also in other four countries in explaining the productivity variation. We then proposed some policy proposals toward the higher productivity for the Japanese software engineers based on our findings.

1　はじめに

1.1　問題意識

日本のソフトウェア技術者の生産性と労働条件の現状について主要 4 カ国と比較した同志社大学 (2016) によると，日本のソフトウェア技術者は，自身の生産性に対する評価が低く，時給と労働時間で見た労働条件も 5 カ国中最も低いものであった．このような日本のソフトウェア技術者の状況はなぜ発生したのだろうか．今日までの日本の成長の源泉であったものづくりを先導したハードウェア技術者や

※　同志社大学　総合政策科学研究科

生産現場の労働者たちの高い生産性とコントラストを示す現状は，一義的には当事者にとって喫緊の課題であり，同時に彼らが生み出すソフトウェアが現代社会の基本インフラであることを考えると，社会のすべての人々にとっても，可及的速やかな効果的対策を必要とする最重要な社会経済的課題である．

本論文では，この課題に対するアプローチとして，まず生産性及び労働条件の決定モデルを先行研究の成果に基づき提示する．続いて，この決定モデルが，日本のソフトウェア技術者の生産性の低さの実態をどの程度説明できるかを，同志社大学 (2016) が用いたものと同じデータによって検証する．限定されたデータではあるが，モデルに説明力があることが確認できれば，今後の課題解決に向けての政策的対応のオプションに対して，一つの指針が得られるからである．

続いて，このモデルが，海外 4 カ国のソフトウェア技術者の生産性と労働条件についても，日本と同様に説明可能なモデルであるかを検討する．もし，説明可能であれば，ソフトウェア技術者に関して，当該モデルの一般性が示唆され，このモデルに基づく政策対応をこれらの国々に対しても提案できることになる．

1.2 本論文の構成

上記課題に答えるために，本論文の構成を以下のように設定する．

まず，「2．モデルの構築と検証方法」では，ソフトウェア技術者の生産性及び労働条件の決定メカニズムに関連する先行研究を検討し，更には，多数の国内外のソフトウェア技術者への聞き取り調査の結果も踏まえて作成された 4 要因モデルの説明を行う．続いて，このモデルの検証のために用いるデータを提示し，そのデータの特徴を述べる．最後に実証のための統計的方法を解説する．

続く「3．日本のソフトウェア技術者の生産性と労働条件の決まり方」では，順にソフトウェア技術者の自己評価生産性，職務満足度，時間当たり給与，及び労働時間について 2．で提示したデータを用いてモデルの統計的適合性を検証する．「4．米，中，独，仏 4 カ国のソフトウェア技術者の生産性と労働条件の決まり方」では，3．で用いた生産性と労働条件決定モデルが，米，中，独，仏 4 カ国のソフトウェア技術者の生産性と労働条件の実態をどの程度説明できるかを，これも同志社大学 (2016) が用いたものと同じデータで検証する．「5．まとめと示唆」では，それまでの日本及び海外 4 カ国のソフトウェア技術者の生産性と労働条件の決まり方に関する発見を基に，日本のソフトウェア技術者の生産性改善のための施策を検討し，具体的な提言を行う．

2 モデルの構築と検証方法

2.1 ソフトウェア技術者の生産性と労働条件の決定モデル

同志社大学 (2016) では，ソフトウェア技術者の生産性指標として，職務を通して生み出す価値の自己評価指標 (自己評価生産性) と職務満足度が採用されている．その根拠としては，「技術者個々人から収集が可能なデータで，かつ国を超えての比較が行える」ことと，前者については，その指標が，「生産活動におけるインプット，その結果としてのアウトプット，更にはそのアウトプットの価値の 3 点についての自己評価の統合指標」であること，後者については，「職務満足自体が技術者にとっての価値 (生産活動が従事者に与える価値) であることに加え，客観的生産性と高い相関性を持つことも先行研究で明らかにされている」ことがあげられている．

そこで，労働条件決定に関する先行研究の成果も踏まえソフトウェア技術者の生産性と労働条件（時給、労働時間）に影響する要因をモデル化する必要がある．近年の組織心理学と組織社会学の成果，そして労働経済学を総括して導出した※1．ここでは，主たる影響を与える要因が 4 つに集約されることから，「生産性・労働条件の 4 要因モデル」と呼ぶことにする．

このモデルは，ソフトウェア技術者の生産性と処遇など

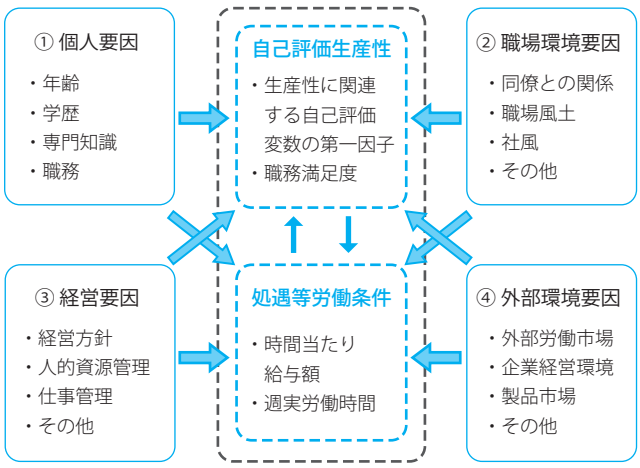


図 1　生産性・労働条件の 4 要因モデル

※1　福谷 (2007)，中田・電機総研 (2009)，古田 (2017) 等を参照．

労働条件が 4 つの要因に影響を受けることを示している．要因を以下に説明する．なお要因は“要素”で構成される．

① 個人要因：

技術者の年齢，学歴，専門知識，あるいは実際に担当する職務の内容などである．（以下ではこれらを，要因を構成する要素と呼ぶ．）これらの個別要素が，労働生産性に対して影響を与えることは，多くの先行研究で証明されている※2．

② 職場環境要因：

協力的な同僚と共に働けるか，自由闊達な議論のできる職場の雰囲気か，新たな挑戦が評価される職場であるか，などの要素を考える．

③ 経営要因：

どのような経営方針で会社が運営され，また，その方針が社員に理解され，共感されているか．あるいはどのように社員の働きが評価され，処遇されているか．また，日々の仕事がどのように管理され，一人一人の就業時間が労働効率的に決められているかなどである．つまりより良い経営が生産性を高める関係を表している．

④ 外部環境要因：

最後の要因は，会社を取り巻く経済社会環境である．労働市場が発達し，必要な人材は容易に市場から調達できるか．会社が生産する製品やサービスの市場が競争的かそれとも少数の企業によって，独占されているか．更には企業経営に対し，株主は様々な形で希望する経営のあり方について，意思表示を行い，実質的にも議決権を行使し，介入しようとするか，などである．

このモデルがどの程度現実の日本のソフトウェア技術者の生産性や時給，労働時間などの労働条件の実態を説明できるのか．その検証は，このモデルに基づく回帰分析モデルを作成し，日本のソフトウェア技術者について収集した，上記 4 要因の様々な要素に関するデータと生産性及び労働条件データとの関係をどれ程うまく説明できるかを見ることで評価する．

日本の技術者データとしては，同志社大学 (2016) が用いた電機連合データを使用する．データ取集時期とその収集方法は以下の通りである（表 1）．

3 日本のソフトウェア技術者の生産性と労働条件の決まり方

3.1 日本のソフトウェア技術者の生産性と労働条件の決まり方

電機連合が収集した日本のソフトウェア技術者データを用いて 2．で説明した 4 要因モデルの現実説明力を，重回帰分析で検証した．表 2 にその結果を示した．左の列には，4 要因の各要因に含まれる要素を列記した．これらが回帰分析における説明変数である．4 列目からは，2 つの生産性関連指標と労働条件指標である時給と週労働時間を被説明変数とする回帰分析の結果を示した．なお，生産性関連指標は，同志社 (2016) に準拠した．各欄の◎，あるいは○は，各列の被説明変数に対する各行の説明変数が，統計的に有意な係数を持つかを示している．◎は 5% の統計的有意水準で係数が推定された場合，○は対応する欄の係数が 10% の統計的有意水準であることを示す．

3.2 生産性指標

生産性指標に関して，多くの要素が 2 つの生産性指標に影響を持つことが確認できる．被説明変数全体のばらつきに対して，説明変数全体の説明割合を示す R2 乗値は，ほぼ 0.5 であった．ばらつきの約半分がこれらの説明変数によって説明可能であることが分かる．また，2 つの生産性指標の説明要因については，以下の 3 点が指摘できる．

- 職務満足度は，性，年齢などの個人属性要因の影響を受けない

表 1　電機連合ソフトウェア技術者データ

データ収集実施時期と収集方法	回収サンプル数				協力／委託機関
	回収数（注）	ERP ソフト技術者	組込みソフト技術者	その他のソフト技術者	
2015 年 12 月～ 2016 年 1 月 日本国内の電機連合組合員及び管理者に対し調査票（紙）で実施	3115	364	493	164	電機連合（全日本電機・電子・情報関連産業労働組合連合会）

資料出所:同志社大学(2016)
(注) 回収サンプル数には,ソフトウェア技術者以外の技術者 2094 人を含む.

※2　労働生産性一般については，Lazear (2000)やBlack and Lynch(2005)，ソフトウェア技術者については，Trendowicz and Münch(2009)を参照.

表２　日本のソフトウェア技術者の生産性と労働条件の決まり方

(注) 表の○, 及び◎は, それぞれの回帰分析結果において, 対応する要因に含まれる説明変数の係数が統計的に有意であることを示す. なお, ◎は5%, ○は10%の有意水準の場合を示す.

	回帰分析被説明変数：		生産性		処遇など労働条件	
			職務満足度	自己評価生産性	時給	週労働時間
個人要因	個人属性	性別ダミー			○	◎
		年齢		◎	◎	○
		年齢の２乗		◎	◎	◎
		勤続年数			◎	
	学歴・能力	学歴			◎	
		能力限界感	◎	◎		
		今後 10 年，自分の技術に関する自信	◎	◎	◎	
		専門職力	○	◎	◎	
		組織人力	◎	◎	○	
		マネジメント力	◎	◎	◎	
	職務・キャリア	職務とのマッチング	◎	◎		
		管理職務			◎	◎
		専門職として今の会社にとどまりたい	◎	◎	○	
		昇進より好きな仕事をしたい				
		専門を生かして転職したい	◎	○		
職場環境要因	職場環境	愛社心	◎	◎		
		オープンで進取の職場	◎	◎		
経営要因	経営管理	経営意思の社内での共有と共感	◎			
		劣悪な仕事管理	◎	◎	○	○
		時間管理の弱さ	◎			
		能力開発に積極的	◎			
		良好な評価処遇制度の運用	◎	◎		
外部環境要因	外部市場（労働・商品）	職種＝組込みソフトウェア技術者				
		職種＝その他ソフト				
		業種＝家電	○			◎
		業種＝通信				◎
		業種＝情報	◎	○		◎
		業種＝部品	○		◎	◎
		業種＝その他			○	○
		R2 乗	0.537	0.499	0.588	0.085
		修正済 R2 乗	0.526	0.488	0.578	0.062
		サンプル数	1274	1274	1172	1210

- 共通に影響を与える要素は，能力，職務マッチング，専門職志向，職場環境，そして仕事と人的資源管理である
- 自己評価生産性は，職務満足と比較して外部市場要因から受ける影響は限定的である

3.3 労働条件

労働条件に関しては，生産性指標と比較して，統計的に関連する要素数が大幅に減少する．時給は，関連する要素が個人属性と学歴・能力に限定される．他方，週労働時間では，個人属性と外部環境要因，とりわけ商品市場の影響が強い．また，時給と週労働時間を比べると，モデルの説明力に大きな違いがある．時給の場合は，全体のばらつきの半分以上がこのモデルで説明されるのに対し，週労働時間については，モデルの説明力は，１割にも満たない．

4 米, 中, 独, 仏４カ国のソフトウェア技術者の生産性と労働条件の決まり方

3. で見た４要因モデルの世界の中での適合性を検討する．ここでは，米，中，独，仏４カ国と較べる．

4.1 生産性指標

表３は，自己評価生産性指標について，表４は，職務満足度に関する回帰分析結果である．自己評価生産性に関する回帰分析の説明力は，どの国でもほぼ半分程度で，大きな差異はない．統計的有意性の高い変数（要素）の数では，日本は，５カ国中最も多い．また，日本とアメリカでは，統計的有意性の高い変数（要素）が類似する．日本については統計的有意な変数は 15 個あるが，アメリカの場合それら 15 個の内９個が統計的に有意である．しかし，中国では５個，ドイツは４個，そしてフランスでは５個のみが

有意である．具体的には，個人属性では年齢変数，学歴・能力では専門職力などの３個の個人能力変数，職務キャリアでは，職務とのマッチングの良さ，職場環境では，２個の職場環境変数，そして経営管理変数では，仕事管理の悪さ変数が，日米両国で統計的に有意である．これら９個の日米共通の，統計的に有意な変数の中で，他の３カ国と異なる変数は，年齢変数と３個の能力変数である．年齢（経験年数）も個人間の経験の過多に基づく能力差を表すと考えると，日米のソフトウェア技術者に特徴的な，生産性評価に影響を与える変数は，個人の能力変数である．

4.2 職務満足度

表４に職務満足度の分析結果を示した．第一の特徴は，５カ国共通にモデルの説明力が高いことである．この５カ国の中で説明力が最も低い日本でも５割を超え，最も高いフランスでは，７割ほどがこのモデルで説明できている．第二の特徴は，日本の統計的有意係数の多さである．日本において説明変数の有意性が確認できなかったのは，個人属性変数だけである．

4.3 時給

時給の結果は表５に示した．ここでもモデルの説明力の高さで日本が際立つ．修正済 R2 乗値では，日本技術者の時給ばらつきの６割弱がモデルで説明できている．しかし，ほかの４カ国では説明力は，２割から４割である．また，統計的に有意な変数の数でも，日本はフランスと並んで最多である．日本のもう一つの特徴は，統計的に有意な変数が個人要因に偏在することである．具体的には，個人属性と学歴・能力要素においては，10 要素中９要素が統計的有意性を示す．残りの４カ国については，国による差異が大きい．例えばフランスは，時給のばらつきの半分程度はモデルで説明できるのに対し，アメリカやドイツは２割程度

表３　自己評価生産性：モデルの国際的適合性に関する検証

			アメリカ	中国	ドイツ	フランス	日本
個人要因	個人属性	性別ダミー					
		年齢	◎	◎			◎
		年齢の２乗	◎				◎
		勤続年数				◎	
	学歴・能力	学歴					
		能力限界感		◎			◎
		今後 10 年，自分の技術に関する自信			◎		◎
		専門職力	◎	◎			◎
		組織人力	◎				◎
		マネジメント力	◎				◎
	職務・キャリア	職務とのマッチング	◎	○	◎	◎	◎
		管理職務					
		専門職として今の会社にとどまりたい				◎	◎
		昇進より好きな仕事をしたい	◎	◎		○	
		専門を生かして転職したい					○
職場環境要因	職場環境	愛社心	○				◎
		オープンで進取の職場	◎		◎	◎	◎
経営要因	経営管理	経営意思の社内での共有と共感	◎				
		劣悪な仕事管理	◎			◎	◎
		時間管理の弱さ				○	
		能力開発に積極的		◎		○	
		良好な評価処遇制度の運用		◎	◎		◎
外部環境要因	外部市場（労働・商品）	職種＝組込みソフトウェア技術者					
		職種＝その他ソフト				○	
		業種＝家電					
		業種＝通信					
		業種＝情報					○
		業種＝部品					
		業種＝その他					
		R2 乗	0.506	0.447	0.461	0.519	0.499
		調整済み R2 乗	0.467	0.365	0.406	0.441	0.488
		N	399	219	314	210	1274

注）◎は5%の統計的有意水準で係数が推定された場合，○は対応する欄の係数が 10%の統計的有意水準であることを示す．　表４以下においても同様．

である．とくにアメリカは，統計的に有意は変数が 5 つしかなく，このモデルでは時給のばらつきが上手く説明できていない．このアメリカにおけるモデルの適合性の低さについては，一つには時給計算に用いた年間総労働時間数データの特性に起因すると思われる．アメリカのソフトウェア技術者については，労働時間データを，幅のある労働時間帯の中から選択する形で収集した．このことが，労働時間数に関する推計の不確かさを高め，ひいては時給データの誤差を大きくしたと思われる．

以上の結果より 4 要因モデルの時給説明力は低いと結論できる．

4.4 週労働時間

表 6 が，週労働時間の結果である．どの国においてもモデルの説明力は低い．また，29 の説明要素の中で，5 カ国共通にその統計的有意性が確認できるのは，管理職務変数のみである．以上の結果から 4 要因モデルは，ソフトウェア技術者の労働時間の個人間差異を説明するモデルとしては不適切であると結論する．

5 考察

5.1 決定モデル

以上の日本のソフトウェア技術者の生産性に関連する 2 つの指標（自己評価生産性指標と職務満足度）及び時給と週当たり労働時間の決まり方に関する分析結果の要点は，以下の通りである．

- 生産性に関連する 2 つの指標については，両指標の各国内での分散の約 4 割から 7 割を 4 要因モデルが説明できた．かつ個人要因，職場環境要因，経営要因が，5 カ国すべてで統計的に有意であり，外部環境要因についても 5 カ国中 3 カ国で統計的有意性が確認できた．

表 4　職務満足度：モデルの国際的適合性に関する検証

	回帰分析被説明変数：職務満足度		アメリカ	中国	ドイツ	フランス	日本
個人要因	個人属性	性別ダミー					
		年齢	◎	○	◎		
		年齢の 2 乗	◎		◎		
		勤続年数					
	学歴・能力	学歴					
		能力限界感				◎	◎
		今後 10 年，自分の技術に関する自信			◎	○	◎
		専門職力	◎				○
		組織人力	◎				◎
		マネジメント力	○				◎
	職務・キャリア	職務とのマッチング	◎	◎	◎	◎	◎
		管理職務		○	◎		
		専門職として今の会社にとどまりたい		○	◎	◎	◎
		昇進より好きな仕事をしたい	◎			◎	
		専門を生かして転職したい		◎		○	◎
職場環境要因	職場環境	愛社心		◎			◎
		オープンで進取の職場	◎	○	◎	◎	◎
経営要因	経営管理	経営意思の社内での共有と共感	◎	◎		◎	◎
		劣悪な仕事管理	○		◎	◎	◎
		時間管理の弱さ					◎
		能力開発に積極的		◎		○	◎
		良好な評価処遇制度の運用	◎	◎	◎	◎	◎
外部環境要因	外部市場（労働・商品）	職種 = 組込みソフトウェア技術者					
		職種 = その他ソフト					
		業種 = 家電					○
		業種 = 通信					
		業種 = 情報					◎
		業種 = 部品			◎		○
		業種 = その他					
		R2 乗	0.590	0.577	0.609	0.732	0.537
		調整済み R2 乗	0.557	0.515	0.570	0.689	0.526
		N	396	218	315	208	1274

表５　時給：モデルの国際的適合性に関する検証

	回帰分析被説明変数：時給		アメリカ	中国	ドイツ	フランス	日本
個人要因	個人属性	性別ダミー				○	○
		年齢（満，10 月 1 日現在）	◎	◎	◎	◎	◎
		年齢の 2 乗	◎		◎	◎	◎
		f6 勤続年数		◎			◎
	学歴・能力	教育		◎	◎	◎	◎
		能力限界感					
		今後 10 年，自分の技術に関する自信	○				◎
		専門職力		◎		◎	◎
		人間力					○
		マネジメント能力					◎
	職務・キャリア	職務とのマッチング	◎	◎			
		管理職務			◎		◎
		専門職として今の会社にとどまりたい				◎	○
		昇進より好きな仕事				○	
		専門を生かして転職する				◎	
職場環境要因	職場環境	愛社心					
		オープンで進取の職場		◎	○	○	
経営要因	経営管理	経営意思の伝達			◎		
		悪い仕事管理				◎	○
		忙しくて学びの時間なし		◎			
		能力開発に消極的				◎	
		良好な評価処遇制度の運用	○				
外部環境要因	外部市場（労働・商品）	職種＝組込みソフトウェア技術者		○			
		職種＝その他ソフト			◎		
		業種＝家電				◎	
		業種＝通信					
		業種＝情報				◎	
		業種＝部品					◎
		業種＝その他				○	○
		R2 乗	0.271	0.392	0.28	0.505	0.588
		調整済み R2 乗	0.213	0.295	0.193	0.406	0.578
		N	393	205	269	176	1172

以上の結果から，本論文で提示した 4 要因モデルが，日本のみならず，比較対象であるアメリカ，中国，フランスそしてドイツのソフトウェア技術者の生産性関連指標の実態についても，相当程度の説明力を持つことが確認された．

- 日，仏両国については，生産性指標が，会社組織を取り巻く外部環境である労働市場や製品・サービス市場の状況の差異を体現する業種変数によっても影響を受けていることが確認できた．
- 労働条件については，時間給と週労働時間共に，4 要因モデルの説明力は一部の対象国においてある程度高いもののその説明力に貢献する要因変数も限定された．このことから，4 要因モデルは，ソフトウェア技術者の労働条件の説明モデルとしては不適切である．時給，労働時間などの労働条件の決定については，異なるモデルと仮説に基づき，あらためての検討が必要である．

5.2　政策的示唆

以上の発見を踏まえると，日本のソフトウェア技術者の生産性が，世界の中での極めて低い位置にあることについては，今後の政策対応に関して以下の示唆が得られる．

- 改善策の選定においては，この 4 要因の中の統計的有意性が高いと評価された要素の中から行うことで，政策的な効果が期待できる．具体的には，ソフトウェア技術者に対して専門職力，組織人力，そして経営管理力の能力 3 要素の底上げに資する施策が示唆される．
- 企業においては，職場環境と経営管理のあり方の改善に取り組む必要がある．最も最先端のイノベーション現場であるべきソフトウェア職場が，保守的でリスク回避的な空気に覆われていては，創造的な企業活動ができにくい．経営の最重要なリソースである人のマネジメント，そしてタイムマネジメントのあり方とその効率的な実施において，改善の余地が大きい．

表６　週労働時間：モデルの国際的適合性に関する検証

	回帰分析被説明変数：週労働時間		アメリカ	中国	ドイツ	フランス	日本
個人要因	個人属性	性別ダミー			○	◎	◎
		年齢（満，10 月 1 日現在）					○
		年齢の 2 乗					◎
		f6 勤続年数					
	学歴・能力	教育					
		能力限界感			○		
		今後 10 年，自分の技術に関する自信					
		専門職力		◎			
		人間力					
		マネジメント能力					
	職務・キャリア	職務とのマッチング		◎		○	
		管理職務	◎	○	◎	◎	◎
		専門職として今の会社にとどまりたい	◎			◎	
		昇進より好きな仕事			◎		
		専門を生かして転職する			◎		
職場環境要因	職場環境	愛社心					
		オープンで進取の職場				○	
経営要因	経営管理	経営意思の伝達		◎			
		悪い仕事管理					○
		忙しくて学びの時間なし				◎	
		能力開発に消極的	◎				
		良好な評価処遇制度の運用	◎				
外部環境要因	外部市場（労働・商品）	職種＝組込みソフトウェア技術者		○		◎	
		職種＝その他ソフト				◎	
		業種＝家電				◎	◎
		業種＝通信				◎	◎
		業種＝情報				◎	◎
		業種＝部品				◎	◎
		業種＝その他				◎	○
		R2 乗	0.133	0.237	0.195	0.453	0.085
		調整済み R2 乗	0.065	0.121	0.111	0.356	0.062
		N	400	213	306	194	1210

・労働条件の改善にむけては，労働条件を説明するモデルとして，4 要因モデルの有効性が低い結果から明確な示唆の抽出は困難である．しかし，日本においては，労働条件に対しては，産業，業種の影響が強いことが確認されたことから，産業，業界単位での給与水準の引き上げや，超勤の削減などの労働条件改善の取り組みが有効と思われる．

謝辞

本論文は，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（SEC：Software Reliability Enhancement Center）が実施した「2014 年度ソフトウェア工学分野の先導的研究支援事業」の支援を受けたものである．また，本論文の改訂については，2 名の査読者から多くの有益な助言を受けたことを記す．

【参考文献】

同志社大学「日本のソフトウェア技術者の生産性及び処遇の向上効果研究：アジア，欧米諸国との国際比較分析のフレームワークを用いて」，『2014 年度ソフトウェア工学分野の先導的研究支援事業成果報告書』，2016 年.

中田喜文・電機総研　『高付加価値エンジニアが育つ』，日本評論社，2009 年.

古田克利　『IT 技術者の能力限界の研究』，日本評論社，2017 年.

福谷正信　『研究開発者技術者の人事管理』，中央経済社，2007 年.

Black, Sandra and Lisa M. Lynch, “How to compete:The Impact of Workplace Practices and Information Technology on Productivity”, Review of Economics and Statistics, Vol.83, no. 3, pp.434-445, 2005.

Lazear, Edward,” Performance Pay and Productivity”, American Economic Review, Vol. 90, no. 5, pp. 1346 - 1361, 2000.

Trendowicz, Adam and Jürgen Münch, “Factors Influencing Software Development Productivity - State of the Art and Industrial Experiences”, Advances of Computers, Vol.97, pp.185-241, 2009.

企業固有スキルのレベルを判断するための社内独自試験実施の取り組みとその効果

清谷 佳史※1

ITプロフェッショナルの知識保有レベルを判断するにあたって，情報処理技術者試験などの信頼性の高い団体試験を活用することは，有効な手段である．しかしながら，ビジネス遂行において，他社との差別化を図るために必要な企業固有スキルの分野については，そのような試験が存在しないことが多い．そこで，弊社は「社内独自試験」を作成し，実施した．実施するにあたって，弊社では試験問題作成の経験がほとんどなかったため，有効性のある試験が実施できるか懸念されたが，十分に計画を練って，思慮深く作業を進めることにより，問題なく試験を実施することができた．また，その有効性についても確認できたので，ここに事例として紹介すると共に，この経験を基にして立案した社内独自試験の実施方法を提案する．

Implementation and Effectiveness of In-house Test to Measure Skill Level Specific to Company's Employees.

Yoshifumi Kiyotani※1

It is an effective means to utilize highly reliable group tests such as information processing engineers examination as a means for judging the knowledge holding level of IT professionals. However, in the field of company-specific skills to differentiate from other companies, such tests often do not exist. Therefore, we created and carried out "in-house unique test". We had no experience of making tests. For that reason, We were worried whether effective tests could be carried out. However, we could do it without problems by thoroughly planning it and proceeding with thoughtful work. And since we were able to confirm the effectiveness, we will introduce it as a case. In addition, we propose an in-house unique test implementation method based on this experience.

1　はじめに

IT企業におけるITプロフェッショナルの育成において，従来よりも企業の経営目標と明確にマッチングした個人別育成目標の設定とスキル保有レベルの判断を客観的な評価で行うことの必要性が高まっている．

弊社においても，人材育成が個々の担当内に閉じた主観的な判断で行われたことに起因する担当者の技術力不足が発生した．この事実は，過去数年間に発生した赤字プロジェクトの原因分析を行う中で顕在化した．

そのため，より明確で客観的な指標をもとに人材育成を行うように変革することが全社指示事項として経営層から全社員に示され，全社的に人材育成の変革に取り組むようになった．

その取り組みの中心として，弊社の業務タスクと業務遂行のために必要な技術スキル（業務分野，システム基盤技術分野，開発技法，マネジメントなどのスキルカテゴリごとにスキルの習熟度を可視化できる最小単位まで技術スキルを詳細な項目に分類したもの）をスキルマップとして紐付けて明確な見える化を実施した．そして，担当者自身が業務遂行のために必要な技術スキルが現在どのレベルにあり，レベルアップが必要な技術

※1　NTTデータシステム技術株式会社

スキルが何かを把握し，目標設定する取り組みを行っている．この取り組みのイメージを図１に示す．

この取り組みは，独立行政法人情報処理推進機構（IPA）が推進している「i コンピテンシディクショナリ（iCD）」[IPA2015] の考え方に近いものである．弊社では，具体的なタスクディクショナリ，スキルディクショナリの詳細項目とは異なるカテゴリ分けでスキルマップを設定しているが，今後運用を改善していく中で，「i コンピテンシディクショナリ（iCD）」と近い考え方で運用していけるのではないかと考えている．

このように，IT プロフェッショナルの育成における方針や目標設定の考え方については，「i コンピテンシディクショナリ（iCD）」の考え方に準じることで明確化が可能である．

しかしながら，まだ重要な課題がある．それは，IT プロフェッショナルが獲得したスキル保有レベルの判断をどう客観的に行っていくかということである．

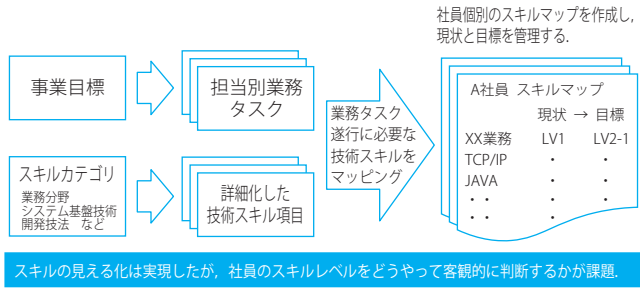


図１　スキルマップ・イメージ図

2　スキルレベルの判断について

IT プロフェッショナルのスキルレベルの判断においては，そのスキルレベルの明確な基準を設けて，その基準への達成度合いを判断することが基本となる．

例えば「i コンピテンシディクショナリ」のスキル熟達度判定基準を見ると，メソドロジのレベル３には，「課題に応じて手法の使い分けができる／現場にて手法を活用し結論を導いた事がある」とある．この場合は技術者の業務経験の内容を，上司あるいは有識者が面接などで，技術的な技量の発揮状況をヒアリングしてスキルレベルの判断を行うことになると考えられるが，どの程度の手法の活用を行ったのか，更に明確なガイドラインがないと誰が評価しても同一の評価となるような客観性を得ることは難しいのではないかと考えられる．

このようなスキルレベルの判断には，独立行政法人情報処理推進機構（IPA）による「社内プロフェッショナル認定の手引き」[IPA2009] が参考になる．筆者は，株式会社 NTT データがグループ会社内で行っている「プロフェッショナル CDP 制度」[Tanaka2010] にて，２００８年に「シニア IT スペシャリスト（ネットワーク）」に認定され，その後半年に一度の周期で，下位となるアソシエイトレベルの技術者を認定するための面接を実施している．この面接は，認定の公平性，妥当性を確保するために２名一組の面接員で行っている．実際，筆者が経験した範囲では，２名の面接員で認定可否の評価が分かれることはなかった．これは「プロフェッショナル CDP 制度」の認定基準が明確であるためである．

「プロフェッショナル CDP 制度」の具体的な認定基準について公表されている情報は少ないが，認定基準の考え方としては「社内プロフェッショナル認定の手引き」で書かれている内容に近いものである．

スキルレベルの判断をするには，IT プロフェッショナルとして必要な知識を保有していることと，それを活用して IT プロフェッショナルとしての技量を発揮した経験があることを併せて判断することが重要となる．一般的に「資格を持っていても仕事ができなければ意味がない」とよく言われるが，それは主に IT ヒューマンスキルが足りないために，IT プロフェッショナルとして自身の業務を遂行するときに十分な技量が発揮できていないためである．また，その逆で，面接による技量の発揮状況の確認だけでは，十分な知識を保有しているか確認するのは困難である．それは，限られた面接時間の中で，網羅的に知識の保有状況を確認することが難しいためである．

そのため，IT プロフェッショナルのスキルレベルの判断においては，試験の合格による知識の保有状況の確認と，面接による技量を発揮した経験の確認をセットで行うことにより，客観性の高い判断が可能になる．スキルレベル判断のイメージを図２に示す．

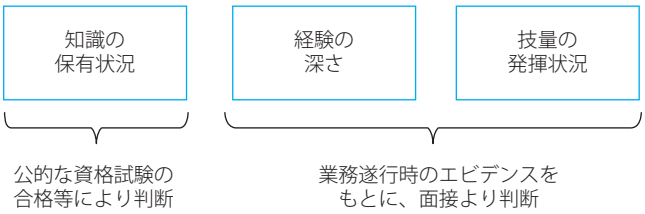


図２　スキルレベル判断のイメージ図

IT プロフェッショナルの知識の保有状況確認のためには，情報処理技術者試験などの信頼性の高い団体試験を活用することが有効である．しかしながら，このような試験が存在しない企業固有スキルの分野については，知識レベルをどう判断するかが課題になる．

3　企業固有スキルについて

企業固有スキルの位置付けについては，「i コンピテンシディクショナリ解説書」[IPA2015] にある「スキルディクショナリ構成図」に分かりやすく表現されている．この図で表されるスキル特性の中で，「メソドロジ」「テクノロジ」「関連知識」は情報処理技術者試験の知識項目や主要知識体系を参照元としている．そのため，これらのスキルは情報処理技術者試験や，主要知識体系の関連団体が主催する試験を受験することで，スキルの習得レベルを確認することができる．

スキルカテゴリごとに，知識の保有レベルの確認方法を示したイメージを図３に示す．

図３に示すような，試験が存在しない分野となる企業固有スキルとして，「IT システムの業務仕様に関するスキル」が該当する．

例えば弊社で開発，維持管理している IT システムの主なものは金融分野であり，勘定系システムにおける融資業務や，資金決済業務などがある．更に，情報系システムにおける各種統計業務など様々な業務を扱っている．こうした業務について，リスク管理，法令順守，経営戦略など，お客様のニーズに適したシステム開発を行うには，お客様と同等のレベルで仕様調整ができる業務知識を保有することが必要となる．

こうした個別のお客様システムに必要となる業務知識の保有レベルは，既存の試験で判断することは困難である．これは，金融分野のシステムに限らず，個別の業界のニーズに深く適合したITシステムを開発する場合においても同様である．こうした独自の業務知識は，IT企業における重要な企業固有スキルである．

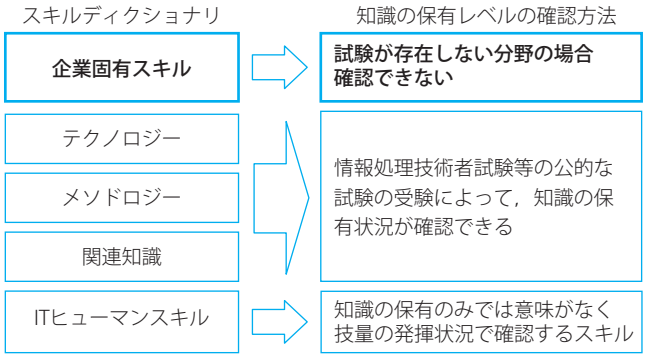


図３　スキルカテゴリごとの知識の確認イメージ図

4　社内独自試験作成の経緯

弊社は事業部制組織を採用しており，筆者は自身の所属する事業部（社員数２００名程度の組織）内で，「事業推進担当」という部署に所属している．「事業推進担当」は，システムの開発や維持管理を行う部署ではなく，収支管理や事業戦略，人材育成を企画する部署である．この中で筆者は，事業部内での社員のスキルを，スキルマップにより見える化し，更にスキルの習熟度を判断する仕組みを検討する中心的役割にあった．

事業を推進するために必要となる社員のスキル習熟度を判断する仕組みは，株式会社NTTデータの「プロフェッショナルCDP制度」を参考にして，「経験」「技量」「知識」を判断することにした．具体的には，業務遂行のために必要なスキルとして切り出した個々の詳細なスキル項目ごとに，「経験」「技量」については上司による面談で判断して，「知識」については，試験で判断することにした．

この場合に，「ネットワーク関連技術」や「データベース関連技術」など，システム基盤技術分野のスキルであれば，高度情報処理技術者試験の合格によって必要な知識の保有状況を判断することが可能である．しかしながら，弊社の「企業固有スキル」となる「所管するシステム固有の独自な業務知識」のスキル習熟度をどう測定するかが課題となった．

この課題について，関係者で議論を重ねた結果，「独自試験を作成するしか測定方法はない」という結論に達した．

しかしながら，筆者らの事業部では，本格的な試験問題の作成経験が皆無であり，事業部の社員数も比較的小規模であるため，筆者は円滑に試験を実施する運用の実現は不可能と考え，議論の中では最後まで社内独自試験の実施に反対した．しかし，検討した結論の妥当性に納得できたことと，議論の参加者全員の試験の作成，実施に対するモチベーションが高かったことから，社内独自試験を作成しその運用を成功させることにトライしようと考えるようになった．

5　試験問題の作成過程

筆者らに本格的な試験問題作成とその実施の経験がないことから，十分に計画を練って，その準備活動を行った．活動の中心となる試験問題作成の作業は，各担当から選抜された担当業務の有識者により結成されたワーキング・グループの活動で行うこととし，検討から試験実施まで約半年間で行う計画を立て，筆者が主催者となって活動を推進した．

ワーキング・グループによる活動は以下のような順序で行った．

5.1　試験に求められる条件の確認

ワーキング・グループのメンバ全員が試験を作成した経験がないことから，最初に筆者が一般的な試験に必要な条件を調べ，その内容をメンバに情報提供した．一般的に考えられている，試験に求められる絶対的な条件を表１に示す．

表１　試験に求められる条件

妥当性	その試験が本当に測定したいものを測定しているか
信頼性	その試験の結果は，状況にあまり左右されず，一貫した解答を期待できるか
公平性	同じ知識を持っているならば，同じくらいの点数が取れること．可能な限り，まぐれ当たりを避けること

5.2　出題形式の確認

一般的な試験の出題形式としては，「多岐選択式」と「論述式」があり，それぞれのメリット・デメリットを表２の内容に整理して，メンバに情報提供を行った．

5.3　試験のコンセプト及び出題レベルの意識合わせ

この試験の合格者となる人物像として「担当業務の第一人者となり後進の育成もできるレベル」の人材を設定した．これは，ＩＴスキル標準Ｖ３[IPA2012]における「レベル４」に相当する．より幅広いレベルを設定して，複数のレベルに対応した試験を作成してはとの意見もあったが，筆者らの事業規模では，複数のレベルに対応した試験を作成し，維持することは不可能であると判断した．

また，このように設定したレベルを測定する試験のコンセプトとして，「お客様と仕様調整ができるレベルの業務知識があることを合格レベルとする」と定義してワーキング・グループ内の意識合わせを行ったが，このコンセプトは比較的順調にメンバと意識を合わせることができた．メンバは自身の担当業務の中で，システム更改や改修時に何度もお客様と仕様調整を行い，システムの仕様を取り決めた経験のある，現場の第一人者だったので，出題レベルをイメージしやすかったのがその要因だと考えられる．

振り返ってみると，試験のコンセプトを分かりやすい短い言葉でまとめ，関係者の意識を合わせたことが，業務独自試験の実施を成功させる上での重要な要素となったのではないかと考えている．

5.4　出題形式，試験時間の決定

試験の出題形式として，「多岐選択式」と「論述式」のメリット，デメリットを考慮して検討した結果，両方を複合して行ったほうが良いとの結論になり，「多岐選択式」１０問と「論述式」３問で行うこととした．

これは，「多岐選択式」１０問と「論述式」１問で「企業固有スキル活用に必要な知識の網羅性を確認」することが可能で，「論述式」

表２　論述式と多岐選択式のメリット・デメリット

形式	メリット	デメリット
多岐選択式	・解答の傾向分析が容易 ・採点が非常に容易 ・採点基準が明確で，誰が採点しても結果が同じになる ・解答方式の説明が容易．解答方式の間違いによるミスが避けられる	・出題範囲を網羅するために問題数を多数つくらなければならない ・問題の陳腐化を防ぐための見直しの頻度が多くなる ・まぐれの正解を完全には防げない
論述式	・総合的な力が計れる ・まぐれの正解を排除できる ・画一的ではないので，様々な観点から解答の評価ができる	・少数の出題ゆえに，ヤマがかけやすい ・基礎知識や論理的思考ができていても，文章力に評価が左右されてしまうことがあり，テストの目的と違う評価となってしまう危険性がある ・採点時の手間が膨大になる （どう書けば何点，ここまで書ければ何点，という基準を明確にしないと，不公平が生じる．採点するために解答を読まなければならないので，その時間がかなり負担になる．）

２問で「必要な知識を応用した，実践的な課題解決力の確認」が可能だと考えたためである．こうすることにより，ＩＴスキル標準Ｖ３における「レベル４」で定義されている「プロフェッショナルとしてスキルの専門分野が確立し，自らのスキルを活用することによって，独力で業務上の課題の発見と解決をリードするレベル」を充足させるために必要な知識を保有していることが確認できる．

そして，その試験時間は，最大１２０分とし，順調に解ける人は６０分程度で解ける問題量を目指すことにした．

5.5 試験サンプルの作成

ここまで議論した中で，「多岐選択式」については，情報処理技術者試験などの世間一般に広く行われている試験を参考にすれば問題をイメージすることが比較的容易であったが，「論述式」については，何を参考にするかで，ひと工夫必要であった．

メンバの中に総合評価落札方式で行われるシステム開発の入札案件にて，「業者提案依頼書」に回答する提案書を作成した経験のある者が複数名いて，そのメンバから「業者提案依頼書」の内容を参考にすれば良いのではないかとの提案が出された．

確かに，総合評価落札方式の入札では，入札対象のシステムを開発する実力があるかを応札する業者に問うために，「業者提案依頼書」は作成されるので，その内容は試験問題作成の参考となる．しかしながら，「業者提案依頼書」は入札が終われば返却しなければならず，その内容も口外禁止な場合がほとんどであるため，直接参考にすることはできない．ただし，そのメンバは入札対応を数多く行った経験があったため，その提案依頼内容そのものではなく，「業者提案依頼書」では応札する業者にどういうことを問うていたかといった考え方を理解していたので，自身の担当業務で，その考え方に基づいたサンプル問題を作成してくれることになった．

そのほかのメンバも，「多岐選択式」のサンプル問題を作成した．そして，それらをたたき台にして，試験問題作成における課題を抽出し，その課題をある程度クリアにしたところで本格的に問題作成を行うことにした．

ここまで進めた中で，積極的にサンプル問題作成を進めようとするメンバと，「提供された情報が少ないのでこれでは問題の作成ができない」と否定的なメンバに分かれた．しかし，経験のない作業を行う場合には仕方のないことと割り切り，積極的にサンプル問題を作成するメンバが検討を主導する形でワーキング・グループの活動を進行した．

5.6 課題の抽出とその対応

サンプル問題を作成する中で，幾つか課題が抽出され，その対応を行った．主な課題を以下に示す．

5.6.1 参考資料の持ち込みを可とするか不可とするか

参考資料の持ち込みを不可にした場合，とくに「論述式」の解答において，問題の出題意図には十分答えているのにもかかわらず，細かい文言の誤りで不合格になるケースが生じることが懸念された．しかしながら，持ち込み可としても，大きなシステムの設計書だとキングファイル何十冊にもなるので，持ち込むことも試験中に検索することも困難になる．

その解決策として，試験問題に，解答のヒントになるワードと意味のないワードを羅列し，「XX について，以下に示すワードを参考にして答えよ」といった形式で出題することとした．

5.6.2 試験の採点と合格基準をどうするか

「多岐選択式」は解答が１つとシンプルであるが，「論述式」は１つに定まらないので採点基準を決めておく必要がある．

そこで，試験問題の中で論述のポイントを数点示しておき，そのポイントを解答するにあたっての採点基準を設定して，ポイントごとの採点基準を満たしていること，例えば，ポイントが４個あれば，その４個のポイントについて採点基準を満たしていれば正解とすることにした．

そして，「多岐選択式」は１０問中８問以上正解であること，「論述式」３問とも正解の８０％以上の論述ができていることを合格基準とした．問題文のイメージを図４に示す．

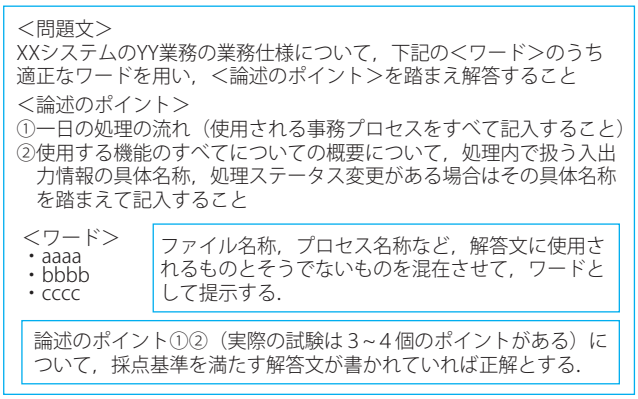


図４　問題文のイメージ図

5.6.3 試験問題のレビューをどうするか

試験問題を作成する分野は独自性が高いことから担当によっては，レビュアとなれる有識者がいないケースがあった．あるいは，レビュアにふさわしい有識者がいても試験受験対象者であるため，レビュアになれないケースもあった．

その対策として，一般的なチェック項目を整理したチェックシート（表３）を作成して，セルフレビュー時に活用してもらうことにした．

更に，レビュアになれる社員がいない担当では，一緒に長年開発の仕事をしているパートナー会社の人に試験問題をレビューしてもらうことも実施した．

表３　試験問題のレビュー観点

チェック項目	チェック観点
試験問題は，システムの業務仕様を問うようになっているか	OS，ミドルウェア，ハードウェアなどのシステム基盤仕様を問う内容になっていないか
試験問題は明確か	複数の解釈が成り立つような，曖昧な文章になっていないか
設問には，ひとつの正しい解答を導く十分な情報があるか	文章内に不足している情報はないか
設問及び選択肢は相互に関連性があるか	例えば「台帳管理対象のうちどれが…」という設問であれば，すべての選択肢は台帳管理対象であることが必要である
設問に，正答となるような単語あるいは語句が表現されていないか	設問の文章内に，正答となる単語，語句そのものが含まれていないこと

5.7 早めのアクションを実施

このように，作成する試験の内容を明確化したところで，ワーキング・グループでの検討結果を「試験問題作成要領」にまとめ，本格的に試験問題を作成する段階に進めることができた．

ここまで約３カ月間と当初の予定通りに作業は進捗し，試験作成期間は約２カ月と，当初余裕を見て設定した期間を最大限使用することができたが，それでも，本来業務が多忙なために，計画した期日までに試験問題が作成できない担当が発生することが懸念された．

そのため，試験作成が間に合わない見込みの担当は早めの報告をするように依頼した．こうすることで，その担当には試験問題作成締切日の１カ月前にアクション会議を行うことができた．

その結果，その担当は試験問題作成者の上司となる管理職の人が試験問題を作成することになり，最終的に当初予定よりも１カ月遅れで試験問題作成が完了した．

6 試験実施に向けて

試験問題は完成したが，試験実施を成功させるにあたっては，試験受験者の立場に立った思慮深い準備が必要である．事業推進担当で，試験実施までに準備した内容を以下に示す．

6.1 受験者へ必要な情報の提供

試験時間，出題形式，持ち込み可能なものなど，試験受験者が事前に知っておくことが必要な内容を試験実施要領にまとめた．

6.2 試験問題と解答用紙の印刷への配慮

試験問題用紙に，印刷のかすれや文字が見にくくなることがないか，実際に印刷して複数人の目で確認した．例えば“，”と“．”の違いが分かりにくいと判断した場合には，フォントのサイズを大きくしたり種類を変えたりして，見間違いが発生しないように注意した．

解答用紙については，試験問題作成者が作成した模範解答と解答欄の枠を見比べて，枠が小さいために解答が書き切れない状況が発生しないかどうかを注意深くチェックした．

また，模範解答を見て，フリースペースの解答欄では，期待される解答に到達できないことが懸念された場合には，解答欄と問題文にひと工夫加えることとした．例えば「ｘｘに必要な要素を答えよ」といった問題で，模範解答として２個の要素を挙げることが書かれていた場合は，試験問題作成者に「ｘｘに必要な要素を２つ答えよ」という問題文にして良いか確認して，解答用紙は，解答用の枠を２つ設けるようにした．

6.3 試験日時の調整及び試験環境の準備

こちらから一方的に試験日時を決定することはせず，候補日を複数設定し，どの候補日も都合がつかない場合，受験者個別に調整するなど，受験者の業務都合に配慮した試験日程調整を行った．

更に，試験会場が，静かで集中できる場所か，試験中に時計が見えて，受験者が残り時間を確認できる場所かなどを確認して，会場の確保を行った．

7 試験結果と考察

論文執筆時点までに，計２回（２０１６年１１月，２０１７年３月）の試験を行った．それぞれの，試験実施分野数，受験者数，合格者数は表４の通りであった．

表４における「業務分野」とは弊社における企業固有スキルを細分化したものである．例えば，情報系システムにおける統計業務の場合，企業の景気動向に関する統計や，経済取引に関する統計など，統計の対象となる分野ごとに細分化している．業務分野数と受験者数の関係は「一対多」の関係になっており，一部の業務分野は，複数人の受験者がいたことを表している．

表４　試験実施結果

回次	業務分野数	受験者数	合格者数
1	6	8	2
2	8	１５	5

表４が示すように，合格者数は受験者の半数以下という厳しい結果であった．しかし，試験が担当の第一人者になるための登竜門的役割を果たすと考えると妥当な結果と考えられる．

試験について，結果の妥当性，得られた効果，今後の課題について，以下に示すような考察を行った．

7.1 結果の妥当性

試験実施結果の最大の懸念は，「問題が不適切であったため，本来は合格となる実力のある受験者が不合格となってしまう」事象が発生することであったが，試験問題作成者（採点も実施）にヒアリングを行った結果，そのような事象は発生しなかった

ことが確認できた．試験結果より，不合格となった受験者のすべてが，自身の担当システムの業務を遂行するにあたって知識が必要な問題に対して不十分な解答しかできていなかったことが明白となった．更に，合格者については，自身の担当業務の範囲内に関する問題に偏ることなく「知識の網羅性を確認」する問題を網羅的に正解できていた．このことにより，試験問題，試験結果の妥当性は確保できたと考えている．

7.2 得られた効果

受験者は日常の業務遂行において，ＩＴスキル標準Ｖ３における「レベル４」相当の技量を発揮していると上司が判断して選定されたので，不合格者はどのようなスキルが不足して不合格となったのかについて，考察を行った．不合格となった受験者個別の出題ごとの解答状況を表５に示す．

表５より，多くの受験者は「実践的な課題解決力を確認」する問題２問のうち１問以上は合格レベル，あるいはそれに近い解答ができていたが，「知識の網羅性を確認」する問題については，とくに 2016 年実施の試験に顕著な傾向が表れているが，解答率は低かったことが分かる．

このことにより，不合格となった受験者は日常の業務遂行に関連の深い知識については，実践的な課題解決力を発揮できるレベルまで知識を定着させることができているが，必要な知識の網羅性は不足している可能性が高いと考えることができる．

また，試験問題作成者（採点も実施）へヒアリングした際に，「この人がこの問題が解けないのは意外だった」という感想が多く聞かれた．これも，受験者の必要な知識の網羅性が不足していることが原因と考えられる．

表５　不合格となった受験者の解答状況

受験者	出題形式			
	知識の網羅性を確認		実践的な課題解決力を確認	
	多岐選択式	論述式 1	論述式 2	論述式 3
2016-1	A	C	B	C
2016-2	C	C	C	B
2016-3	C	C	A	B
2016-4	B	C	A	A
2016-5	A	C	C	C
2016-6	B	C	C	A
2017-1	A	A	A	C
2017-2	B	B	A	A
2017-3	B	C	B	B
2017-4	C	B	B	B
2017-5	A	B	A	A
2017-6	B	C	B	B
2017-7	A	C	B	B
2017-8	B	C	B	B
2017-9	C	A	A	C
2017-10	C	C	C	C

【凡例】　A　：合格レベル（80%以上）の解答
B　：合格レベル未満,50%以上の解答
C　：50%未満の解答
※すべての出題に対する解答がAになれば合格となる.

上司による主観的な判断では，日常の業務遂行で実践的な課題解決力を発揮できていることで，ＩＴスキル標準Ｖ３における「レベル４」相当のスキルがあると判断してしまいがちだが，業務遂行に必要な知識を網羅的に獲得していないと，自身が未経験の課題が発生した場合に対応できない可能性が高い．そのため，この状態では「獲得しているスキルは限定的である」と判断すべきである．

従来は，企業固有スキルについて「獲得しているスキルが限定的か否か」を判断することは情報が不足するため困難だった．しかしながら，社内独自試験を実施することにより，スキル判断の客観性を高めることが可能になることが，試験実施によって明らかになった．

また，一般的な傾向として，担当業務に従事する作業者は直近の業務遂行に直結しない知識については，学習する意欲が低くなりがちであるが，社内独自試験を受験する機会があることで，関連する知識を網羅的に学習する機会，動機付けを与えることができる．このことも，社内独自試験を実施する大きなメリットである．

更に，社内独自試験を行うことにより，受験者の弱みが明確化し，今後のスキルアップのための啓発点を正確に示すことが可能になる．例えば，表５の「2017-1」行の受験者は，かなり合格レベルに近いスキルを獲得しているが「実践的な課題解決力」の獲得に課題があることが分かる．また，「2017-2」行の受験者は「知識の網羅性」に課題があることが分かる．このように受験者の解答状況を確認することで，今後のアドバイスを具体的かつ的確に行えるようになり，効率的なスキルアップの実現が期待できる．

効率的なスキルアップが実現することにより，後進の育成期間が短縮するので，単一のシステムに有識者として長年従事している社員を後進の社員に交代して，別の業務を経験させるジョブ･ローテーションが実現できるなど，人材育成上のメリットは大きい．

以上より，企業固有スキルのレベルを判断するために社内独自試験を実施することは，ＩＴ技術者の人材育成上の効果が大きいことが確認できた．

7.3 今後の課題

ここまで述べたように，現段階では社内独自試験の実施を成功させることができたと考えているが，今後継続的に運用していくにあたって課題と考えている点を以下に示す．

7.3.1 ワードの与え方（出題上の工夫）

解答の補助となるワードの与え方が，文字の羅列だと分かりづらいといった声が受験者より複数あがった．

そのため，図や表などを使用していきたいと考えている．一例をあげると，処理日付を意識する必要がある業務処理の問題で，カレンダーを効果的に使って出題している担当があった．

このように，分かりやすさを重視した出題とするための工夫が必要であると考えている．

7.3.2 問題の難易度の明確化

担当業務により，かなり深い知識が必要な場合とそうでない場合があり，担当によって問題の難易度に差があることが分かった．

この点については，試験実施を重ねることで，業務経験が何年程度で合格レベルに達するかが分かってくると考えている．

そして，「試験に合格するための業務経験年数の目安」が分かり難易度を明確にできるのではないかと考えている．

こうすることで，担当業務ごとに第一人者になるために必要な経験年数も明確化するので，ジョブ・アサインやジョブ・ローテーションといった人材育成施策を計画する上でも有益なのではないかと考えている．

7.3.3 試験問題の維持管理

今回取り決めた試験運用では，試験問題の維持管理負担を軽減させるために，試験問題を再利用することを前提（試験問題は事業推進担当が厳秘の扱いで厳重に管理している）とし，受験後の持ち出し禁止を徹底することとした．

しかし，法律の改正や技術の進歩などによって，問題が陳腐化しないか，再受験者向けに異なる問題を用意する必要があるかを試験ごとに検討し，必要があれば試験問題の再作成を行う必要がある．このような場合に，運用負担が大きくならないよう考慮した維持管理運用を行っていく必要がある．

現状では，試験の実施を半期に一度の周期で設定しているので，半期に一度，試験問題作成担当者に事業推進担当より，試験問題変更の要否を確認する運用を行っている．

8 社内独自試験の実施方法についての提案

これまで述べてきた経験を踏まえて，試験問題を作成した経験のないIT企業が社内独自試験を実施する場合の作業の進め方と，試験問題の選定を中心に社内独自試験の実施方法について提案する．

8.1 作業の進め方

社内独自試験の実施に必要となる作業プロセスのフローを図５に示す．

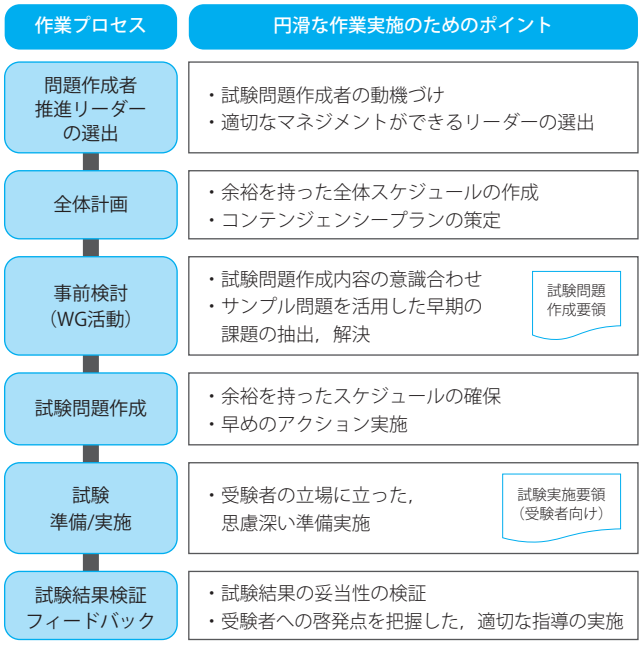


図５　作業プロセスのフロー図

図５に示した作業プロセスごとに，円滑な作業を実現するためのポイントを以下に示す．

8.1.1 問題作成者・推進リーダーの選出

「実践的な課題解決力を確認」する問題を作成することから，問題の作成はコストセンター部門に従事している社員ではなく，プロフィットセンター部門において現場の第一人者となっている有識者の社員が行うべきである．

しかしながら，このような有識者は本来業務で多忙な可能性が高いため，タスク割り当てにおける配慮と，十分な意識付けが必要である．意識付けについては，社内独自試験の作成が人材育成上のメリットが大きい側面と，自身の知識を試験問題として形式知化することで，自身の技術者としてのスキルアップが可能となる側面の２面で行うと良い．

推進リーダーは，問題作成者が本来業務を行いながら，試験作成を行うことを考慮したマネジメントができる人材が行うべきである．

8.1.2 全体計画

推進リーダーは問題作成者が本来業務と並行して作業することを考慮し，余裕を持ったスケジュールで全体計画を立案すべきである．

また，問題作成者の本来業務の状況により，問題作成が不可能になる事態が発生することも想定すべきである．

そのため，コンテンジェンシー・プランを作成し，問題作成者の担当業務の責任者と，その内容についてあらかじめ合意しておくことが必要である．

8.1.3 事前検討（WG活動）

問題作成者が問題作成内容やレベルについて早期にイメージすることができるようになれば，試験問題作成の期間を長く設定することができる．

そのために，可能な限り早めにサンプル問題を作成し，課題の抽出，解決を行うことが重要である．この作業プロセスで取り決めたことは「試験問題作成要領」としてドキュメント化すべきである．

こうすることで，試験問題の維持管理や追加の試験問題作成を円滑に実施できるようになる．

8.1.4 試験問題作成

計画した期日までに試験問題が作成できるかどうかの判断ポイントを早め（遅くとも期日の１カ月前）に設定して，問題が発生していた場合は，早期のアクションを実施することが重要である．

8.1.5 試験準備／実施

受験者の立場に立った思慮深い試験準備が必要である．そのために分かりやすい内容で「試験実施要領（受験者向け）」を作成することも重要である．

8.1.6 試験結果検証・フィードバック

試験の合否に一喜一憂するのではなく，結果を正しく分析して，受験者のスキルアップに効果的なフィードバックを行うことが重要である．

8.2 試験問題の選定方法

企業固有スキルについてＩＴスキル標準Ｖ３における「レベル４」相当の知識を確認するための試験問題として，表６に示す内容で試験問題を選定することを提案する．

表６ 「レベル４」相当の確認に適した試験問題

出題目的	出題形式	出題数	出題内容の例
知識の網羅性を確認	多岐選択式	10 問	・業務の時限性、関連法規など
	論述式	1 問	・重要な業務プロセスの特徴など
実践的な課題解決力を確認	論述式	2 問	・障害への対応 ・仕様変更要求への対応

出題目的別の具体的な出題内容を以下に示す．

8.2.1 知識の網羅性を確認する出題内容

多岐選択式は多種多様な問題の出題に適している．多岐選択式の問題を作成する前に，企業固有スキルとして必要な知識の範囲を再度確認して，偏りのないように，幅広く出題すべきである．筆者らの場合は，試験問題作成前に企業固有スキルとしての出題分野の全体を表形式または箇条書きで，体系的に整理した資料を作成した．

出題内容の例としては，業務の時限性を問う問題や，業務に関連する法規に関する問題などが考えられるが，情報処理技術者試験の午前問題を参考にすることで，多種多様な問題を想起することは容易である．なお，ここで述べている「業務の時限性」とは，関連する取引相手や法規上の都合により，帳票を出力する時間に制限がある業務や，資金決済の取引で，取引発生から決済完了までの時間に制限がある業務などを指す．このような時限性の理解は，担当業務を遂行する際の重要な知識となる．

また，多岐選択式の問題だけでは，知識の理解度の深さを判断することが難しいため，論述式の問題も出題すると良い．出題内容の例としては，業務プロセスのフロー図を提示する中で，重要なプロセスを空白にして，その名称，概要，特徴を論述する問題が考えられる．とくに特徴を論述する問題では，受験者がその特徴の本質をどれだけ理解しているかを確認することが可能となる．

8.2.2 実践的な課題解決力を確認する出題内容

実践的な課題解決力を確認するには，障害対応の事例問題や仕様変更要求の事例問題が適している．

障害対応の事例としては，障害発生後に直接の原因を除去して処理を再実行するだけでは解決しないような事例が良い．例えば，回復にあたって障害個所とは直接関連しないオンライン処理の取り消し処理を実行して，再入力しないとデータベースに不整合が発生してしまうなど，業務処理の関連性を正確に把握していないと影響範囲の判断を誤るような問題であれば実践的な課題解決力を確認することができる．

具体的な出題内容の例としては「障害の事象から想定される原因」「障害復旧方法」「復旧にあたって考慮すべき事項」を問うと良い．

仕様変更要求の事例としては，例えばデータベースの項目追加の要求があった場合に，外部システムへ連携するファイルのレイアウトが変わるケースや，帳票の出力内容が変わるケースなど，その変更に対する影響範囲が広い事例が適している．

具体的な出題内容の例としては「仕様変更要求の影響範囲」「変更する際に考慮すべき事項」を問うと良い．

出題数は，限定的なスキルのみで問題のすべてを解答できてしまうことを防ぐために，1 問では不十分と考え 2 問と設定した（最適な出題数については，企業固有スキルの特性によって変化すると考えられるが，問題作成者，受験者の負担を考慮して出題数を決定すべきである）．

9 総括

IT 企業における IT プロフェッショナルの育成において，企業固有スキルとなる分野の社内独自試験を作成し実施することは，担当者が担当業務の遂行に必要な知識をどのレベルまで保有しているかを客観的に判断することが可能となり，更に，弱点の見える化により今後の啓発点が明確化するなど，メリットは大きい．

中小規模の組織で，試験問題を作成し維持することは負担が大きいと感じられるかもしれないが，一日も早く社内独自試験を定着させることが，IT 企業としての強みとなる企業固有スキルについて，他社の上をいくレベルの技術力を獲得し，競争力を高めビジネスを有利に展開できることにつながる．

試験問題を自ら作成した経験のない IT 企業が，社内独自試験の実施を成功させるのは容易なことではないが，本稿を参考にすることで，成功の可能性は高くなると考えられる．

本稿をきっかけに，社内独自試験のみならず企業固有スキルについての人材育成が注目され，新たな知見が提示されることを期待する．

【参考文献】

[IPA2009] 独立行政法人情報処理推進機構（IPA），社内プロフェッショナル認定の手引き ,2009.
https://www.ipa.go.jp/jinzai/itss/download.html
[IPA2012] 独立行政法人情報処理推進機構 (IPA)，ＩＴスキル標準Ｖ３,2012.
https://www.ipa.go.jp/jinzai/itss/download.html
[IPA2015] 独立行政法人情報処理推進機構（IPA），ｉコンピテンシディクショナリ解説書 ,2015.
https://www.ipa.go.jp/jinzai/hrd/i_competency_dictionary/kiyaku4.html
[Tanaka2010] 田中久也 , 社内プロフェッショナル認定に対する IPA の取り組み , 独立行政法人情報処理推進機構 (IPA),2010.
https://www.ipsj.or.jp/10jigyo/forum/software-j2010/tanaka.pdf

国際規格に基づく総合的なソフトウェア品質評価の枠組みとその実製品への適用による品質ベンチマーク


鷲崎 弘宜[※1※4]


津田 直彦[※1]


本田 澄[※1]


中井 秀矩[※1]


深澤 良彰[※1]


東 基衛[※1]


込山 俊博[※2]


中野 正[※3]


鈴木 啓紹[※3]

ソフトウェア製品の品質を業界の中で定量的に明らかとし改善可能とするため，多面的な品質を測定評価する共通の枠組みが必要である．我々は国際規格シリーズ SQuaRE（Systems and software Quality Requirements and Evaluation）の具体化により，製品によらず共通に製品品質と利用時品質を総合的に測定評価可能な枠組みを実現した．更に枠組みを 21 製品に実適用し，品質特性別の傾向，品質特性間の関係，利用時品質・製品品質間の関係，及び製品コンテキストと品質特性の関係の一端を，適用した範囲において明らかとし，品質ベンチマークとして公開した．

Software Quality Evaluation Framework based on International Standards and Benchmark obtained by applying the Framework

Hironori Washizaki[※1※4], Naohiko Tsuda[※1], Kiyoshi Honda[※1], Hidenori Nakai[※1], Yoshiaki Fukazawa[※1], Motoei Azuma[※1], Toshihiro Komiyama[※2], Tadashi Nakano[※3], Hirotsugu Suzuki[※3]

※1　早稲田大学　※2　日本電気株式会社　※3　一般社団法人コンピュータソフトウェア協会
※4　国立情報学研究所，株式会社システム情報，株式会社エクスモーション

We established a comprehensive software quality evaluation framework based on ISO/IEC 250nn series SQuaRE (Systems and software Quality Requirements and Evaluation). Our framework successfully realized many product measures and quality in use measures originally defined in the SQuaRE. By applying the framework to 21 commercial ready-to-use software products, we revealed the current status of software product quality and opened it as a benchmark to the public. The benchmark includes the trends of quality measurement values, relationships among quality characteristics, relationship between quality in use and product quality, and relationship between quality characteristics and product contexts within the limits of the application.

1　はじめに

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り [東 15]，その実現に向けて利用者のほか多様な利害関係者への影響に着目したソフトウェアの利用時の品質の把握，開発者視点の製品品質の把握，並びにその間の関係の把握が欠かせない．しかしながら従来のソフトウェア製品の品質評価の取り組みは，機能適合性や信頼性などの特定の品質特性に集中しており，様々な品質特性を多面的かつ具体的に捉える取り組みを欠く．結果として，製品や組織を超えた測定データが限られており，品質の実態，更には品質特性間や利用時の品質と製品品質間の関係は不明であり，効果的な品質向上の取り組みの検討と実施の妨げとなっている．

製品や組織を超えた多面的，客観的かつ標準的な品質測定評価に役立てられることが期待される枠組みとして，ISO/IEC 250nn シリーズ SQuaRE [ISO25000] が挙げられる．SQuaRE では，特定のドメインや製品によらずに，一般的に重要と考えられる品質特性，品質測定量及び評価方法をまとめている．しかしながらそれらの定義は国際規格という性質上，汎用的かつ抽象的なものにとどまり，利用にあたって具体化が必要である．

そこで我々は，IPA から 2015 年度ソフトウェア工学分野の先導的研究支援事業（RISE）の委託を受けて，第三者が利用可能な形で提供されるパッケージソフトウェアやクラウドアプリケーションなどのソフトウェア製品を対象に，製品や組織を超えて共通に適用できるように SQuaRE における品質測定量を品質測定法として具体化及び拡張し，得られる測定値を多面的な品質特性群の単位でまとめ上げる総合的な品質評価の枠組みを実現した．更に同枠組みを 21 の実ソフトウェア製品群に共通に適用することで，品質特性別の傾向，品質特性間の関係，利用時品質・製品品質間の関係，及び製品コンテキストと品質特性の関係の一端を，製品や組織を超えて明らかとした．我々はそれらの結果を世界初の総合的な品質ベンチマーク WSQB17: Waseda Software Quality Benchmark[WSQB] として公開した．企業において枠組みと品質ベンチマークの活用が始まっている [小島 17].

本稿では 2 節で関連研究を説明する．3 節で品質評価枠組みを提案し，4 節で製品群への適用による調査の結果を報告すると共に業界などへの提言を述べる．最後に 5 節で成果や展望をまとめる．

2　関連研究

上流工程で検証される静的な内部品質が，下流工程で検証される動的な外部品質に影響し，外部品質が運用時に評価される利用時の品質に影響を与えることがうたわれている [ISO25000]．その前提のもとで，開発の上流工程では設計や実装において内部品質を作り込み，テストにおいて外部品質を確認及び補正する．しかし，その影響関係は限定された範囲でのみ明らかであり，結果として総合的及び様々な立場における品質評価に向けて品質測定法や測定結果は十分に活用されていない [鷲崎 07]．例えば設計モデルやソースコードを測定して内部品質を評価する枠組みは QMOOD[Bansiya02], Ortega らのスイート [Ortega03], 我々の過去のスイート [鷲崎 07][鷲崎 10] など数多く存在するが，最終的な顧客満足度を含む利用時の品質までを網羅するものではなく，それらの関係も明らかではない．

SQuaRE では利用時の品質と製品品質の代表的な品質特性及びサブカテゴリーとしての品質副特性を 2 層構造でモデル化した ISO/IEC 25010:2011 [ISO25010] (JIS X 25010:2013)，同品質モデルに基づき利用時の品質測定量をまとめた ISO/IEC 25022:2016 [ISO25022]，製品品質測定量をまとめた ISO/IEC 25023:2016 [ISO25023] (JIS X 25023:2018) が策定されている．SQuaRE（及び前身にあたる ISO/IEC 9126 シリーズ）におけるそれらの定義は汎用性を重視し抽象的であり [Arban03][Heidrich14]，直ちに利用可能な具体的なソリューションには至っていない [Biscoglio14]．加えて，測定量間の関係も実証されていない．

パッケージ製品やクラウドアプリケーションの品質要求として国際規格 ISO/IEC 25051:2014 [ISO25051] (JIS

X 25051:2016) があり，同規格及び IPA のソフトウェア品質説明のための制度ガイドライン [IPA14] に基づき，コンピュータソフトウェア協会（CSAJ）により JIS X 25051:2016 に準拠した PSQ 認証制度 [PSQ] が 2013 年から開始されている．PSQ は業界初のソフトウェア品質認証制度であり，一定水準のテスト結果に基づき，ソフトウェア製品が品質要求を満足することを認証するものである．本稿の品質評価枠組みはこれらに対して，同様の製品を対象に品質特性単位の詳細な測定と品質特性間の関係分析を通じて補う関係にある．

開発組織を超えたソフトウェア開発の実態調査としては IPA データ白書 [IPA15] や Cusumano らによる調査 [Cusumano03]，Jones による調査 [Jones09] などがあるが，いずれも製品品質と利用時の品質の一部の関係分析にとどまり，品質特性について網羅性を欠く．

3　総合的な品質評価枠組み

本研究が実現する枠組みの目的を，国際規格に基づくことで製品や組織を超えて共通に実適用可能とし，更に，多面的かつ具体的に様々な品質特性を捉えられることとする．本研究の結果として得られる枠組みは，製品の開発側や運用側において，開発・保守・運用中あるいは運用検討中のソフトウェア製品の品質を，客観的，定量的かつ総合的に評価し，開発・保守における品質要求定義や品質改善，取捨選択の判断材料に役立てられることが期待される．実際に，枠組みを企業において製品に適用して品質特性単位の相対的なポジションを明らかにし，以降の品質改善に役立てる活用が始まっている [小島 17].

実現した総合的な品質評価枠組みの全体像を図 1 に示す．ソフトウェアの利用時の品質と製品品質（外部品質及び内部品質）のすべてについて，ISO/IEC 25051:2014 や PSQ 認証制度を参考に，ISO/IEC 25010:2011 における品質モデルの品質特性を網羅的に扱う．更に，製品の別を問わずにおおむね共通に得られる属性を用いることで ISO/IEC 25022:2016 及び ISO/IEC 25023:2016 における測定量について測定の方法を具体化して測定法としてまとめる．具体化にあたり，我々が実績を有する信頼性評価の仕組みやソースコード解析の仕組みを組み入れる．

測定値を後述の方法でスコアとして正規化し，品質特性単位で集約可能とする．品質特性単位のスコアを用いて，品質特性間の関係及び利用時の品質と製品品質間の関係を分析し，その結果を品質ベンチマークとしてまとめる．また製品ごとに品質評価結果，品質ベンチマークにおける製品のポジション，並びに，品質向上に向けた人手による助言を品質診断レポートとしてまとめて製品開発元に提供した．レポートの例については [WSQB] を参照されたい．

測定の詳細を以下に説明する．

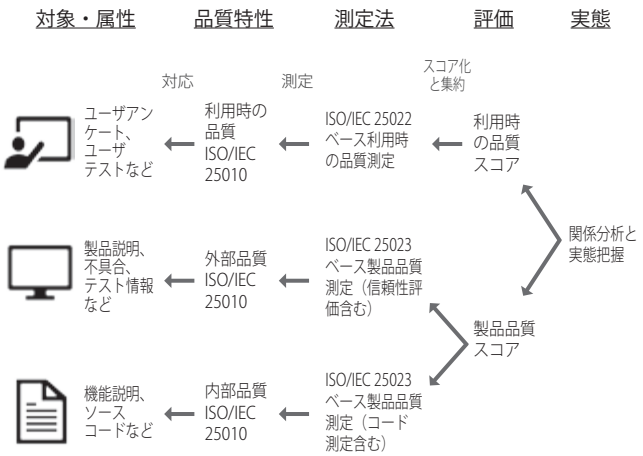


図 1　品質評価枠組みの全体

3.1　製品品質の測定

品質測定法の具体的な定義にあたり，測定可能な事柄や測定法ありきで検討すると，本来の目的を見失う可能性がある [鷲崎 16][Washizaki17]．そこで、目標指向の枠組みである Goal-Question-Metric（GQM）法 [Basili94] を適用し，製品品質の品質副特性を網羅するように合計 66 の品質測定法を定義した．適用にあたり，JIS X 25051:2016 を参考とし，ISO/IEC 25010:2011 の品質特性を明確に評価することを目標とした．また，ISO/IEC 25023:2016 における製品品質測定量を，測定法として具体化する際の初期候補とした．測定法の一覧を付録 A.1 に示す．

測定法の定義の例を図 2 に示す．図ではセキュリティの副特性である否認防止性について経路のデジタル署名に着目し，目標達成可否を判断するための種々の側面を網羅するように質問を設定し，最終的には測定結果の分かりやすさを重視して種々の側面を総合する形で一つの測定法へまとめあげている．

ISO/IEC 25023:2016 の解釈や測定データの現実的な利用可能性について SQuaRE 及びその JIS 化を担当した国内委員会のメンバからの助言を得た．そして測定に必要なデータを入力し測定値を得るための記入様式を策定し公開した [WSQB]．例えば否認防止性については様式において，経路数及びそのうちの署名経路数の記入を求める．

21 製品に適用した結果，57 の測定法について 1 製品以上で測定値が得られた（表 1）．大多数について，測定値が得られたことから，もともと抽象的な測定法を具体化して実際の製品に適用し，客観的かつ多面的に品質を測定でき

たことが分かる．また測定できた割合は 66 測定法× 21 製品のうちで 34% であった．未測定の原因の多くは根拠となるデータの開発中の未記録である．

測定法導出にあたり目標に基づき，我々が実績を有する信頼性評価及びソースコード測定法を組み入れた．

- テスト時の発見欠陥数の記録に対して信頼性成長モデル [Honda17] を適用して，①欠陥の発見状況を安定したと考えられるタイプ（安定タイプ），②漸増的に増加しつつあるタイプ（漸増タイプ），③爆発的に増加しつつあるタイプ（爆発タイプ）の 3 種に大別する．
- ソースコードの静的解析ツールとして Understand と CheckStyle を用いて，保守性の副特性であるモジュール性と再利用性の測定法（複雑度，結合度，コーディング規約違反数など）[鷲崎 07][鷲崎 10] を組み入れた．

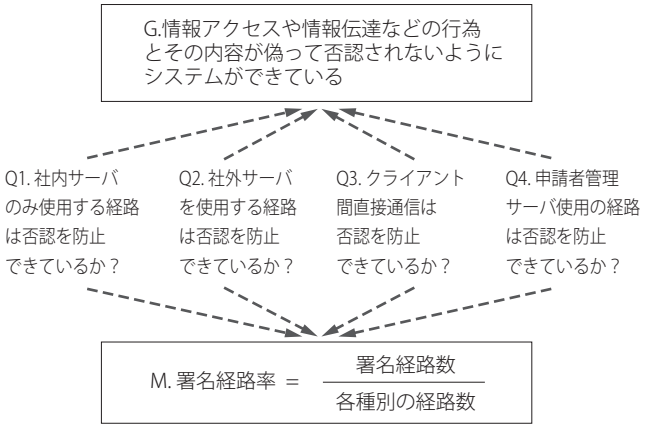


図 2　GQM 法適用例（G: 目標，Q: 質問，M: 測定法）

表 1　製品品質の測定法の定義数
（あり:測定値が得られたもの, なし:測定値が得られなかったもの）

特性	機能適合性	性能効率性	互換性	使用性	信頼性	セキュリティ	保守性	移植性
あり	4	10	2	9	13	6	8	5
なし	0	1	0	0	3	0	4	1

3.2 利用時の品質の測定

GQM 法を適用し，利用時の品質の品質副特性を網羅するように合計 17 の品質測定法を定義した．適用にあたり，JIS X 25051:2016 を参考とし，ISO/IEC 25010:2011 の品質特性を明確に評価することを目標とした．また，ISO/IEC 25022:2016 における利用時の品質測定量を，測定法として具体化する際の初期候補とした．測定法の一覧を付録 A.2 に示す．そして測定に必要なデータを入力して測定値を得るための記入様式，ユーザアンケート，及び，ユーザテストの実施方法を策定した．

- 満足性，リスク回避性及び利用状況網羅性の測定法の多くは利用者の実感に基づくものであるため，その測定のためのユーザアンケートを作成し公開すると共に，製品開発元から実利用者へ配布及び回収した [WSQB].
- 有効性と効率性の測定法の多くは利用時の具体的な操作に基づくものであるため，ユーザテストの実施により測定可能な形で定義した．限られた時間内で効率よく主要な機能の正常系及び異常系テストを実施するため，製品開発元が機能一覧及び正常系テスト項目を用意し，その参照により外部評価機関の支援を受けて対象領域やテスト項目への理解を深めながら我々が異常系テスト項目を定義し，我々が実利用者に代わりテストした．

21 製品に適用した結果，全 17 の測定法について 1 製品以上で測定値が得られた．すべての測定法について，測定値が得られたことから，もともと抽象的な測定法を具体化して製品へ実適用し，客観的かつ多面的に品質を測定できたことが分かった（表 2）．また測定率は 17 測定法× 21 製品のうちで 24.4% にとどまった．未測定の原因の多くは，製品開発元における都合などの理由によりユーザテスト及びユーザアンケートを実施できなかったことにある．

表 2　利用時の品質の測定法の定義数

特性	有効性	効率性	満足性	リスク回避性	利用状況網羅性
あり	4	2	6	3	2
なし	0	0	0	0	0

3.3 測定値に基づく品質評価

測定値の閾値を，人手を介さずに測定値の分布の違いによらず妥当に決定する方法として，パーセンタイルの利用が知られている [Alves10]．そこで我々の枠組みにおいて，値域の異なる測定法の違いによらず妥当に正規化し集約可能とするために，図 3 に示すように多くの製品群から得られた測定値の集合におけるパーセンタイルのランク値を測定法のスコアとする．測定法のスコアの平均を副特性のスコアとし，その平均を品質特性のスコアとする．ランク値は 0 ～ 1 の値を取るが，{0,1,5,5} のように集合中で最大の

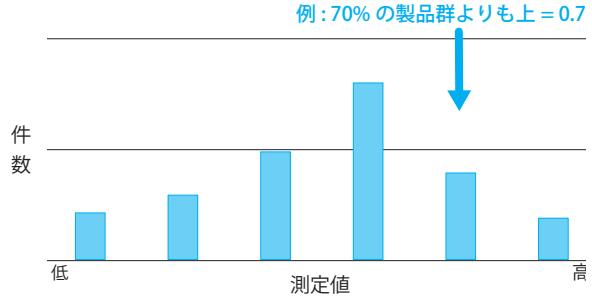


図 3　パーセンタイルによるスコア化

値（ここでは 5）が複数ある場合，ランク値の最大値は 1 にならない（ここでは 0.5）．

測定法によっては，特定の品質特性の観点から値が大きいほうが望ましいものと，値が小さいほうが望ましいものがある．例えばセキュリティの観点から，通信経路がデジタル署名によって改ざんから保護されている割合を示す「ネットワーク経路のデジタル署名対応率」は大きいほうが望ましい．一方，保守性の観点から，関数の制御フローグラフにおける線型独立な経路数を示す「関数のサイクロマティック複雑度」は，同じ要求を満足する限り複雑さが抑えられていることが望ましいため，値が小さいほうが望ましい．そこで，前者の場合は上位 30% のランク値が 0.7 となるように計算し，後者の場合は逆に下位 30% のランク値が 0.7 となるよう計算する．この方法は，測定値が適当な範囲の値を取ることが望ましい場合を考慮しておらず，今後の課題である．

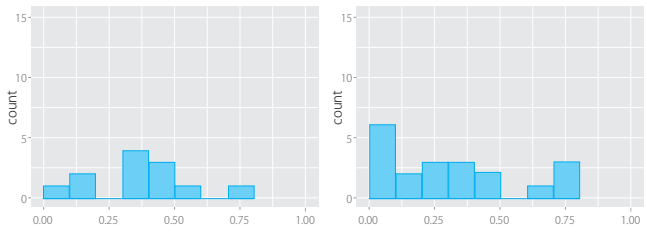


図 4　機能適合性（左），性能効率性（右）の傾向

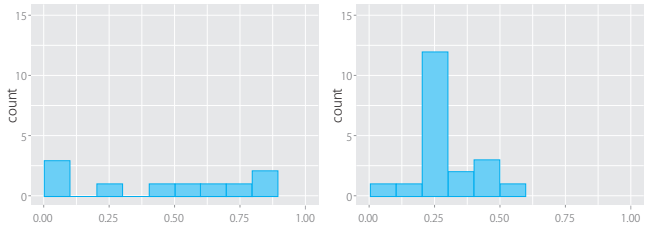


図 5　互換性（左），使用性（右）の傾向

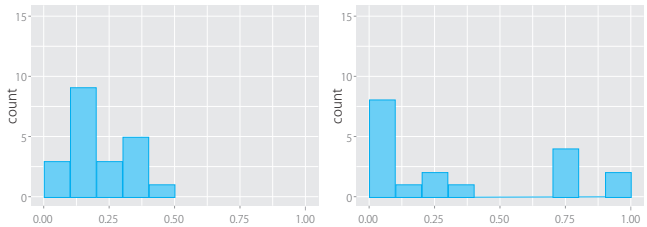


図 6　信頼性（左），セキュリティ（右）の傾向

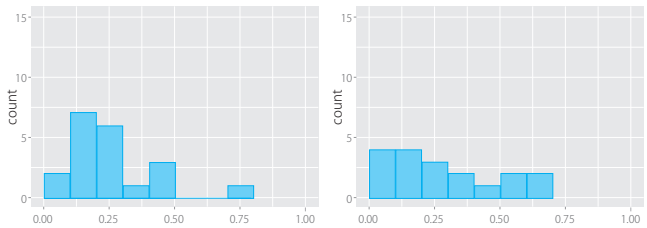


図 7　保守性（左），移植性（右）の傾向

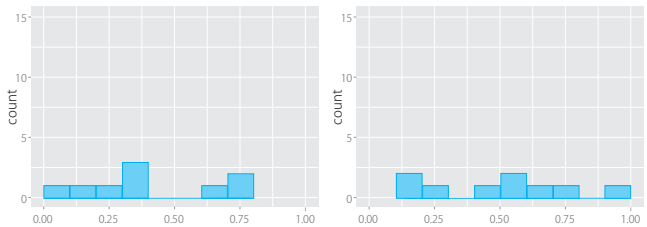


図 8　有効性（左），効率性（右）の傾向

4　枠組み適用による調査結果

製品や組織を超えた品質の実態が不明であることが，効果的な品質向上の取り組みの検討と実施の妨げとなっている．

そこで我々は CSAJ の協力を経て様々な組織が開発した 21 製品に枠組みを適用し，品質特性別の傾向，品質特性間の関係，利用時品質・製品品質間の関係，及び製品コンテキストと品質特性の関係を測定評価できた範囲において明らかとし，品質ベンチマークとしてまとめた．ただし製品ごとに，提出データに基づき測定評価できた品質特性は異なる．以下に結果を報告する．加えて，品質向上及び測定法拡充の観点から考察する．

4.1　品質特性別の傾向

品質特性ごとに，スコアの製品数分布を図 4 ～図 8 に示し，それぞれの結果と考察を以下に述べる．最初に製品品質について以下に述べる．

(1) 機能適合性，性能効率性，移植性：

- ●結果：分布としてなだらかに広がり，製品によりそれらの品質の程度が異なることが分かる．
- ●考察：製品ごとに異なる品質要求の程度に応じた結果と推測され，自然なものと捉えられる．

(2) 互換性：

- ●結果：2 極化している．データ交換などの互換性に通じる仕組みを一部の製品において考慮していないことが原因として挙げられる．また，SQuaRE における定義から具体化できた測定法が 2 つにとどまったため，1 つの測定値がスコアへ多大な影響を及ぼすこととなった．
- ●結果：互換性は，品質モデルが ISO/IEC 9126-1:2001 から ISO/IEC 25010:2011 へ改訂される際に品質特性として格上げされたものである．国際規格側で，具体化して実適用可能な測定量の拡充が望まれる．

(3) 使用性：

- ●結果：得られたスコアの範囲内で低いほうに製品が集中している．

	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	有効性	効率性	満足性	リスク回避性	利用状況網羅性
機能適合性	0.31	0.19	**-0.72**	0.37	-0.05	0.50	0.31	-0.14	0.52	1.00	1.00	1.00
性能効率性		0.44	0.24	0.36	-0.17	0.37	0.32	0.32	-0.10	-0.50	-0.50	-0.50
互換性			0.04	0.17	-0.06	0.36	-0.04	-0.14	0.05	-0.50	-0.50	-0.50
使用性				0.17	-0.21	0.11	**0.44**	-0.09	-0.20	-1.00	-1.00	-1.00
信頼性					0.30	**0.41**	**0.45**	-0.08	0.11	1.00	1.00	1.00
セキュリティ						-0.06	0.19	**0.64**	-0.34	0.50	0.50	0.50
保守性							0.26	-0.29	0.01	1.00	1.00	1.00
移植性								-0.21	**0.67**	0.50	0.50	0.50
有効性									0.03	-1.00	-1.00	-1.00
効率性										1.00	1.00	1.00
満足性											1.00	1.00
リスク回避性												1.00

図９　品質特性単位の品質スコア間の相関係数

●考察：原因としては，使用性を十分に考慮できていないこと，あるいは，エンドユーザ対象ではないといったことから意図的に考慮していない製品が多かった可能性がある．スコアの低い製品について，品質要求との対応関係を確認し，意図通りでない場合は使用性向上のための機能の作り込みといった取り組みが必要と考えられる．

(4) 信頼性：

●結果：分布として狭く，全体的に同程度のスコアを取っている．

●考察：狭義の品質として高い信頼性を作り込んだ上で製品をリリースしていることがうかがえ，日本らしいソフトウェア製品作りと捉えられる．

(5) セキュリティ：

●結果：2 極化している．一部の製品で，暗号化や破損防止などの高セキュリティ化に通じる仕組みを考慮していないことが原因である．

●考察：すべてがつながる IoT/IoE 時代において，企画時にほかとの接続利用を想定していない製品についても保守や派生の中でほかと接続が求められる可能性がある．従って，製品に必要なセキュリティについて慎重な検討が必要と考えられる．

(6) 保守性：

●結果：得られたスコアの比較的広い範囲内で低いほうに製品が集中している．

●考察：全体的に保守性を十分に考慮できていない製品が多いと考えられる．製品寿命や品質要求との対応関係を確認し，意図通りでない場合は設計・実装上の複雑さを抑えるといった保守性向上の取り組みが必要と考えられる．

続いて利用時の品質について以下に述べる．

(7) 有効性，効率性：

●結果：2 極化している．ユーザテスト実施時にタスク実行に難がありタスクを達成しにくい製品が一部見られたためである．

●考察：製品の価値を判断する立場は本来利用者であり，これらの利用時の品質が低い製品についてユーザ中心の製品デザインといった利用者視点のタスク実行のしやすさの考慮が必要と考えられる．

(8) 満足性，リスク回避性，利用状況網羅性：

●結果：測定評価できた製品数が 3 製品と極めて限られ，意味のある傾向を得られなかった．

●考察：多くの製品について開発者視点の製品品質を重視するあまり，利用者視点の利用時の品質の把握と改善を軽視していた可能性がある．これらの品質特性の測定評価のためにユーザアンケートのような利用者の声・実感を直接的に把握する取り組みが不可欠である．

4.2　品質間の関係分析

21 製品のうち測定評価できたものについて品質特性スコア間のスピアマンの順位相関係数を図 9 に示す．図 9 に記載した数値は相関係数の大きさである．また，相関係数について p 値が 0.10 未満であるものは偶然に高い相関係数が算出された可能性が低い（つまり統計的に有意）と考えられるため，点線の枠で印を付けた．統計的に有意と見られた関係のそれぞれの結果と考察を以下に述べる．

(1) 信頼性と保守性，移植性の関係：

●結果：信頼性が高いほど保守性や移植性が高い傾向にあった．

●考察：高信頼が求められる製品において，併せて，長期間における保守や様々な環境への適合並びに移植が求められる場合が考えられる．そのために正の相関が見られた可能性がある．

(2) 移植性と使用性，信頼性，効率性の関係：

●結果：移植性が高いほど，使用性や信頼性及び効率性が高い傾向にあった．

●考察：様々な環境に対する移植のしやすさを製品へ作り込む過程において，併せて，移植検討先の様々な環境の都合から当該製品の品質を多面的に再検討する場合が考えられる．そのために結果として，高い信頼性などの製品への作り込みにつながった可能性がある．

(3) 機能適合性と使用性の関係：

●結果：機能適合性が高いほど，使用性が低い傾向にあった．

●考察：原因として幾つかの可能性が考えられる．製品の当初の企画通りの機能仕様の満足を最重視した結果，利用者にとって本来重要な使いやすさを損なってしまったという副作用の可能性がある．また逆に，使いやすさを重視した結果として，一部の機能を満足できなかったという可能性がある．あるいは品質要求として，そもそもエンドユーザ向けの製品ではないといった理由から，使いやすさを重視していなかった可能性がある．

(4) セキュリティと有効性の関係：

●結果：セキュリティが高いほど，有効性が高い傾向にあった．

●考察：これは製品によってはセキュリティ関連の機能（ログインなど）が一定割合を占め，それらをユーザテストにおいて正確に実行できたことが影響している可能性がある．

4.3 欠陥発見状況タイプと品質特性

時系列の欠陥票を得られた 9 製品について信頼性成長モデルを適用した結果，欠陥発見状況は安定タイプが 3 製品，漸増タイプが 3 製品，爆発タイプが 3 製品であった．図

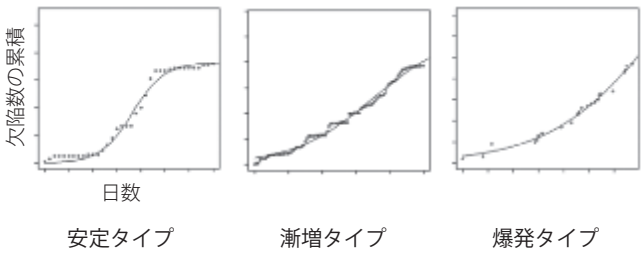


図 10　欠陥発見状況と欠陥数予測（横軸：日数，縦軸：欠陥数の累積）

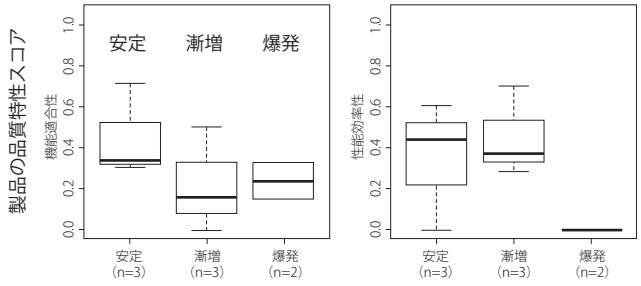


図 11　欠陥発見状況タイプ（左から安定，漸増，爆発）と機能適合性（左），性能効率性（右）の関係

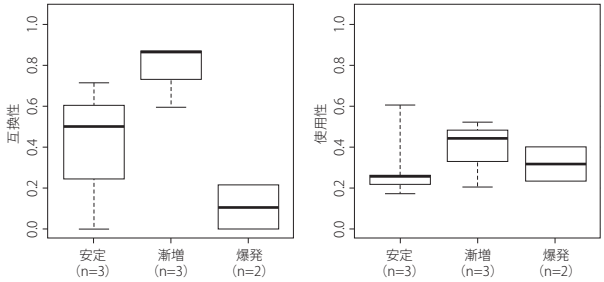


図 12　欠陥発見状況タイプ（左から安定，漸増，爆発）と互換性（左），使用性（右）の関係

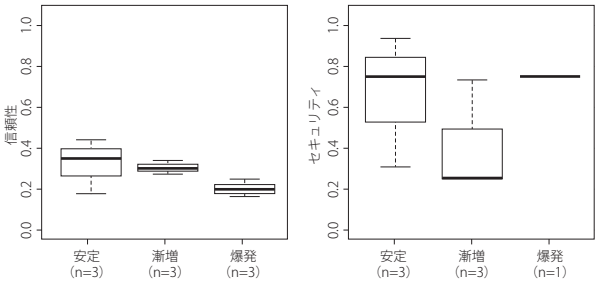


図 13　欠陥発見状況タイプ（左から安定，漸増，爆発）と信頼性（左），セキュリティ（右）の関係

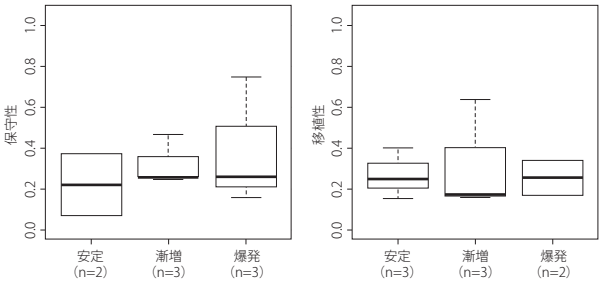


図 14　欠陥発見状況タイプ（左から安定，漸増，爆発）と保守性（左），移植性（右）の関係

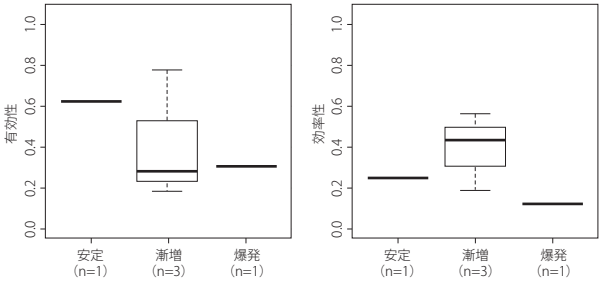


図 15　欠陥発見状況タイプ（左から安定，漸増，爆発）と有効性（左），効率性（右）の関係

10 に各タイプの欠陥発見状況及び欠陥数予測の例を示す．安定タイプについては十分な欠陥を発見できていると考えられる．漸増タイプについては，テストを実施するたびに欠陥が徐々に発見されており，リリース後も欠陥が発見される可能性があると考えられる．爆発タイプについては今後も多くの欠陥が発見されると考えられる．

欠陥発見状況のタイプ別の品質特性のスコアの分布を図 11 ～図 15 に示す．ただし利用状況網羅性，リスク回避性，満足性については，測定評価できた製品数が限られたため省略した．品質特性別に結果と考察を以下に述べる．

(1) 機能適合性，信頼性，有効性：

- ●結果：機能適合性，信頼性，有効性について安定タイプにおいて高品質であった．
- ●考察：これは，機能適合性，信頼性，有効性が高い製品について，十分にテストされ欠陥を発見していると考えられる．

(2) 性能効率性，互換性：

- ●結果：性能効率性，互換性については爆発タイプにおいて低品質であることが分かった．
- ●考察：これは，性能効率性，互換性が低いソフトウェア製品については，テストで十分に欠陥を発見できておらず，今後も欠陥が発見される可能性が高いと考えられる．

(3) ほかの品質特性：

- ●結果：ほかの特性については，顕著な違いは見られなかった．
- ●考察：これは対象とした製品数の数が少ないことも原因と考えられる．

4.4 プロジェクトコンテキストと品質特性

製品コンテキストとしてドメイン，対象開発期間，開発形態，提供種別，機能数，プログラムソースコード行数の各情報を製品開発元から得た．製品によっては一部の情報が得られていない．コンテキストの一部と平均スコアの関係を図 16，図 17 に示す．コンテキストの種別のうちで顕著であった結果と考察を以下に述べる．

(1) ドメイン別の傾向：

- ●結果：互換性及びセキュリティについてドメインにより平均スコアに顕著な差が見られた．エンドユーザ向けサービス製品においてセキュリティは極めて高く，数値計算シミュレーション製品において低い結果となった．
- ●考察：重視される品質特性の違いに起因すると考えられる．もう一つの理由としては，具体化できた測定法が限られていたこともあげられる．

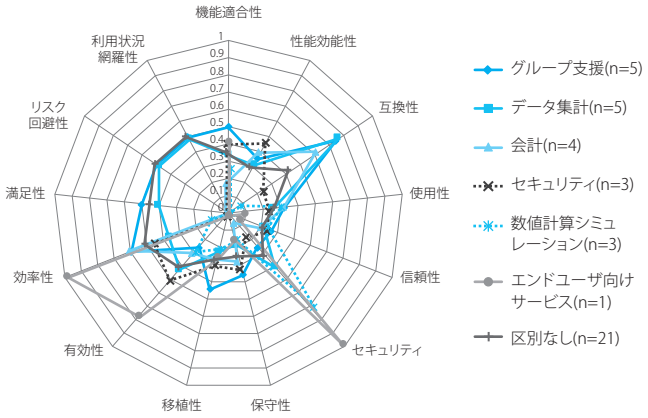


図 16　ドメイン別の品質評価スコア平均

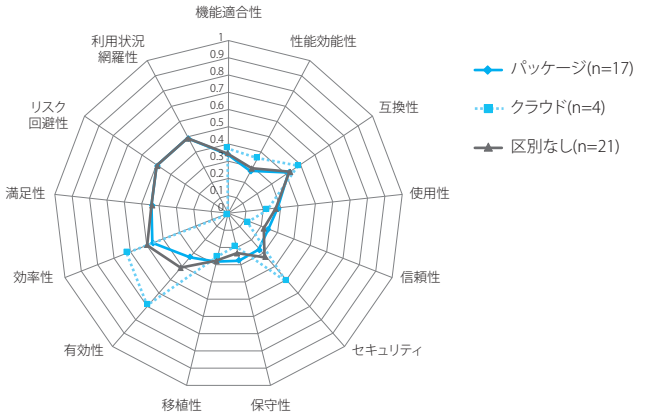


図 17　提供種別に基づく品質評価スコア平均

(2) 提供種別・パッケージ製品の傾向：

- ●結果：パッケージ製品はクラウド製品に比べてセキュリティが著しく低い結果となった．
- ●考察：ネットワーク接続や利用環境の変化を考慮する場合にセキュリティの強化が課題と言うことができる．

(3) 提供種別・クラウド製品の傾向：

- ●結果：クラウド製品についてはパッケージ製品に比べて保守性や移植性がやや低い．
- ●考察：原因として，クラウド製品におけるソフトウェアの保守及び移植においては DevOps に代表される運用体制と連携した機能拡張や，仮想化システムの標準への対応といったクラウド環境特有の考慮が必要となり，パッケージ製品とはやや異なる点が測定評価に用いた国際規格シリーズ SQuaRE において考慮されていないと推測できる．また，信頼性についてもクラウド製品はパッケージ製品に比べて低いが，これは SQuaRE において測定量の定義にあたりクラウド環境の考慮が不足している可能性がある．SQuaRE におけ

る品質測定方法の多くは 1990 年代までのソフトウェア製品の形態及び開発方法を念頭においており，アジャイル開発やプラットフォームとしてのクラウドに対する考慮を幾らか欠いており，今後の対応強化が期待される．

なお規模，期間，開発形態のそれぞれにおいては，品質に顕著な傾向の違いが見られなかった．

4.5 調査結果のまとめと提言

実現した枠組みを 21 製品へ実適用して得られた結果を以下にまとめると共に，産業界へ以下を提言する．なおこれらの提言は，調査対象の製品提供元へも伝達済みである．

- 提言 1. すべてがつながる IoT/IoE 時代に最重要となるはずの品質であるセキュリティ及び 互換性が一部低いパッケージ製品が見られた．ソフトウェアの開発プロセスや方法，取り巻く環境が変革されつつある中で，ソフトウェア品質に対する意識の変革も必要である．
- 提言 2. 機能適合性の作り込みの結果として，本来の利用者にとって重要な使用性が損なわれている可能性がある．関連して，利用時の品質である効率性と有効性について 2 極化の傾向が見られた．製品の価値を判断する立場は，本来は利用者であり，様々な品質特性を多面的に考慮するユーザ中心の取り組み（例えばユーザ中心の設計など）が今後は求められる．同様に，開発者視点の製品品質を作り込む中で，常に利用者視点の利用時の品質も併せて意識し，それらの測定評価及び改善の取り組みが今後はより一層重要である．
- 提言 3. 製品品質と利用時の品質の間の関係としては，信頼性と有効性及び移植性と効率性の間に正の相関が見られた．加えて，信頼性成長モデルにより特定される欠陥発見状況が安定タイプである場合に有効性が高く，逆に漸増及び爆発タイプであると有効性が低い傾向にあった．有効性を高めるためには，開発中において確実に摘出欠陥数を収束させることが望ましい．
- 提言 4. 21 製品の協力を得たが，品質特性単位で見ると実際に測定評価できた製品数は半数に満たないものがほとんどであった．この原因としては，各製品開発元において，様々な品質特性の観点から評価可能とするための根拠となるデータを記録していないこと，及び，SQuaRE においてもともと想定している品質測定の方法において必要な目標値がほとんど設定されていないことが挙げられる．多面的な品質測定評価を可能とするためのデータの記録，並びに，ベンチマークを参照した上での目標値設定が重要である．

4.6 制限

製品への実適用を通じて提案する枠組みを具体的に利用できることを確認した．また上述のように，企業において枠組みと品質ベンチマークの活用が始まっている [小島 17]．しかしながらそれらをもって枠組みの妥当性のすべてを明らかとしたこととはならない．枠組みにより得られる評価結果と，各製品の品質要求との対比や，枠組みが直接に扱わない製品リリース後の特徴量や実績値（例えば保守性であればリリース後の保守工数）との対比を通じて，枠組みの妥当性をより明らかとすることが今後の課題である．

測定法や品質特性によっては対象製品数や取得できたデータが少なく統計的に母数として十分な数量とは言い難いものもあることに留意が必要である．上述の分析結果はあくまでも今回取得したデータ規模における推計である．また，今回の研究はソフトウェアのすべての領域にわたっているわけではなく，パッケージなどの供給側の立場からの品質評価が主になっている点にも留意が必要である．

ユーザテストについて，上述のように実利用者に代わり，我々が異常系テスト項目を定義し，正常系テスト項目と併せて我々がテストした．対象領域やテスト項目への理解を深めながら進めており，少なくとも初心者相当の利用者の視点を確保できたと考えられる．しかしながら，現実の業務活動の環境下において実施したものではないため，幅広い利用者の視点からの様々な利用をテストできていない可能性がある．また，操作の熟練度において実利用者の平均的なものよりも劣っている可能性があり，これらは有効性及び効率性の測定評価における妥当性への脅威である．その緩和や克服に向けては，各製品の実利用者もしくは一定時間をかけて熟練した被験者を準備することが対策として考えられる．

5 おわりに

得られた枠組みと品質ベンチマークは，ソフトウェア製品の開発側や運用側において，開発・保守・運用中あるいは運用検討中のソフトウェア製品の品質を，客観的，定量的かつ総合的に評価可能とし，評価結果を開発・保守における品質要求定義や品質改善，取捨選択の判断材料に役立てられることが期待できる．実際に，企業において枠組みと品質ベンチマークの活用が始まっている [小島 17]．また我々は，具体的な測定データに基づいて産業界への提言を示した．

得られた調査結果を品質実態の調査結果として捉えようとする場合に，品質実態の限られた一端を明らかとするこ

とにつながったと考えられるが，品質特性によっては記録の不足や目標値未設定といった理由からデータが得られた製品数が限られている．そこで更なる品質実態の調査に向けて，今後データ数を増やし，継続的に関係分析結果を更新する予定である．また，データ数が一定数に達した時点で，より精緻な関係モデルの導出を試みる予定である．

A　付録

品質特性ごとに一覧を次頁以降に示す．表中の “SQuaRE” は，SQuaRE において対応する測定量を示す．対応する測定量がない場合は空欄である．“ID” は，我々が定義した各測定法に一意に付与した識別子である．

A.1 製品品質の測定法一覧

(1) 機能適合性

(副) 特性	SQuaRE	ID	名称	定義	詳細
機能完全性	FCp-1	FCp.1.1	要求実装率	X=A/B	A= 見送らず，着手・実現した要求の数 B= 対象期間内で挙がった要求の数
機能正確性	FCr-1	FCr.1.1	深刻不具合 除去率	X=1-A/B	A= そのうち，残存している不具合の数 B= 対象期間内で発見した深刻な不具合の数
機能適切性	FAp-1	FAp.1.1	システム試験数目 標達成率	X=A/B	A= 対象期間内でのテストケース実施数 B= 対象期間内でのテストケースの実施目標数
機能適切性	FAp-1	FAp.1.2	ユーザの意図に則 す度合い	X'=X のユーザ群での 平均	X= 意図に即す度合のユーザアンケート回答

(2) 性能効率性

(副) 特性	SQuaRE	ID	名称	定義	詳細
時間効率性		PTb.0.1	時間効率性の試験 実績	X=A/B	A= 応答時間，ターンアラウンドタイム，スループットのうち， 実施しているものの数 B=3
時間効率性	PTb-1	PTb.1.1	応答時間　平均	X'= タスク群での X 平均値	X= ある性能試験タスクでの実測値
時間効率性	PTb-2	PTb.2.1	応答時間　実測対 目標	X'= タスク群での X 平均値 X=A/B	A= ある性能試験タスクでの実測値 B= その性能試験タスクでの目標値
時間効率性	PTb-3	PTb.3.1	ターンアラウンド タイム　平均	X'= タスク群での X 平均値	X= ある性能試験タスクでの実測値
時間効率性	PTb-4	PTb.4.1	ターンアラウンド タイム　実測対 目標	X'= タスク群での X 平均値 X=A/B	A= ある性能試験タスクでの実測値 B= その性能試験タスクでの目標値
時間効率性	PTb-5	PTb.5.1	スループット 目標達成率	X'= タスク群での X 平均値 X=A/B	A= ある性能試験タスクでの実測値 B= その性能試験タスクでの目標値
資源効率性		PRu.0.1	資源効率性の試験 実績	X=A/B	A=2 種のうち，実施しているものの数 B=2 (CPU 使用率系試験，メモリ使用率試験)
資源効率性	PRu-1	PRu.1.1	CPU 使用率 最大値	X'= タスク群での X 平均値	X= ある性能試験タスク実施中での CPU 使用率 最大値
資源効率性	PRu-2	PRu.2.1	メモリ 使用率　最 大値	X'= タスク群での X 平均値	X= ある性能試験タスク実施中でのメモリ使用率 最大値
容量満足性		PCa.0.1	容量満足性の試験 実績	X=0 または 1	X= ユーザ同時アクセス数系の試験を実施していれば 1，さも なくば 0
容量満足性	PCa-2	PCa.2.1	ユーザ同時アクセ ス可能数　目標達 成率	X'= タスク群での X 平均値 X=A/B	A= ある性能試験タスクでの実測値 B= その性能試験タスクでの目標値

(3) 互換性

(副) 特性	SQuaRE	ID	名称	定義	詳細
共存性	CCo-1	CCo.1.1	他製品共存試験実績	X'= サーバマシン，クライアントマシンの X 平均 X=A/B	A= 試験環境で意図的に共存させているソフトウェア種別（セキュリティソフトウェア，対象製品） B=2
相互運用性	CIn-1	CIn.1.1	ファイル形式のインポート / エクスポート両対応率	X=A/B	A= インポート・エクスポート両対応のものの数 B= 取り扱っているファイル拡張子の数

(4) 使用性

(副) 特性	SQuaRE	ID	名称	定義	詳細
適切度認識性	UAp-2	UAp.2.1	機能の動画説明対応率	X=A/B	A= 説明動画が公開されている機能の数 B= 機能の数
習得性	ULe-1	ULe.1.1	機能の説明カタログ記載率	X=A/B	A= 機能のうち，カタログに載っている数 B= 機能の数
習得性	ULe-1	ULe.1.2	機能の説明マニュアル記載率	X=A/B	A= 機能のうち，マニュアルに載っている数 B= 機能の数
運用操作性	UOp-6	UOp.6.1	機能の Undo 対応率	X=A/B	A= 元に戻せる機能の数 B=Undo が必要な機能の数
ユーザエラー防止性	UEp-1	UEp.1.1	機能での入力内容チェック対応率	X=A/B	A=B のうち，エラーメッセージや警告文が出る機能の数 B= ユーザインプットを求めている機能の数
ユーザインターフェース快美性	UIn-1	UIn.1.1	UI の使いやすさ度合い	X'=X のユーザ群での平均	X=UI の使いやすさのユーザアンケート回答
アクセシビリティ	UAc-3	UAc.3.1	機能の聴覚ハンディキャップ配慮率	X=A/B	A= 機能のうち，聴覚ハンディキャップへの配慮があるものの数 B= 機能の数
アクセシビリティ	UAc-4	UAc.4.1	機能の視覚ハンディキャップ配慮率	X=A/B	A= 機能のうち，聴覚ハンディキャップへの配慮があるものの数 B= 機能の数
アクセシビリティ	UAc-5	UAc.5.1	言語の対応度合い	X=A/B	A= 各言語の対応度重み総和 B= 対応言語の数

(5) 信頼性

(副) 特性	SQuaRE	ID	名称	定義	詳細
成熟性	RMa-1	RMa.1.1	不具合除去率 （単体試験）	X=A/B	A=B のうち修正されたものの数 B= 対象期間内に発見した欠陥数 (fault)
成熟性	RMa-1	RMa.1.2	不具合除去率 （結合試験）	X=A/B	A=B のうち修正されたものの数 B= 対象期間内に発見した欠陥数 (fault)
成熟性	RMa-1	RMa.1.3	不具合除去率 （システム試験）	X=A/B	A=B のうち修正されたものの数 B= 対象期間内に発見した欠陥数 (fault)
成熟性	RMa-2	RMa.2.1	MTBF　目標 達成率	X=C/D	A= 稼働時間 B= 故障発生回数 C=MTBF 実測値 =A/B D=MTBF 目標値
成熟性	RMa-3	RMa.3.1	不具合発見率 （単体試験）	X=A/B	A= 対象期間内での不具合発見数　（実測） B= 対象期間内での不具合発見数の目標値
成熟性	RMa-3	RMa.3.2	不具合発見率 （結合試験）	X=A/B	A= 対象期間内での不具合発見数　（実測） B= 対象期間内での不具合発見数の目標値
成熟性	RMa-3	RMa.3.3	不具合発見率 （システム試験）	X=A/B	A= 対象期間内での不具合発見数　（実測） B= 対象期間内での不具合発見数の目標値
成熟性	RMa-3	RMa.3.4	不具合発見率 （チケットベース）	X=B/(B-abs(B-A))	A= 対象期間内での不具合発見数　（実測） B= 対象期間内での不具合発見数の予測値 abs= 絶対値関数 ※ B を信頼性曲線を元に算出
成熟性	RMa-4	RMa.4.2	試験実施率 （システム試験）	X=A/B	A= 対象期間内でのテストケース実施数 B= 対象期間内でのテストケースの実施目標数
可用性		RAv.0.1	運用試験実績	X=0 または 1	運用試験を実施していれば 1，さもなくなば 0
可用性	RAv-1	RAv.1.1	運用実時間　対 規定時間	X=A/B	A= 継続運用時，実際に製品が正常稼働できた時間 B= 継続運用時，製品が正常稼働し続けられる時間の予想値
可用性	RAv-2	RAv.2.1	システムダウン 時間　実際対 目標	X=(A/B)/C	A= システムダウンしていた時間合計 B= システムダウン回数 C= システムダウン時間平均の目標値
障害許容性	RFt-1	RFt.1.1	fault-pattern テスト ケース (結合試験)	X=A/B	A=B のうち，成功した数 B= 対象期間内での fault Pattern テストケース数
回復性	RRe-1	RRe.1.1	システムダウン 回復時間　実際 対目標	X=(A/B)/C	A= システムダウン回復にかかった時間合計 B= システムダウン回数 C= システム回復時間平均の目標値

(6) セキュリティ

(副) 特性	SQuaRE	ID	名称	定義	詳細
機密性	SCo-1	SCo.1.1	データのアクセス権限管理対応率	X=A/B	A=B のうち，アクセス権限管理が可能なものの数 B= データ種別の数
機密性	SCo-2	SCo.2.1	データの暗号化対応率	X=A/B	A=B のうち，暗号化されているものの数 B= データ種別の数
インテグリティ	SIn-2	SIn.2.1	データの破損防止策対応率	X=A/B	A=B のうち，破損防止機能のあるものの数 B= データ種別の数
否認防止性	SNo-1	SNo.1.1	ネットワーク経路のデジタル署名対応率	X=A/B	A=B のうち，デジタル署名有効なものの数 B= 利用している通信経路の種別数
責任追跡性	SAc-1	SAc.1.1	データのアクセスログ対応率	X=A/B	A=B のうち，アクセス履歴がログに残るものの数 B= データ種別の数
真正性	SAu-1	SAu.1.1	ログイン機能での認証方式対応率	X'= ログイン機能についての X X=A/B	A= 製品でサポートしている認証方式の種別数 B=6 種：固定パスワード，ワンタイムパスワード，期限付きパスワード，物理的トークン，生体認証，解読型

(7) 保守性

(副) 特性	SQuaRE	ID	名称	定義	詳細
モジュール性	MMo-1	MMo.1.1	クラスの結合度	X'= クラス群での X 平均	X= クラス結合度（クラス単位）
モジュール性	MMo-2	MMo.2.1	関数のサイクロマティック複雑度	X'= 関数」群での X 平均	X= サイクロマティック複雑度（関数単位）
再利用性	MRe-1	MRe.1.1	クラスの凝集性の欠如	X'= クラス群での X 平均	X=100-LCOM2（クラス単位） LCOM2= 集性の欠如の定義 ver2 ※ここでは LCOM2 が 0 ～ 100 の値
解析性	MAn-1	MAn.1.1	データのアクセスログ　対応率	X=A/B	A=B のうち，アクセス履歴がログに残るものの数 B= データ種別の数
修正性	MMd-3	MMd.3.1	不具合除去率（単体試験で発見分）	X=A/B	A= 修正済み不具合数 B= 発見済み不具合数
修正性	MMd-3	MMd.3.2	不具合除去率（結合試験で発見分）	X=A/B	A= 修正済み不具合数 B= 発見済み不具合数
修正性	MMd-3	MMd.3.3	不具合除去率（システム試験で発見分）	X=A/B	A= 修正済み不具合数 B= 発見済み不具合数
試験性	MTe-1	MTe.1.1	モジュールの単体試験　実施率	X=A/B	A= 単体試験実施済みモジュール数 B= 全モジュール数

(8) 移植性

(副) 特性	SQuaRE	ID	名称	定義	詳細
適応性		PAd.0.1	複数環境の試験実現	X1=A/B X2=C/B	A= 複数環境で試験済みの主機能数 B= 主機能数 C= 複数環境で試験成功した主機能数
設置性		PIn.0.1	インストールの試験実現	X=0 または 1	インストール試験を実施していれば 1、さもなくば 0
設置性	PIn-1	PIn.1.1	インストール時間　平均	X'= タスク群での X 平均 X=A/B （インストール時間）	A= あるタスクでの実測値 B= あるタスクでの目標値
設置性	PIn-2	PIn.2.1	インストーラ提供形態　対応率	X=A/B	A=3 種の内，対応している製品提供形態 B=3 ※種別：　Web，CD，セットアップ代行
設置性	PIn-2	PIn.2.2	インストールオプション対応率	X'= サーバソフトでの X X=A/B	A= 以下の 8 種のうち，対応しているインストールオプション B=8 ・アンインストール時に，あらゆる設定情報を消去できるか ・アンインストール時に，利用者が希望すれば設定情報をレジストリに残せるか ・アンインストール時に，利用者が希望すれば設定情報をレジストリ以外の場所に残せるか ・サーバについて複数台構成にできるか ・サーバについて一台編成にできるか ・root フォルダのインストール先フォルダの変更可能か ・インストーラを途中で中断できるか（最初からやり直しにならずに） ・事前にインスト―ルしておくべきソフトや環境がない状態でインストールしようとしたとき，それ以上先に進めなかったり警告したりしてくれるか
置換性	PRe-1	PRe.1.1	追加学習必要度合い	X'=X のユーザ群での平均	X= 満足度のユーザアンケート回答

A.2 利用時の品質の測定法一覧

(副) 特性	SQuaRE	ID	名称	定義	詳細
有効性	Ef-1	Ef.1.1	タスク完了率	X=A/B	A= 完了したタスク数 B= 全タスク数
有効性	Ef-3	Ef.3.1	タスク当たりエラー数	X=A/B	A= 全エラー数 B= 全タスク数
有効性	Ef-4	Ef.4.1	エラーが発生したタスクの率	X=A/B	A= エラーがあったタスク数 B= 全タスク数
有効性	Ef-5	Ef.5.1	エラーを起こした被験者の率	X=A/B	A= エラーを起こした人数 B= 被験者人数
効率性	Ey-1	Ey.1.1	タスクにかかった時間の平均	X'= タスク群での X 平均	A= タスクの開始時刻 B= タスクの終了時刻
効率性	Ey-5	Ey.5.1	タスク中の総アクションの無駄でないアクションの率	X'= タスク群での X 平均	A= 必要アクション数 B= 全アクション数
実用性	SUs-1	SUs.1.1	製品に対する満足度	X'=X のユーザ群での平均	X= 満足度のユーザアンケート回答
実用性	SUs-1	SUs.1.2	ネットプロモータースコア	X'=X のユーザ群での平均	X= ネットプロモータースコアのユーザアンケート回答
実用性	SUs-2	SUs.2.1	機能に対する満足度	X'=X のユーザ群での平均	X= 各機能満足度のユーザアンケート回答結果のユーザにおける平均
信用性	STr-1	STr.1.1	信用度合い	X'=X のユーザ群での平均	X= 信用度のユーザアンケート回答
快感性	SPl-1	SPl.1.1	快感度合い	X'=X のユーザ群での平均	X= ストレスのない利用に関するユーザアンケート回答
快適性	SPo-1	SCo.1.1	快適度合い	X'=X のユーザ群での平均	X= 快適度のユーザアンケート回答
経済リスク緩和性		REc.0.1	経済的損失の無さ	X'=X のユーザ群での平均	X= 経済的な影響の感じなさのユーザアンケート回答
健康・安全リスク緩和		RHe.0.1	健康や人命への影響の無さ	X'=X のユーザ群での平均	X= 健康や人名への影響の感じなさのユーザアンケート回答
環境リスク緩和性		REn.0.1	環境への影響の無さ	X'=X のユーザ群での平均	X= 自然・社会環境への影響の感じなさのユーザアンケート回答
利用状況完全性		CCm.0.1	主要な目的以外での製品利用	X'=X のユーザ群での平均	X= 主要な目的以外で製品を利用することのユーザアンケート回答
柔軟性		CFl.0.1	非主要目的での製品利用時タスク達成度合い	X'=X のユーザ群での平均	X= 主要な目的以外の利用時の目的達成のユーザアンケート回答

謝辞：本研究は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（SEC：Software Reliability Enhancement Center）が実施した「2015 年度ソフトウェア工学分野の先導的研究支援事業」の支援のもと行われた．調査対象製品開発元，評価機関，CSAJ ほか関係各位に謝意を記す．

【参考文献】

[東 15] 東基衛, " 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 ", SEC Journal, 10(5), 2015.

[ISO25000] ISO/IEC 25000:2014 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE

[WSQB] WSQB17: Waseda Software Quality Benchmark, http://www.washi.cs.waseda.ac.jp/?page_id=3479

[小島 17] 小島嘉津江, 森田純恵, 廣瀬竹男, 若本雅晶, 菊池慎司, 鷲崎弘宜, " ソフトウェア品質技術が品質特性に与える効果の見える化と活用の一考察 ", ソフトウェア品質シンポジウム, 2017.

[鷲崎 07] 鷲崎弘宜ほか, " プログラムソースコードのための実用的な品質評価枠組み ", 情報処理学会論文誌, 48(8), 2007.

[Bansiya02] J. Bansiya and C.G. Davis, "A Hierarchical Model for Object-Oriented Design Quality Assessment," IEEE Transactions on Software Engineering, 28(1), 2002.

[Ortega03] M. Ortega, M. Perez and T. Rojas, "Construction of A Systematic Quality Model for Evaluating A Software Product," Software Quality Journal, 11(3), 2003.

[鷲崎 10] 鷲崎弘宜, 田邉浩之, 小池利和, " ソースコード解析による品質評価の仕組み ", 日経エレクトロニクス, 2010 年 1 月 25 日号

[ISO25010] ISO/IEC 25010:2011 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models

[ISO25022] ISO/IEC 25022:2016 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Measurement of quality in use

[ISO25023] ISO/IEC 25023:2016 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Measurement of system and software product quality

[Arban03] A. Abran, et al., "Usability meanings and interpretations in ISO standards." Software Quality Journal, 11(4), 2003.

[Heidrich14] J. Heidrich, et al., "Model-based quality management of software development projects," Software Project Management in a Changing World, 2014.

[Biscoglio14] I. Biscoglio and E. Marchetti, "Definition of Software Quality Evaluation and Measurement Plans: A Reported Experience Inside the Audio-Visual Preservation Context," 9th International Joint Conference on Software Technologies (ICSOFT), 2014.

[ISO25051] ISO/IEC 25051:2014 Software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing

[IPA14] IPA, " 製品・システムにおけるソフトウェアの信頼性・安全性等に関する品質説明力強化のための制度構築ガイドライン ", 2014.

[PSQ] CSAJ, PSQ 認証制度 , http://www.psq-japan.com/

[IPA15] IPA/SEC, " ソフトウェア開発データ白書 2014-2015", 2015.

[Cusumano03] M. Cusumano, et al., "Software Development Worldwide: The State of the Practice," IEEE Software, 20(6), 2003.

[Jones09] C. Jones, "Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies," McGraw-Hill, 2009.

[鷲崎 16] 鷲崎弘宜, " 実践的ソフトウェア品質測定評価のための 4 つの「落とし穴」と 7 つの「コツ」", 品質, 46(3), 2016.

[Washizaki17] H. Washizaki, "Pitfalls and Countermeasures in Software Quality Measurements and Evaluations," Advances in Computers, 106, Elsevier, 2017.

[Basili94] V. Basili, et al., "Goal, Question, Metric Paradigm," Encyclopedia of Software Engineering, 1, 1994.

[Honda17] K. Honda, et al., "Generalized Software Reliability Model Considering Uncertainty and Dynamics: Model and Applications," Int'l J. Software Engineering and Knowledge Engineering, 27(6), 2017.

[Alves10] T. L. Alves, et al., "Deriving metric thresholds from benchmark data," IEEE International Conference on Software Maintenance (ICSM), 2010.

FRAM（機能共鳴分析手法）による成功学に基づく安全工学


野本 秀樹※1


道浦 康貴※1


石濱 直樹※2


片平 真史※2

レジリエンス・エンジニアリングのモデリング手法である，FRAM（Functional Resonance Analysis Method：機能共鳴分析手法）を使用し，実製品開発で安全分析を行うための方法及び手順，とくにモデリング方法と評価方法を構築した．FRAM を使った安全分析では，従来のリスクや失敗原因を追及するだけではなく，反対に成功要因を分析する．成功学に基づく安全分析で具体的に着目すべき観点をまとめた．

New Safety Engineering using FRAM (Functional Resonance Analysis Method)

Hideki Nomoto※1, Yasutaka Michiura※1, Naoki Ishihama※2, Masafumi Katahira※2

This paper describes modeling and analysis methodology and procedure using FRAM (Functional Resonance Analysis Method). FRAM is the methodology to realize safety analysis based on resilience engineering. The analysis will be performed based not on failure but on success. Required perspectives to conduct success-based safety analysis will be shown.

1　はじめに

FRAM(Functional Resonance Analysis Method: 機能共鳴分析）は，レジリエンス・エンジニアリングにおける安全分析のための手法である [Hollnagel 2013]．機能共鳴とは，複数の機能が相互にインタラクションした結果外乱に柔軟に対応する一方で，逆にエスカレーションなどを起こし，安全を脅かすことを指す．つまり，FRAM における安全分析とは，複数の機能が互いにどのようにインタラクションするのかを明らかにし，その関係性の中に安全にかかわるシステムの長所や短所を見出すことであると言うことができる．本稿では，FRAM の分析の考え方を述べた後に，具体的な分析手法を記述する．

2　従来の安全工学との違い

2.1 FRAM の特徴

従来の安全解析は，ハザードの発生など，システムが失敗する事象を定義し，その原因を分析していくものである [FAA]．これに対して，FRAM はシステムの失敗事象を何ら定義せずに分析を行うことに大きな特徴がある．これは，安全に対する思想の相違から生まれる特徴である．FRAM

※1　有人宇宙システム株式会社　Japan Manned Space Systems Corporation
※2　研究開発法人宇宙航空研究開発機構　Japan Aerospace Exploration Agency

を生んだ安全工学をレジリエンス・エンジニアリングと呼ぶ．レジリエンス・エンジニアリングは，システムの安全は環境変動や意図しない入力に対する柔軟さによって達成されると考える [Hollnagel 2012][Holling 1973]．一方，その柔軟さが逆に意図しないシステム挙動を生む可能性もある．硬質のボール同士の衝突によるその後の挙動は予測可能であるが，柔軟なアメーバ同士の衝突に続いて次に何か起こるのかは予測が難しい．

つまり，入力の変動に対して柔軟に変動できる能力の高さが安全性の高さ（成功要因）であると同時に，変動することそのものは安全のリスク（失敗要因）でもあると言うことができる．これを，レジリエンス・エンジニアリングでは，「失敗と成功の同義性」と呼んでいる [Hollnagel 2016]．すなわち，安全を脅かす要因は，「失敗」や「故障」だけではなく，「成功」も安全を脅かす原因となる．

2.2 従来の安全解析手法との比較

従来の安全解析手法では，FTA(Fault Tree Analysis) のように，ハザードの原因を，明示的に「故障」に求める [FAA]．また，STAMP/STPA(System Theoretic Accident Model and Process) のように，コンポーネント間のインタラクションが「遅れる」「間違う」「途中で止まる」「提供されない」など，故障以外の要因を求める技術もある [Leveson 2012]．いずれにしても，これらの技術は，物事の「失敗」に着目するものである．

従来の安全工学とのもう一つの大きな違いは，分析がボトムアップに行われる性質にある．FTA はハザードをトップ事象として，その原因を詳細に分解していく．STAMP も同様にハザードをトップ事象とし，ハザード制御がどのように破たんし得るのかをトップダウンに詳細化していく．一方 FRAM は，まず個々の機能の詳細な定義から始め，分析の結果として全体ネットワークがモデル化され，システムの成功要因が導出される．従来の手法が演繹法であり，「失敗に基づく分解」を行っているのに対して，FRAM は帰納法であり，「成功に基づく統合」を指向している．

FRAM では，上述したように，最初のステップで個々の機能の詳細な定義を行う．機能の変動につながる表 1 に示す 6 つの要素を分析する．

2.3 制御システムの分析例

ここで，人工衛星の姿勢制御に関する分析を考える．従来の安全解析では，例えば，姿勢レートセンサと姿勢制御装置とのインタラクションを分析する際，姿勢レートセンサから出力される姿勢レート値が届かない場合，姿勢レートが間違っていた場合などのケース分けをして影響評価を行っていく．その多くは，異常処置の内容を精査することにつながる．

表 1　機能の 6 要素

I: 入力	機能の動作トリガー
P: 前提	機能が動作開始するための事前条件
C: 制御	機能の挙動方法を操作する事後条件
R: 資源	機能の動作に必要な資源（事後条件）
T: 時間	機能実行可能時間（事後条件）
O: 出力	機能の出力

一方，姿勢制御機能と姿勢レート測定機能がインタラクションするという関係性を FRAM により分析する場合，やり取りされるデータの変化よりも，機能そのものの変化に着目する．すなわち，姿勢制御機能がその機能を変えるのは何によってであるのかを分析する．姿勢制御機能がその制御方法を変更するのは，入力されるセンサデータの有効フラグの値が変わったとき（前提条件），周期処理スロット内にデータが入力されたか否か（時間），小さなデッドバンド幅で制御を行う精姿勢制御モードと大きなデッドバンド幅で制御を行う粗姿勢制御モードが入れ替わったとき（制御）など，とくに入力の異常に限定せず（と言うよりも，むしろ大部分が正常ケース）識別していく必要がある．

このような，機能間インタラクションのモデリングを行う場合は，通常のインターフェース仕様に定義される「データフォーマット」など，シンタクスを規定するのではなく，「前提条件」「制御」「資源」といったセマンティクスを規定することになる．それにより，単純にデータが来ない・間違う・遅れるなどの異常系の評価を指向せず，正常・異常を全く分け隔てすることなく，機能間の関係を定義する．

現代的システムでは，ダイナミックに機能同士が通信し，互いに制御し合う分散的な関係を持つアーキテクチャが主流になりつつある [Ding 2007][Aoki 2014]．このようなカップリングを多用するシステムにおいては，ある機能とある機能が極めてまれな条件の組み合わせのときのみ，危険な相互作用をするというような，検出しにくい問題の方が重大な問題を引き起こしやすい傾向を持つと考えられる．なぜなら，まれにしか発生しない問題は，そのような問題が存在すること自体が認識されていないことが多いからである．

発見しがたいカップリングを発見するためには，カップリングの様子を詳細に可視化する手法が必要となる．しか

も，そのモデルには，依存関係，タイミング，制御・被制御関係，制約・被制約関係など，様々な関係を一挙に俯瞰できることが求められる．なぜなら，単純な制御・被制御の関係だけから問題が発生することはまれだが，例えば「制御する側が制御される側よりも遅いタイミングで起動される」というような複数の側面の特殊なカップリング関係こそが発見しがたいまれなカップリングになり得るからである．そのようなカップリングを見出すためには，タイミング・モデルと制御・被制御モデルを別々に作る手法よりも，一つのモデルにそれらをすべて取り込んだモデルが効果的となるであろう．なぜなら，モデルを使った分析とは一種のひらめきを喚起するための道具の使用法であるが，ひらめきとは，複数の認知リソースの同時活性化によって生まれるとされているからである [Suzuki 2009].

上述したように，モデルを見てカップリング挙動をイメージするためには，機能間のつながり方の情報量の豊かさが重要となる．そのためには，機能と機能との結び付き方は，単純な矢印 1 本で描画されるのではなく，結合の意味（セマンティクス）を伴うモデルの作成が必要となる．FRAM が機能間の接続子として，「入力」「前提条件」「制御」「資源」「時間」「出力」の 6 つの意味的接続子を使っているのはそのためである．図 1 には，全く同一の結び付きを，STAMP の Control Structure Diagram (CSD) 図と，FRAM モデル図とで表した．FRAM のモデルが STAMP のモデルに比べて情報量が多いことを示すため，それぞれの機能やコンポーネントの意味は伏せたままで，それぞれがそのままでどこまで評価可能なのかを以下に示す．

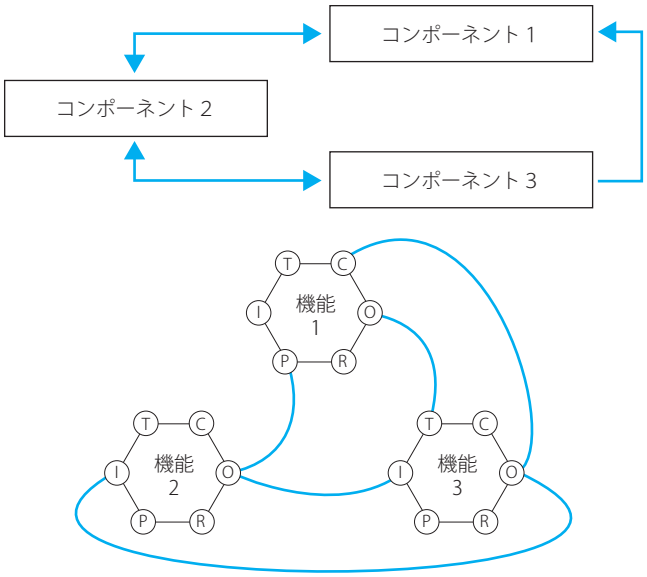


図 1　STAMP モデルと FRAM モデル

FRAM モデルを見ると，機能 2 と機能 3 との間にフィードバックループの関係があり，そこに機能 1 が割り込むような形で時間制約を与えている．その時間制約が行われるか否かは，機能 2 の出力が前提条件となって決定される．かつ，その時間制約は機能 3 の出力によって内容が変動するため，不意なタイミングで機能 3 が処理時間超過を宣告されるようなリスクがあることを読み取ることができる．

機能 2 と機能 3 との間に定常的に存在しているフィードバックループは無限ループの形を取っている（入力と出力が双方に接続）ため，基本的には安定したループであるが，機能 1 の存在によって，その安定した関係に変化が生じる．そして，その変化を生み出しているのは，機能 2 及び機能 3 自身の出力である．つまり，機能 1 と機能 2，そして機能 3 の間には，非常にダイナミックな共鳴関係が築かれているということになる．

上述したような評価を可能としているのは，FRAM モデル特有の 6 つの要素という情報量の多さである．図としてのシンプルさとは裏腹に，極めて多くを語るモデル化手法と言うことができる．一方，図 1 の STAMP モデルからは，モデルが持っている情報量が FRAM モデルと比較して少ないため，上記のような分析を導くことは難しい．STAMP モデルに描かれているインタラクションはやり取りされるデータにラベルが付けられていなければ理解することはできない．そして，インタラクションのタイミングに関しては，たとえラベルを付けられていても，明らかにすることができない．タイミングに関する情報量はほぼゼロとなっている．

相互影響のありようをモデル化した後，その相互影響のありようにどのような変動要素があるか否かを分析し．次に，そのような変動要素は，本質的に危険な変動なのか，それとも安全な変動なのかを最後に評価する．この一連の手順は自明なものではない．本研究では，この評価手順を 3 節に示すように構築した．

3　FRAM分析手順

提案する FRAM による分析手順は以下の 2 つステップから成る．

(1)　モデリング手順

(2)　評価手順

3.1　FRAM モデリング手順

FRAM モデリング手順は，以下の順序で実施する：

(1)　質問による機能の把握

(2)　各機能の定義

(3) 可視化

(4) 可視化したモデルを使った分析

FRAM のモデリング手順は，分析対象システムの設計者，あるいは，当事者などに対するインタビューを通じて，システムの中の重要「機能」の特徴を明らかにするところから始める．その手順を 3.1.1 項に示す．

次に，質問の答えをもとに，各機能の入出力を定義する．これがモデル化作業である．その手順を 3.1.2 項に示す．

各機能の入出力データ名が明確になれば，それをツールに入力することによって可視化が完成する．その手順を 3.1.3 項に示す．

最後に，可視化されたツールを俯瞰しつつ評価を行う．評価の詳細は，3.2 項「モデルの評価」に示すが，その全体像を 3.1.4 項に示す．

3.1.1 質問による機能の把握

解析対象を俯瞰して，最も重要な機能というものを把握する．FRAM は機能間の関係に着目するので，最も重要な機能を決めたら，そこからつながりのある機能を数珠つながりにモデリングしていく．その際，重要な機能とほかの機能との結び付きをいきなりモデリングしようとしてしまうとうまくいかない．なぜなら，ある機能がある機能に対して「制御」を提供しているのか，「前提条件」を提供しているのか，などという問題は，単純ではないからである．そのような微妙な作業を実施するためには，様々な関連情報を揃えておく必要がある．そこで，まずは機能の概要を把握するために以下の質問を行う．

(1) その機能の目的は何か？

(2) 機能はどのような処理を行っているか？

(3) 機能にはどのような入出力が存在するか？

続いて，FRAM のインタビュー技法 [Hollnagel 2013] に基づき，以下の質問を行い機能の詳細を把握する．

(1) その機能の開始トリガー（入力）は何か？（I に関する質問）

(2) 条件が変わった場合，どのように適応するか？

(3) オフノミナル条件にどう反応するか？

(4) リソースは安定的に供給されるか？不安定要因は？（R に関する質問）

(5) 外部環境はどのくらい安定？不安定要因は？

(6) オフノミナル条件はたびたび発生？

(7) 「当然」と思われている前提条件はあるか？（P に関する質問）

(8) 時間制約によるプレッシャーはどこにかかるか？（T に関する質問）

(9) 特別なスキル，特別な高機能，特別な高信頼性を必要とする個所は？

(10) 最適な実行方法というものが存在しているか？（C に関する質問）

FRAM においては，ハザード解析のように，リスクパターンを網羅的に分析するのではなく，機能の 6 要素を網羅的に分析する．この網羅性によって，機能間のインタラクションの見落としがないことが保証され，システムの特徴を漏れなくモデル化することができる．システムの特徴とは，そのシステムが成功できる理由そのものである．

3.1.2 各機能の定義

FRAM の 6 つの機能要素は，表 1 に示したように，1 つの出力と 5 つの入力で構成されている．これらの要素は，3.1.1 項の質問を参考にして，重要な機能から順に定義していく．

定義に必要なのものは，各要素の名前（やり取りされる情報・データ・もの・締め切り時間）と相手の機能の名前である．

3.1.3 モデルの可視化

モデルの可視化は FRAM Visualizer[FMV] を使って行う．ツールには，3.1.2 の機能の定義を入力すれば，図 1 のような機能カップリングの図が描画される．

機能を示す 6 角形がグレーになっているものは，バックグラウンド機能と呼ばれる変動せずに安定的にデータを定期的に出力する，あるいは，出力データそのものに大きな変動が存在しないため，後続の機能に対する変動要因とならず，後続機能のモデル化の必要がないと判断されたような機能を指す．FRAM の解析は，機能の変動に着目するため，これらのバックグラウンド機能は，FRAM モデルの「外縁」に相当する．注目する機能からバックグラウンド機能までが FRAM 分析の対象となる．言い換えると，FRAM においては，分析の結果，分析対象が後付けで決まる．これにより，分析対象から外れる「想定外のインタラクション」を最小化できる．分析範囲を前もって想定しないため，原理的に想定外というものが存在し得ないからである．

3.1.4 モデルを使った分析方法

3.1.1 項に示したインタビュー技法を使い，機能の特徴を十分に引き出すことができれば，次は，システムの成功要因とリスク要因を分析する．

分析は必ず以下の順序で行う：

(1) このシステム（モデル化した範囲全体）の成功要因は何か？

(2) このシステムのリスク要因は何か？

モデル全体を俯瞰して分析を行う場合，必ずシステムの成功要因を分析するところから始める．レジリエンス・エンジニアリングは，リスクの排除によるシステム構築を目指すのではなく，成功要因を識別し，それを育てると共に，成功要因の実現を阻むリスクを抽出することを目指す．

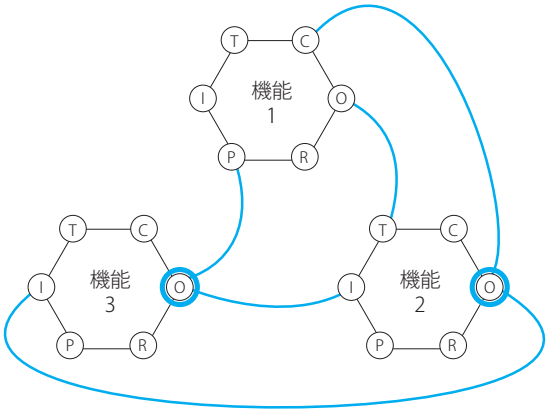


図 3.1.4-1　簡単な FRAM モデル

システムの成功要因は，幾つかの特徴的な点に現れる．例えば，上記の例では，機能 2 や機能 3 からの放射線状の出力や，機能 3 と機能 2 の間のループ構造，機能 1 と機能 2 の間のループ構造，そして，機能 3 から機能 2・機能 1 を経由して機能 3 に帰る大ループ構造など，特徴的な点が幾つかある．成功要因分析では，まず，これらの特徴的な個所を使い，システムの成功要因を明文化する作業を行う．一つの成功要因として，機能 3 と機能 2 のループに着目すると，この 2 つの機能は，通常はシーケンシャルに処理を行っている．したがって，機能 1 からの時間制約が入っても，それによって機能 3 と機能 2 の処理順序が逆転するような並列処理のリスクはなく，常に順序は守られるという特徴を持っている．これが成功要因の一つである．

次に，リスクの分析作業を行う．例えば，上述した成功要因（必ず順序が守られる）が得られると，順序が守られるため発生する問題点が見えてくる．システムは厳密な時間制約によって動いている．その制約を満たせない場合は，機能 2 は処理をストップする．機能 2 が処理をストップすると，機能 3 は開始トリガー（入力）を失うため，システム全体がストップすることになる．通常は，そのように耐性のないシステムは作られないため，機能 3 はタイムアウトを検知して次の周期の処理を実行するであろう．しかし，ここでは機能 2 からの開始トリガー（入力）がないため，通常の処理とは異なる出力が行われる可能性がある．もし機能 3 の出力が異常なジャンプをした場合，それを入力する機能 2 に何が起こるのかは，重要なリスク要因となる．

以上のように，成功要因の分析からリスク要因の分析につなげることにより，レジリエンス・エンジニアリングの思想に準拠した活動が可能となる．3.2 項では，この分析を効果的に進めるための，モデルのパターンごとの具体的な評価手順を示す．

3.2 モデルの評価

本項では，FRAM モデルの評価方法について，その手順を示す．なお，この方法は，ウッズらによる複雑なマルチスレッドシステムの典型的なパターン分析 [Woods 2006] を参考に構築した．

モデルの評価として，機能と機能の結び付き方のパターン（ネットワークトポロジー）に着目した評価方法を以下に示す．ウッズらは，複雑なシステムが示すパターンとして，Tempo（速度変動），Escalation（相互増強），Coupling（相互依存），Reframing（構造変動），Dilemmas（相互干渉）などを挙げ，システムの持つダイナミクスが当初の形態から次第に変化していく典型的な形態を，以下のように示した．

- Tempo はインタラクションの速度の変化
- Escalation はインタラクションの強度の増大
- Dilemmas はインタラクションの強度の減少
- Coupling は，複数のインタラクション間の依存
- Reframing はインタラクション構造の再構築

こうしたインタラクションの変化や複数のインタラクション間の依存性のような，複雑な問題を分析するために，FRAM のモデルをネットワークトポロジー的な特徴で分類し，カテゴリごとに，評価の観点を考察した．

3.2.1 やや複雑な機能共鳴の評価

非常に頻繁に見られ，かつ，複雑なネットワークの中でも比較的単純なパターンとして，下図 3.2-1-1 のような，同一種類の複数入力がある．ネットワークトポロジー的には，ツリー型に属する．

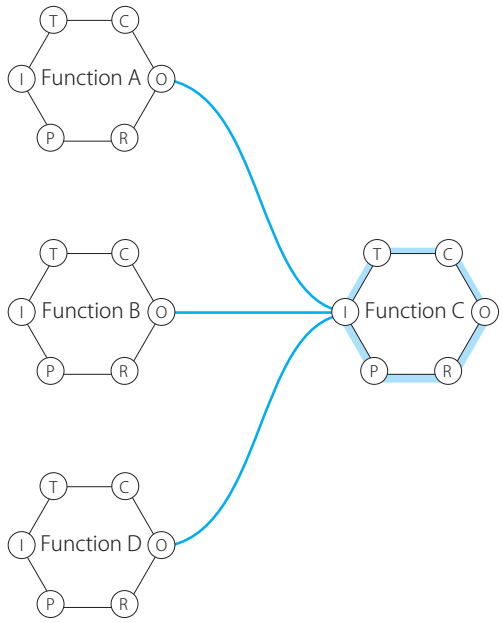


図 3.2.1-1　ツリー型

(1)　入力

- 複数の入力がある場合，どちらかに優先度はあるか？
- 複数の入力がある場合，想定される入力順序はあるか？
- 優先順位，入力順序が狂った場合，何が起こるか？

(2)　前提条件

- すべての前提条件が常に完璧に揃うことが想定されているか？
- すべての前提条件が揃わなかったら何が起こるか？
- すべての情報が揃わず，有効期限切れのデータを使ってしまうと何が起こるか？

(3)　制御

- 複数の制御が入力される場合，互いにエスカレートし合うか？
- 複数の制御が入力される場合，互いに相殺し合うか？

(4)　リソース

- リソース供給が不安定時に，出力はどう変わるか？

(5)　時間

- 時間が不足すると出力はどう変わるか？

(6)　出力

- 出力先から何かを入力している場合，エスカレートし合うか？
- 出力先から何かを入力している場合，相殺し合うか？

3.2.2　複雑な機能共鳴の評価

前項よりも，複雑な機能結合には，幾つかのパターンが存在する．以下，接続パターンごとに評価方法を示す．

(1) 放射線状の出力（出力が複数の相手先にある：スター型）

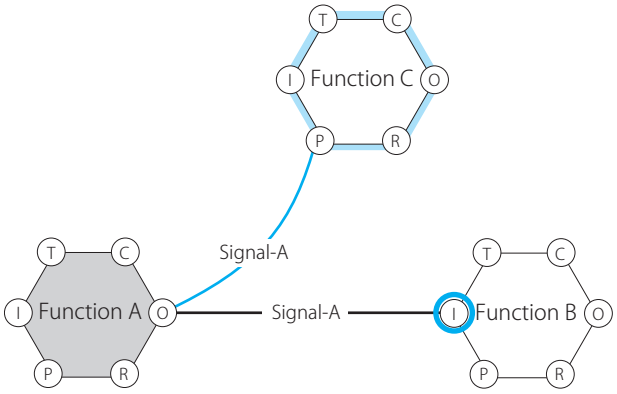


図 3.2.2-1　スター型

この場合，同じ Signal-A であっても，利用されるタイミングが異なる．Function B は，Signal-A 入力と同時に機能を実行し，Function-C は Signal-A が入力されても即座に実行せず，機能開始トリガーの入力待ちを続ける．つまり，Signal-A の内容は，Function-C が使用するときには古い情報になる可能性がある．また，入力タイミングのズレがあるということは，定常状態とデータの不安定な状態とで，システム挙動が変わることを意味する．データの不安定時には，タイミングのズレが Signal-A の利用される時間的ズレを増幅する．

(2) ループ構造（出力先からのフィードバックが返ってくる：リング型）

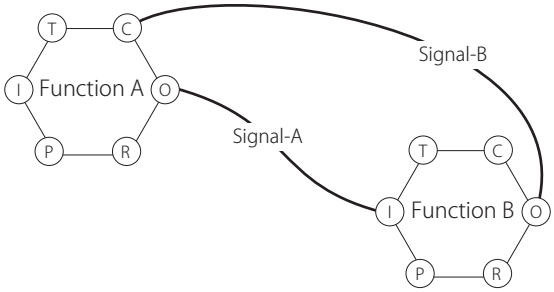


図 3.2.2-2　リング型

この場合，Function A と Function B は，互いに共鳴により強め合う，または弱め合う，もしくは，その複合という関係になることがある．

上図のようにフィードバックの帰り先が「制御」である場合は，いわゆるフィードバック制御を行うので，上記で言う複合関係となる．

下図のように，制御と制御が結合する場合，制御同士の競合，またはエスカレーションを防止するために，中間バッ

ファ的な機能（下図 Function-C）を作る必要があることがある．

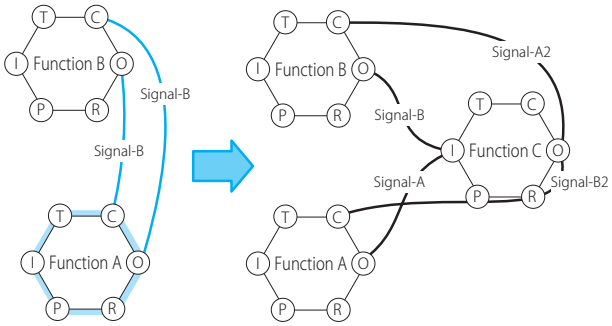


図 3.2.2-3　制御同士の競合を防止するための中間バッファの追加

(3) 入り組んだ親子関係（親が孫の子供になるような関係：メッシュ型／フルコネクト型）

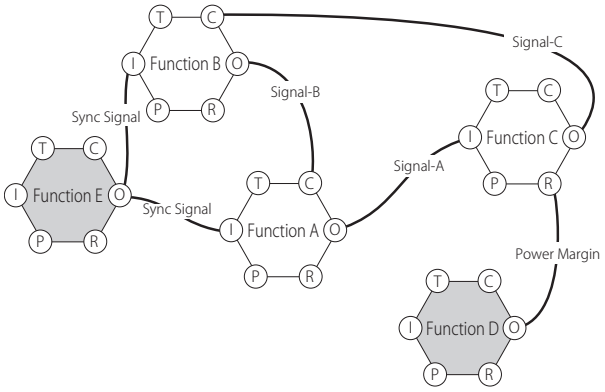


図 3.2.2-4　メッシュ型／フルコネクト型

上図では，Function-B が Function-A の親になっているが，Function-A の子供である Function-C は Function-B の親になっている．

上図の例だと，Function-A と Function-B は Function-E からの同期信号で起動しているので，実行開始は正確に同時であるが，Function-A が処理完了前に Signal-B が Function-A に届くか否かによって，Signal-A の内容は大きく変わる．Function-A と Function-B の実行時間が可変ならば，このシステムはタイミング上の変動要因を持っていることになる．そのとき，Function-C からフィードバックされてくる Signal-C は，Function-B にとっては前周期の自分の出力が反映された結果なのか，それとも前々回の出力が反映された結果なのかが認識できないため，ここを暗黙的に「前回値の反映結果」として使うと共鳴挙動が意図しないものになる可能性がある．

(4) 入り組んだ親子関係（ダイナミックに変わる関係）

あるシーンでは親，あるシーンでは子供というように，シーンによって親子関係が変わるパターンがある．下図の例だと，Power margin が不足したとき，Function-C は強権を発動して Function-B を機能制限するような制御を行うことがある．このシーンにおいては Function-C は Function-B の親である．しかし，通常時においては，Function-C は Function-B の孫に過ぎない．

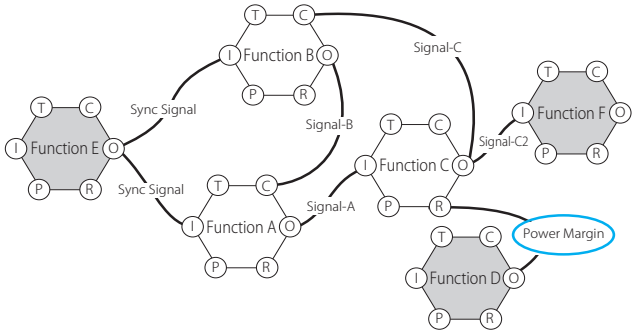


図 3.2.2-5　ダイナミックに変わる入り組んだ親子関係

上記のような例も人間組織や分散システムでよく見られるパターンである．手術室の医療チームの例として，普段は淡々と助手としてサポートを行っている看護師が，手術後の残留物確認の際に，全数チェックの結果 NG であれば縫合延期のトリガーを与えるような機能を持つ例がある．この場合，看護師の提供する機能が安全上のキーポイントになっており，これが同時にリスク源ともなる．かけ持ちで複数の手術が平行しているときなど，この全数チェック機能が最適なタイミングで行えずに医療事故に至った例がある．

また，子機能が親機能の動作を制限するような関係は，構造化プログラミングのアーキテクチャでは原理上発生することはまれだが，分散システムでは十分あり得る．外乱に対して多数のエンティティが並列的・分散的に反応することのできるシステムは柔軟性が高く，自然界の生物のようにロバストなものに進化することが可能であるが，挙動の変動が大きく，変動の条件の組み合わせが無数に存在するため，人工的なシステムとして構築することには困難が伴う．

(5) 三体問題

３つ以上のループが組み合わされることにより，これまで述べてきたような２つのループが接続されているような関係が更に複雑化する要因となる．例えば，下図のような構成は，青のループと灰色のループに，点線のループが絡むことによって，共鳴的な影響が増幅される．

青のループと灰色のループは，途中 Func2 を接点にして枝分かれするが，最終的には，同じ場所に返る．つまり，

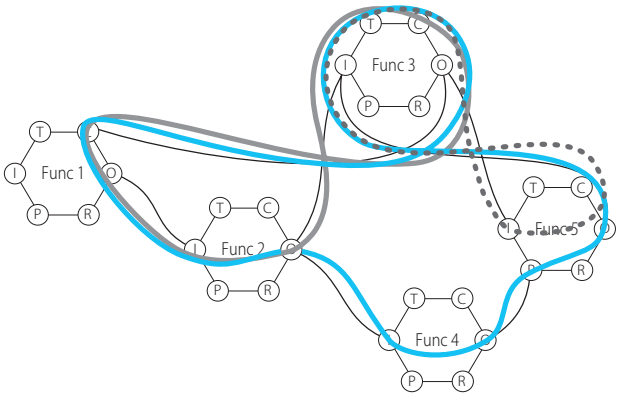


図 3.2.2-6　三体問題

途中青のループが回り道をすることになる．回り道をしている間は，別途時間がかかるが，Func5 への入力が Func5 の前提条件になっているため，青のループが灰色のループと同じタイミングで実行されるか否かは，ひとえに，点線ループが灰色ループと同じタイミングで周期運動するか否かにかかってくる．

つまり，灰色と青のループが同期を取って動作できるかどうかは，点線の実行タイミングによって決定されてしまう．点線の実行周波数が灰色の 1/10 であれば，自動的に青の実行周波数も 1/10 となるという強力な依存関係が存在している．つまり，灰色と青との関係は，灰色と青とのみからでは評価できず，むしろ第三者である点線がどのように関係するかによって決定されるという仕組みになっていることが分析される．

4　課題

FRAM を使った安全性評価が真価を発揮するのは，人間組織や人工知能など，機能が流動的に変化し，環境変動に対して柔軟に対応するタイプのシステムである．これらのシステムは，高い適応能力（つまり柔軟な機能変動）により成功を実現するシステムの典型例である．しかし，そうした高い柔軟性を持つシステムにおいては，成功と失敗の同義性 [Hollnagel 2016] も高く，システムの動作を保証することが困難である．

将来セーフティ・クリティカル・システムに人工知能を搭載することも一般的になる日が来ないとは限らない．そのとき，我々はそうしたシステムの安全保証をせねばならないという課題を突き付けられることになる．自然知能や人工知能のように，その柔軟さ故に，決定論的には動作せず，100% の安全を立証することができないシステムをどのように安全保証するのかが，今後の課題となる．

5　まとめ

FRAM を使ったモデリング方法，評価方法，実例，及び今後の課題について記述した．FRAM の安全評価手法はレジリエンス・エンジニアリングを実現するためのものであり，以下の特徴を有している：

(1)　システムの失敗要因ではなく，成功要因に着目する

(2)　個別のコンポーネントやデータではなく，統合的な視点でネットワークトポロジーに着目する

つまり，「成功学に基づく統合」という特徴が FRAM の特徴である．従来の安全解析手法が「失敗学に基づく分解」であったことの完全に逆の考え方となっている．従来の失敗学に，FRAM の成功学を組み合わせることにより，大規模・複雑な現代システムの安全がより強固に実現可能となる．失敗学，成功学共にどちらが欠けてもそれは不可能であろう．

【参考文献】

[Hollnagel 2013] エリック・ホルナゲル , 社会技術システムの安全分析―FRAM ガイドブック , (2013), p.25-38, 海文堂出版

[FAA] FAA System Safety Handbook (2000), Chapter 9: Analysis Techniques

[Hollnagel 2012] Hollnagel, E. (2012) “A Tale of Two Safeties”

[Hollnagel 2016] Hollnagel, E. (2016) “Introduction to FRAM - The Four Underlying Principles”

[Holling1973] Holling, C. S. (1973) “Resilience and stability of ecological systems”, Annual Review of Ecology and Systematics

[Leveson2012] Leveson, N. (2012) “STPA Primer”

[Wang2011] Hou, C., Wang, Q. (2011), "Software Interface Failure Modes and Effect Analysis Based on UML", DOI: 10.1109/ICM.2011.375

[Ding2007] W.Ding, Integration of MEMS INS with GPS Carrier Phase Derived Velocity: A New Approach," Proc. of 20th International Technical Meeting of The Institute of Navigation, pp.2085-2093, 2007.

[Aoki 2014] 須田 義大 , 青木 啓二 (2014) 自動運転技術の開発動向と技術課題 57 巻 (2014) 11 号 p. 809-817

[Suzuki2009] Suzuki. H. (2009). “Dynamics of insight problem-solving: Its generative, redundant, and interactive Nature” In S. Watanabe, A.P. Blaisdell, L. Huber, & A. Young (Eds), Rational animals, irrational humans. Tokyo: Keio University Press.

[Woods2006] Woods. D. & Hollnagel. E (2006) “Joint cognitive systems: Patterns in Cognitive Systems Engineering”, CRC Press ISBN 0-8493-3933-2, Chapter 9

[FMV] The FUNCTIONAL RESONANCE ANALYSIS METHOD, FRAM Model Visualizer (FMV), http://functionalresonance.com/FMV/index.html

ソフトウェア品質技術が品質特性に与える効果の見える化とその検証


小島 嘉津江[※1]


森田 純恵[※2]


廣瀬 竹男[※1]


若本 雅晶[※3]


菊池 慎司[※3]


椋 晃歓[※4]


鷲崎 弘宜[※5]

ソフトウェア品質に対する要求が多面化する一方で，要求される品質特性を実現するための方法は整備されていない．この問題を解決するために，本論文ではソフトウェアに要求される品質特性とそれを実現するための品質技術の関係を明らかにする「品質ボックス」モデルを提案する．提案モデルを複数のソフトウェア製品に適用し，その妥当性についての評価結果を示す．

Evaluation of Impact of Software Quality Assurance Technique on Quality Characteristics

Katsue Kojima[※1], Sumie Morita[※2], Takeo Hirose[※1], Masaaki Wakamoto[※3], Shinji Kikuchi[※3], Akiyoshi Hando[※4], Hironori Washizaki[※5]

Since software qualities are required to be evaluated from various aspects, it is difficult to determine how to satisfy the required quality in systematic way. To solve the problem, we defined a “quality box” model representing the relation between required qualities and techniques to achieve them. Through the application of the model to some actual software products, we conducted the evaluation of the validity of the model.

※1　富士通株式会社　※2　株式会社富士通ゼネラル（2017年末まで株式会社富士通研究所在籍）
※3　株式会社富士通研究所　※4　株式会社富士通ビー・エス・シー　※5　早稲田大学

1　はじめに

Internet of Things（IoT）や人工知能（AI）など，ソフトウェアが社会に浸透するのに従い，様々な観点からソフトウェアの品質を評価することが求められるようになった．しかしながら，これらの要求品質を実現するための標準的な品質特性のモデル，品質技術の知識体系，標準的な品質測定方法の各要素は揃ってきている一方で，それらの関係は網羅的に整理されておらず，活用するプロセスも未整備である．例えば，ソフトウェア製品そのものに対する要求，利用時の視点での要求，そして，市場競争力強化への要求など，ソフトウェアに対する要求を多面的に捉えることが求められている．そこで，我々は多面化するソフトウェア品質要求をどのように実現できるかという問題に対して，品質要求と品質技術の関係を明らかにする関係性モデル「品質ボックス」を提案する．このモデルを利用することにより，ソフトウェアに対する要件から，その品質特性を実現する品質技術の選択及び，その結果を評価するメトリクスの特定を，開発計画時に行うことが可能となり，品質要件に合理的に応えることができる．筆者らはこれまでにパッケージソフトウェア製品に対する品質ボックスの有用性の評価を行ったが [小島 b2017]，本論文では，パッケージ製品とクラウドサービスの 2 製品に対して評価を行い，異なる形態で提供されるソフトウェアに対して本手法が共通的に利用できることを明らかにした．本評価において使用したメトリクスは，RISE ベンチマーク [早稲田大学 2017] 報告結果 [WSQB] に記載のものを採用した．

本稿では 2 節で関連研究，3 節で品質技術，品質特性とメトリクスを体系的に捉え，4 節で関係性モデル (品質ボックス) を提案する．5 節で品質ボックスを活用した品質確保プロセスを提案し，6 節でその有用性を検証する．最後に，7 節でまとめを行う．

2　関連研究

ソフトウェア品質を客観的に評価するための尺度に関する研究としては，国際規格 (ISO/IEC 25000 シリーズ）などがある．また，CMU/SEI においては ATAM(The Architecture Tradeoff Analysis Method) や ADD(The SEI Attribute-Driven Design) が開発されており，更に品質技術と品質特性，及びプロセスの関係性を示す提案が出ている [Gordon2005] [Firesmith2006]．また，一段細かな設計上での原則としての再利用性，柔軟性，複雑さや経験的な設計特性と品質特性とをマッピングする取り組みも多くある [Jagdish2002]．更に，アーキテクチャ手法群においては，独自の品質モデルに基づいて，種々の技術を品質特性ごとに整理している [Bass2005]．一方，品質要求からのマトリクス作成という観点で，QFD(Quality Function Deployment：品質機能展開）[山田 2016] が関係する．典型的な品質特性間の関係表や品質上の分類をまとめた書籍もある [Zhu2005]．しかしながら，これらは設計以外の様々な技術を網羅するものではない，品質要求に基づいた品質技術の使い分けの具体的なガイドになっていない，定量的でないといった問題があった．そのため，本提案手法は，品質要求に基づく効率的な品質技術の選択の難しさに関する課題を解決し得る点において優れている．

3　品質技術と品質特性とメトリクス

3.1　SQuBOK ガイドによる品質技術の体系化

ソフトウェアに求められる品質特性を実現させるためには，その品質特性とそれを実現するための品質技術の対応関係を明確にすることが重要である．これまでにもソフトウェア品質技術として，様々な技術が提案・実用化されてきている．しかし，品質技術と品質特性の対応付けでは，まず品質技術を網羅し，体系化することが必要である．この体系化において，我々は SQuBOK ガイドを参照することとした．SQuBOK ガイドは，ソフトウェア品質の基本概念，品質マネジメント，品質技術の 3 つのカテゴリから構成されている．本稿で注目するソフトウェア品質技術は，図 1 に示すように，工程に共通と個別の品質技術，使用性やセキュリティなどの専門的品質特性の品質技術の 3 つの副カテゴリに分類される．更に，15 個の知識領域がこれら 3 つの副カテゴリに分類され，個々の知識領域は，複数の副知識領域に細分化される．具体的な品質技術はトピックスと称され，全部で 142 個あり，副知識領域ごとに整理されている．なお，紙面の都合上，図 1 では，副知識領域とトピックスは一例のみ記載した．

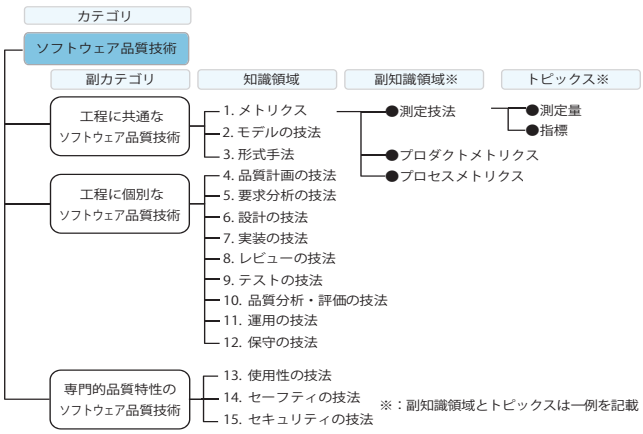


図 1　SQuBOK ガイドの樹形図概観 (品質技術)

3.2 SQuaRE における品質モデル

ソフトウェア品質に関する国際規格である ISO/IEC 25000 シリーズ (SQuaRE) では，ソフトウェア品質モデル及び品質特性を規定している．品質モデルは，以下の 4 つで構成されている．

- 利用時の品質（ISO/IEC 25010）[ISO25010]
- システム / ソフトウェア製品品質（ISO/IEC 25010）[ISO25010]
- データ品質（ISO/IEC 25012）[ISO25012]
- サービス品質（ISO/IEC TS 25011）[ISO25011]

本稿で対象とした利用時の品質モデルとシステム / ソフトウェア製品品質モデルをそれぞれ図 2 と図 3 に示す．両者とも，品質特性とこれを細分化した品質副特性から構成される．データ品質モデル特性については，各特性を固有の視点（正確性，完全性，一貫性，信憑性，最新性，ｱｸｾｼﾋﾞﾘﾃｨ，標準適合性，機密性，効率性，精度，追跡可能性，理解性）とシステム依存の視点 (ｱｸｾｼﾋﾞﾘﾃｨ，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) に分類していることが特徴である．

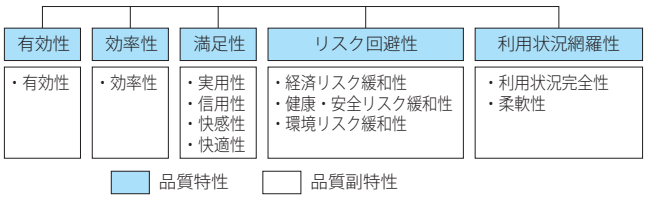


図 2　利用時の品質モデル

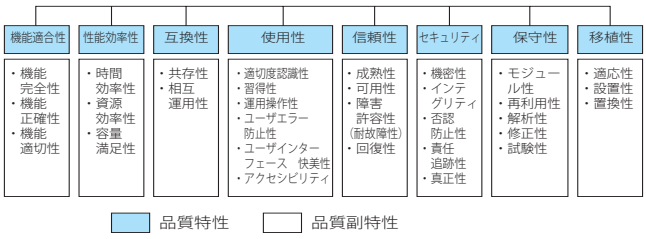


図 3　システム / ソフトウェア製品品質モデル

3.3 メトリクス測定値による評価

品質技術と品質特性の関係の体系化及び，実製品への適用評価を実現するために，我々は，その多様性に対応できる点に着眼し，「異なる品質間の関係を総合的に実証した世界初のベンチマーク（WSQB2017）」[早稲田大学 2017] を活用することとした．上記ベンチマーク中で提案されている RISE メトリクスは，SQuaRE シリーズ中の ISO/IEC 25022，25023 に規定されたメトリクスの中から，GQM 法の適用により実効性のある項目を抽出したものである [WSQB]．当該文献では，利用時の品質として 17 個，システム / ソフトウェア製品品質として 66 個の合計 83 個のメトリクスを品質副特性ごとに定義している．RISE ベンチマークは，複数製品の RISE メトリクス値の分布を評価することで，各製品のポジショニングを可視化できる．本論文では，この RISE メトリクス及びベンチマークの枠組みを活用して，評価を行うこととした．

4　関係性モデル:品質ボックス

本節では，前節で示したソフトウェア品質技術と品質モデルを用いて，それらの関係を示すモデル : 品質ボックスを提案する [品質ボックス 2017]．その品質ボックスの概念図を図 4 に示す．このモデルでは，横軸に SQuaRE で示されるソフトウェアの品質特性 (正確性，可用性など)，縦軸に SQuBOK の品質技術 (形式手法，テストの技法など) でマトリクスを構成し，それらの関係を明らかにしている．また，各品質特性に対応する RISE メトリクスも明示することで，各品質特性における重要メトリクスを示す．我々は，以下の手順で品質ボックスにマーク付けし，品質技術と品質特性の関係を明らかにした．

(1)　Step1：SQuBOK に基づき，汎用的かつ客観的な観点から，品質特性の実現に寄与する品質技術を特定する [小島 a2017].

(2)　Step2：各開発現場の品質保証の経験・実績 ((品質エクスペリエンス)) を基に，個別かつ主観的な観点から，品質特性の実現に寄与する品質技術を特定する [小島 b2017].

上記の 2 ステップについて，4.1 ～ 4.2 節でそれぞれ詳説する．

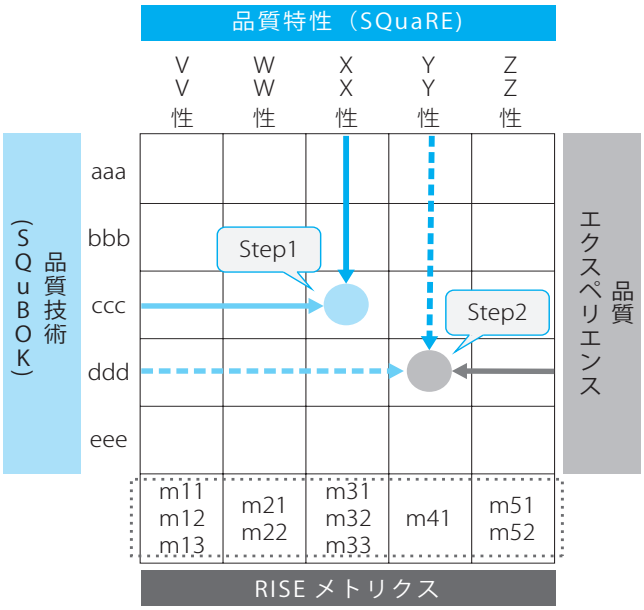


図 4　関係性モデル : 品質ボックス

4.1 汎用的かつ客観的な観点からの品質技術と品質特性の対応付け (Step1)

品質技術 (SQuBOK) と品質特性 (SQuaRE) の対応付けについては，『ソフトウェア品質技術が品質特性に与える効果の見える化』[小島 a 2017] に網羅的に示している．品質技術と品質特性のマトリクスは，品質技術 (トピックス)142 個を縦軸に，品質副特性 57 個を横軸に配置し，筆者らが汎用的かつ客観的に「SQuBOK に品質特性との関係が明記されていること」を基準にマークした．図 5 は，縦軸を品質技術の知識領域（15 個）に，横軸を利用時の品質とシステム / ソフトウェア製品品質の品質特性 (13 個) に集約して，知識領域と品質特性の関係を示したものである．①印が前述のマークした品質特性である．

SQuBOK ガイド第 2 版をベースに作成				ソフトウェア品質特性												
				利用時の品質					システム／ソフトウェア製品品質							
カテゴリ	副カテゴリ		知識領域	有効性	効率性	満足性	リスク回避性	利用状況網羅性	機能適合性	性能効率性	互換性	使用性	信頼性	セキュリティ	保守性	移植性
3. ソフトウェア品質技術	工程に共通なソフトウェア品質技術	1	メトリクス	①	①	①	①	①	①	①	①	①	①	①	①	①
		2	モデル化の技法						①	①			①		①	①
		3	形式手法						①				①			
	工程に個別なソフトウェア品質技術	4	品質計画の技法	①	①	①	①		①	①	①	①	①	①	①	①
		5	要求分析の技法	①	①	①	①	①	①	①	①	①	①	①	①	①
		6	設計の技法						①	①	①	①	①	①	①	①
		7	実装の技法						①	①			①	①	①	①
		8	レビューの技法				①	①	①	①		①	①	①	①	①
		9	テストの技法					①	①	①		①	①	①		
		10	品質分析・評価の技法						①	①		①	①		①	①
		11	運用の技法							①		①	①	①	①	
		12	保守の技法							①	①				①	
	専門的な品質特性のソフトウェア品質技術	13	使用性の技法					①				①				
		14	セーフティの技法				①	①		①		①	①	①		
		15	セキュリティの技法					①		①		①	①	①		

図 5　知識領域と品質特性のマッピング

4.2 個別かつ主観的な観点からの品質技術と品質特性の対応付け (Step2)

ここでは，品質の作り込みフェーズに対応するモデル化の技法，形式手法，要求分析の技法，設計の技法，実装の技法の 5 つの知識領域にターゲットを絞り，ここに分類される品質技術（全 29 種類）と品質特性との関係をより詳細に調査・分析した．具体的には，下記の 3 手法を用い，文献 [SQuBOK2014] 編さんの観点に基づき実施した．

(1) 現場で品質保証に携わった経験をもとに，品質技術ごとに調査・分析を行い，経験から得られた知見に基づいて関係性を抽出した．例えば UML(品質技術) は，コンポーネント単位の仕様を明確に表現でき，モデルレベルで，コンポーネントの再利用性やほかのコンポーネントとの互換性 (品質特性) があると考えられる．

(2) 文献 [SQuBOK2014] を品質技術ごとに更に詳細に調査し，関係性に直接言及していなくとも，前後の文脈などから関係性が推測できるものを抽出した．例えば，離散系のモデル化技法 (品質技術) には，「シミュレーション解析により信頼性や効率性を確認できる」という記載があり，連続系のモデル化技法でもシミュレーション解析を行うことから，これらは信頼性と効率性（品質特性）に寄与すると考えられる．

(3) 顧客要件を実現するという観点から機能適合性と品質技術との関係性を再確認した．

図 6 は，図 5 の①印に加え，上記で述べた 3 つの手法による新たな関係性を発掘した個所に②印，③印，④印をマークして作成したマトリクスの一部抜粋である．

品質向上に最もかかわる「品質の作り込み」フェーズを対象としたマッピング (Step2) の充実により，その関係を見出す手法の違いも考慮した品質技術の選択が可能となり，ソフトウェア開発エンジニアのためのガイドとしてもより実用性の高いものとなったと考えられる．例えば，プロジェクトの特性を考慮して重要視する品質特性を選び，それを実現するために必要な品質技術を，今回得たマトリクスから優先度を考慮して抽出することが可能になる． また，①印が少ない品質特性に対しては，② / ③ / ④印により，見落としがちであった品質技術も選択の対象となり，品質向上に寄与できる．

SQuBOK ガイド第 2 版をベースに作成						ソフトウェア品質特性													
						利用時の品質					システム／ソフトウェア製品品質								
知識領域		副知識領域			トピックス	有効性	効率性	満足性	リスク回避性	利用状況網羅性	機能適合性	性能効率性	互換性	使用性	信頼性	セキュリティ	保守性	移植性
2	モデル化の技法	1	離散系のモデル化技法	1	UML						①	①	②		①		①	①
				2	SysML						①	①	②		①		①	①
				3	構造化チャート (PAD)						①	①	②		①		①	①
		2	連続系のモデル化技法	-	-						④	③	②		③		③	③
		3	ドメイン特化言語	-	-						④	③	②		③		③	③
3	形式手法	1	形式仕様記述の技法	-	-						①				①			
		2	形式検証の技法	-	-				①	①	①				①			

図 6　発掘された新たな関係性（一部抜粋）

品質技術の適用により，期待した品質特性を得られたか否かを判断できることが，品質向上のために求められる．そこでは，品質特性達成度の評価指標の定義と定量化が必要である．

また，RISE メトリクスを対象製品の品質特性達成度の判断に利用すると共に，RISE ベンチマーク結果と照合することにより，強化すべき品質特性の抽出が可能になる．例として，品質特性ごとに整理した RISE メトリクスの代表 (最大 4 つ) を，図 7 に示す (紙面の都合により一部抜粋版).

ソフトウェア品質特性												
利用時の品質					システム／ソフトウェア製品品質							
有効性	効率性	満足性	リスク回避性	利用状況網羅性	機能適合性	性能効率性	互換性	使用性	信頼性	セキュリティ	保守性	移植性
タスク完了率	タスクにかかった時間の平均	製品に対する満足度	経済的損失を感じる頻度	主要な目的以外での製品利用の有無	要求実装率	時間効率性の試験有無	他製品を共存させて試験する意図の有無	機能の動画説明対応率	不具合除去率（単体試験）	データのアクセス権限管理対応率	クラスの結合度	複数環境の試験有無
タスク当たりエラー数	タスク中の総アクションの無駄でないアクションの率	Net Promoter Score	健康や人命への影響を感じる頻度	非主要目的での製品利用時タスク達成度合い	深刻不具合除去率	応答時間平均	取り扱うファイル形式のうち、インポート/エクスポート両対応の比率	機能の説明記載率カタログ	不具合除去率（結合試験）	データの暗号化対応率	関数のサイクロマティック複雑度	インストーラの試験有無
エラーが発生したタスクの率		機能に対する満足度	環境への影響を感じる頻度		システム試験数目標達成率	応答時間実測対目標		機能の説明記載率マニュアル	不具合除去率（システム試験）	データの破損防止策対応率	クラスの凝集性の欠如	インストーラ時間平均
エラーを起こした被験者の率		信用度合い			ユーザの意図に即す度合い	ターンアラウンドタイム平均		機能のUndo対応率	MTBF目標達成率	ネットワーク経路のデジタル署名対応率	コーディング規約違反（測定保留中）	インストーラ提供形態対応率

図７　品質特性ごとの RISE メトリクス（一部抜粋）

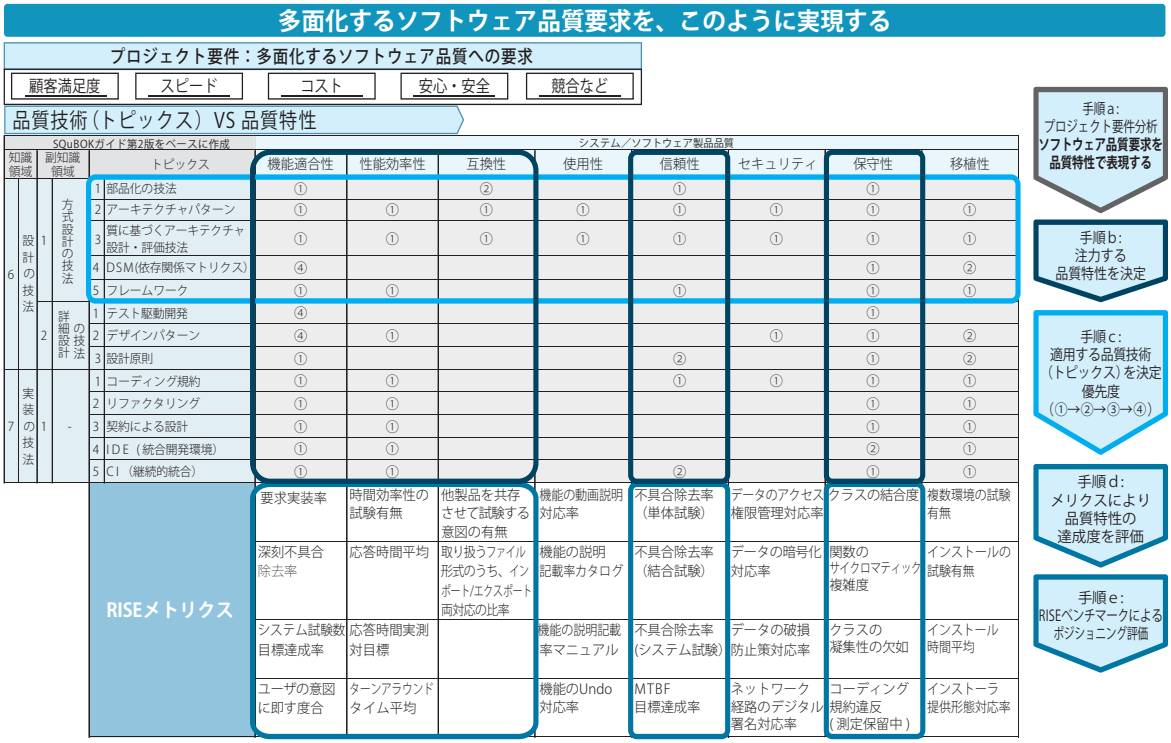


図８　多面化するソフトウェア品質確保プロセス

5　品質確保プロセスの提案

本節では，図 4 で示した関係性モデル : 品質ボックスと RISE ベンチマークを用いて，多面化するソフトウェアの品質確保をするプロセスを提案する．図 8 にそのプロセスを示す．本プロセスにおいては，図の右側に書かれている以下の手順 a ～ e により，ソフトウェアの品質特性の評価を行う．

(1)　手順 a：まず，プロジェクトに求められる要件（品質要求）を，品質特性で表現する．例えば，長期間にわたり利用されるシステムであれば，システム更改を考慮し，互換性や移植性を選択する．

(2)　手順 b：手順 a で抽出した品質特性の中から注力するものを決定する．ここでは，QCD のトレードオフの考慮や，RISE メトリクスやベンチマークの活用などにより，戦略的に方針を決定する．

(3)　手順 c：該当品質特性を実現可能な品質技術を抽出する．

(4)　手順 d：品質特性の達成度合いを RISE メトリクスにより測定・評価する．

(5)　手順 e：RISE ベンチマークによるポジショニングを評価し，製品の強み・弱みを判別する．必要に応じて，品質技術やメトリクス目標値にフィードバックし，ソフトウェアの品質及びプロセスの改善につなげる．

6　品質確保プロセスの検証

本節では，前節で提案した品質確保プロセスの妥当性評価を行う．具体的には，RISE 調査対象でもある製品 A(パッケージ製品), 製品 B（クラウドサービス）の開発において，図 8 の手順に基づき品質ボックスを適用し，検証を行った．

表 1　要件を満たすための品質特性 (製品 A)
（最優先：◎→○→△→×：低優先）

プロジェクト要件	利用時の品質					システム／ソフトウェア製品品質							
	有効性	効率性	満足性	リスク回避性	利用状況網羅性	機能適合性	性能効率性	互換性	使用性	信頼性	セキュリティ	保守性	移植性
データ処理能力重視	○	○				○	◎		○	○	△	○	
早期デリバリー			△	△	△			×					×

表 2　要件を満たすための品質特性 (製品 B)
（最優先：◎→○→△→×：低優先）

プロジェクト要件	利用時の品質					システム／ソフトウェア製品品質							
	有効性	効率性	満足性	リスク回避性	利用状況網羅性	機能適合性	性能効率性	互換性	使用性	信頼性	セキュリティ	保守性	移植性
クラウドサービスとしての運用強化							○			◎	○	○	
W 向け制御機能での競争力強化	◎	△	○	△	△	○		×	△				×

6.1　品質ボックスの適用

(1)　手順 a，b: 品質要件から品質特性を決定

製品 A，製品 B のそれぞれに求められる要件（品質要求）を品質特性で表現し，その注力する品質特性を決定したものを表 1，表 2 に示す．

製品 A は，データ処理能力を強く要求されているため，製品品質の性能効率性を最優先とし，使用性，信頼性を重視した．また，利用時品質として，有効性や効率性を重視した．一方，市場優位性を勝ち取ることがビジネス上の戦略でもあり，早期デリバリーを強く要求されているため，システム動作条件 (OS など) を固定化することとし，互換性・移植性は低優先とした．

製品 B は，クラウドサービスとして，既存利用中の全ユーザに対し自動的に適用される追加機能であるため，信頼性を優先事項とした．また，対象開発での優先機能として Windows 向けの制御機能にて競合他社との差異化要素を追加し，市場競争力の強化を行うことが求められたことから，有効性を高優先とした．一方，いち早く差別化機能を市場投入するため，互換性や移植性は優先度を下げることとした．

(2)　手順 c: 注力する品質特性から品質技術を決定

品質ボックスの品質特性と品質技術のマトリクスの抜粋を図 9 に示す．これを用いて，製品 A について最優先とする性能効率性を高めるために適用する品質技術を決定した．

品質ボックスで定義する性能効率性に関与する知識領域ごとの品質技術数と，その内で本製品開発に適用した数を図 10 に示す．性能効率性という特徴から，実動作での確認を重視し，テストの技法を多く取り入れた．また，昨今強く求められる安心・安全確保を考慮し，該当する品質技術も採用した．

SQuBOK ガイド第 2 版をベースに作成				システム／ソフトウェア製品品質							
知識領域	副知識領域		トピックス	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
9 テストの技法	5 利用に基づいた技法	1	運用プロファイルによるテスト	①				①			
		2	ローカライゼーションテスト	①							
		3	ユーザー環境シミュレーションテスト	①	①		①				
		4	整合性確認テスト	①	①			①			
	6 ソフトウェアの形態に基づいた技法	1	オブジェクト指向テスト	①							
		2	Web システムのテスト	①	①		①	①	①		
		3	GUI テスト	①							
		4	サーバーサイドのテスト	①	①			①			
		5	データベーステスト	①							
		6	並行プログラムのテスト	①				①			
		7	プロトコル適格性テスト	①							
		8	実時間のテスト	①				①			
		9	モバイルアプリケーションのテスト	①	①				①		

図 9　品質ボックスから抜粋

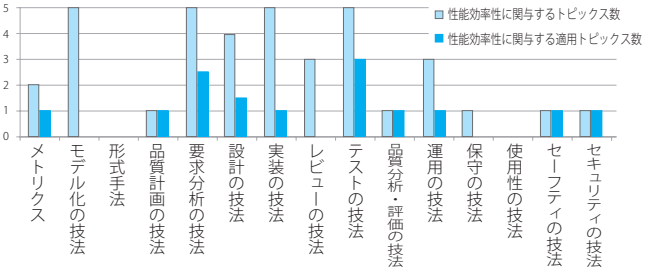


図 10　適用した品質技術数 (性能効率性)

製品 B も同様に最優先とする信頼性を高めるために適用する品質技術を決定した．

(3)　手順 d: 決定メトリクスの測定と達成度評価

品質ボックスの RISE メトリクスの内，検証対象の製品 A，製品 B が最優先とした品質特性に関する測定値をそれぞれ図 11 に示す．どちらも測定条件ごとに目標値と実績値を測定し，RISE メトリクスとして各種平均値や実績対目標値を導き出した．

製品 A は，性能効率性について RISE メトリクスで推奨されている大小比較も参考にすると全般的に適切なメトリクス測定値を得られたことが分かる．一方，製品 B は 信頼性について可用性／障害許容性は適切なメトリクスが得られているが，成熟性の結合試験とシステム試験密度／結合不具合発見率について目標とする水準に達していないことが分かる．この点について開発チームの調査を行ったところ，チーム内で用いている指標値において，プラットフォーム特性に応じた準備ができていなかったことが判明した．このことから，品質確保プロセスの適用によって，開発における課題が抽出できたと言える．

6.2　品質ボックス適用評価

図 8 の手順 e について，製品 A と製品 B の RISE ベンチマークとしてそれぞれ特徴のある一部の品質特性に関し中央値と該当製品のポジションを図 12，図 13 に示し，その分析

製品 A	RISE メトリクス（性能効率性）										
	時間効率性の試験有無	応答時間平均	応答時間実測対目標	ターンアラウンドタイム平均	ターンアラウンドタイム実測対目標	スループット目標達成率	資源効率性の試験有無	CPU 使用率最大値	メモリ使用率最大値	容量満足性の試験有無	ユーザ同時アクセス可能数目標達成率
大 or 小が望ましい	大	小	小	小	小	大	大	小	小	大	大
メトリクス測定値	1	-	-	-	-	2.08	1	0.03	NA	1	NA
測定条件 (サンプル)	-	ログイン画面表示	-	プロセス起動	-	データ蓄積	-	データ保存	データ保存	-	ユーザ数
	-	-	-	データ検索 (500 件)	-	-	-	データ検索	データ検索	-	-

製品 B	RISE メトリクス（信頼性）												
	成熟性									可用性		障害許容性	
	単体不具合除去率	結合不具合除去率	シス不具合除去率	単体不具合発見率	結合不具合発見率	シス不具合発見率	予測数 / 不具合数	結合試験：実際 / 目標	シス試験：実際 / 目標	運用試験有無	運用実時間	異常系結合試験成功率	異常系シス試験成功率
大 or 小が望ましい	大	大	大	小	小	小	-	大	大	大	大	大	大
メトリクス測定値	-	1.00	1.00	NA	0.56	1	1.01	0.42	0.45	1.00	1.00	0.97	0.96
測定条件 (サンプル)	試験項目数／障害数／除去数									クラウドサービス環境	MTBF	アプリ異常等	サーバーインフラ故障等

図 11　メトリクス測定結果

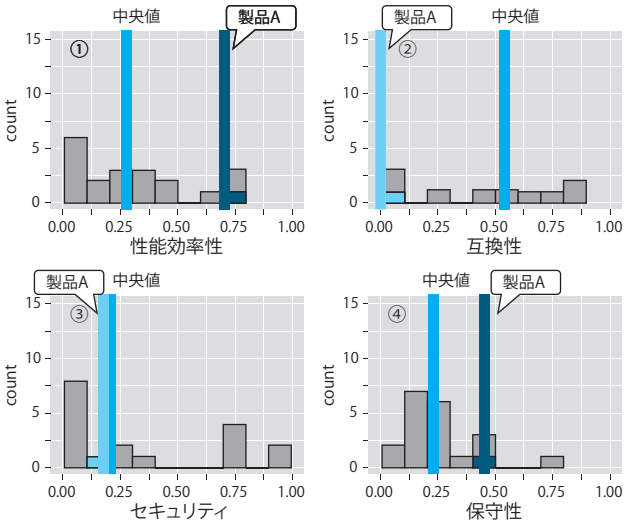


図 12　品質特性のベンチマーク結果 (製品 A)

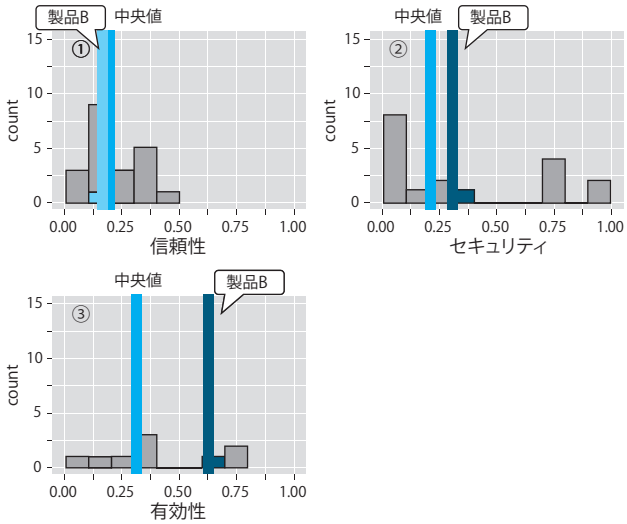


図 13　品質特性のベンチマーク結果 (製品 B)

結果と考察を示す．各グラフは，RISE ベンチマークによってメトリクス測定値をスコア化し，21 種類の他製品の品質指標の分布と，当該製品の位置を示している．

(1) 製品 A

製品 A の RISE メトリクスによるベンチマーク結果を図 12 に示し評価する．

① 性能効率性

結果 : 中央値を上回り，他製品より優れている．

考察 : 最優先とした品質特性 (表 1) であり，品質ボックスの性能効率性に寄与する品質技術からとくにテスト技法を多く取り入れ (図 10)，十分な検証を行った成果であり，製品としての強みが確保できた．

② 互換性

結果 : 中央値を下回り，優位性が低い結果となった．

考察 : 低優先とした品質特性 (表 1) であり，品質ボックスによる品質技術の選択を行わなかった．他製品のスコアが広範囲で分布している特徴があり，エンハンス時には品質ボックスにより意図的に品質技術を利用することで，優位性を高められると考える．

③ セキュリティ

結果 : 中央値とほぼ同等の優位性をもつ．

考察 : 低めの優先度とした品質特性 (表 1) であり，品質ボックスによる品質技術の選択を行ったが適用範囲を一部としたため，本結果になったと考える．なお，本メトリクスにおいては，スコア分布が二極化している特徴が見られる．本製品の優位性を高めるためにも，次回エンハンス時は，品質ボックスから選択した技術の適用範囲拡大を検討する．

④ 保守性

結果 : 中央値を上回り，他製品より優れている．

考察 : 高優先とした品質特性 (表 1) であり，品質ボッ

クスから選択した品質技法としてモデル化の技法や設計の技法を多く取り入れた成果であり，製品としての優位性が確保できた．

(2) 製品 B

製品 B の RISE メトリクスによるベンチマーク結果を図 13 に示し評価する．

① 信頼性

結果 : 中央値をやや下回った．

考察 : 運用強化として最優先とした品質特性 (表 2) であるが，品質ボックスによる品質技術の選択を行なっておらず，メトリクスの数値から弱点であることが判明した．今後は品質技術の適用を考える．

② セキュリティ

結果 : 中央値を上回ったものの，ほぼ平均にある．

考察 : 運用強化として優先とした品質特性 (表 2) であり，品質ボックスによる品質技術の選択もしたが，他分布をみるとログイン認証方式の多様性など，更なる機能追加の検討が必要と認識できた．

③ 有効性

結果 : 中央値を上回り，かなりの優位性を持つ．

考察 : 競争力強化として，最優先とした品質特性 (表 2) であり，品質ボックスによる品質技術としては，要求分析の技法を選択．既存ユーザからの要望の取り入れや，要件の妥当性評価 (第三者試験) の重視が結果につながったと考えられる．

6.3 考察

従来，ソフトウェア開発に採用する品質技術は類似製品開発の経験などから選ぶことが多かった．今回，ソフトウェアに期待される要件から品質特性を特定し，その実現に寄与できる品質技術から適切な品質技術を選ぶプロセスを提案し，一定の効果が得られたと考える．また，ベンチマーク結果から効果が得られなかった点は，その原因を深堀することにより品質ボックスやメトリクス目標値にフィードバック可能であると考える．

7 まとめと今後の課題

本稿では，図 4 の関係性モデル : 品質ボックスと RISE ベンチマークを用いた多面化するソフトウェア品質確保プロセスを提案・検証した．提案手法を 2 種類の異なる製品に適用評価し，重要な品質指標の抽出や，製品の品質評価において有効であることを示した．

今後は，今回提案した品質確保プロセスの開発現場で実践を進めると同時に，IoT や AI などの新しい技術に要求される品質技術と品質特性の関係性の体系的整理を進めていく予定である．それにより，様々な製品に対して，最適な品質技術の選択決定が可能になると考えられる．

謝辞 : 本稿の執筆にあたり，日科技連 SQiP ソフトウェア品質委員会，SQuBOK V3 研究チームの方々，富士通，富士通研究所，富士通ビー・エス・シーの RISE 研究関係者に多くのご協力，ご助言を頂いた．ここに謝意を記す．

【参考文献】

[早稲田大学 2017] 鷲崎弘宜，「異なる品質間の関係を総合的に実証した世界初のベンチマーク（WSQB2017)」，先導的研究支援事業（RISE)，2017
[WSQB] WSQB17:Waseda Software Quality Benchmark，http://www.washi.cs.waseda.ac.jp/?page_id=3479
[品質ボックス 2017] 品質技術と品質特性とメトリクスの関係性モデル．筆者らが公開したもの．
http://www.washi.cs.waseda.ac.jp/wp-content/uploads/2017/07/SQIP_Quality_Box.pdf
[Gordon2005] Dan Gordon，Ted Stehney，Neha Wattas，Eugene Yu，"System Quality Requirements Engineering (SQUARE): Case Study on Asset ManagementSystem,PhaseII"，CMU/SEI-2005-SR-005
[Firesmith2006] Donald Firesmith，"QUASAR: A Method for the QUality Assessment of Software-Intensive System ARchitectures"，CMU/SEI-2006-HB-001
[Jagdish2002] Jagdish Bansiya and Carl G. Davis，"A hierarchical model for object-oriented design quality assessment"，IEEE Transactions on Software Engineering，Vol.28，No.1，2002
[Bass2005] Len Bass，Paul Clements，Rick Kazman，前田卓雄・佐々木明博（訳)，実践ソフトウェアアーキテクチャ，日刊工業新聞社，2005.
[山田 2016] 山田　洋二，マツダ技法 No.33 (2016) 品質機能展開を活用した技術開発プロセス，2016. [Zhu2005] Hong Zhu，"Software Design Methodology: From Principles to Architectural Styles"，Elsevier，2005
[SQuBOK2014] SQuBOK 策定部会編，ソフトウェア品質知識体系ガイド第 2 版，オーム社，2014.
[ISO25010]ISO/IEC25010:2011　Systems　and software engineering – Systems and software Quality Requirements and Evaluation(SQuaRE)–System and software quality models
[ISO25012] ISO/IEC25012:2008　Software engineering–Software product Quality Requirements and Evaluation (SQuaRE) – Data quality model.
[ISO25011] ISO/IEC TS 25011:2017 Information technology -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Service quality models
[小島 a 2017] 小島 嘉津江，森田 純恵，若本 雅晶，宗像 一樹，鷲崎 弘宜，ソフトウェア品質技術が品質特性に与える効果の見える化，情報処理学会第 195 回ソフトウェア工学研究発表会，2017
[小島 b 2017] 小島 嘉津江，森田 純恵，菊池　慎司，鷲崎 弘宜，ソフトウェア品質技術が品質特性に与える効果の見える化と活用の一考察，日科技連ソフトウェア品質シンポジウム 2017，A4-1

ソフトウェア技術者エントリ層教育コースによる人材育成とその評価


福間 和人※1


田村 直樹※2


藤岡 卓※2

筆者らは三菱電機グループ全体の職能教育として，入社 2-5 年のソフトウェア技術者全員を対象とした「ソフトウェア技術者エントリ層教育コース」を企画・設計し，2009 年から実施してきた．
本論文では，開講から 10 年を迎える本コースについて，これまでの取り組みに対する評価と現状の課題を明らかにする．

An Evaluation of Software Engineers Entry-level Discipline Course

Kazuhito Fukuma※1, Naoki Tamura※2, Taku Fujioka※2

We have been providing an occupational ability development program called software engineers entry-level discipline course, to which any software engineers between two and five years after employment by Mitsubishi Electric group companies should apply, since 2009. This paper shows our nine years' operation experiences of this course and evaluates the effectiveness of our approach.

1　はじめに

IoT，AI 等の技術や事業環境が大きくかつスピーディに変化する現代社会において，システムの付加価値を生み出すのはソフトウェアである．情報処理学会誌 2017 年 8 月号 [IPSJ2017] では「ソフトウェアが社会のすべてを定義する時代」と言っている．ソフトウェア技術者の果たす役割はますます重要になってきた．

筆者らは，ソフトウェア技術者全員が一定の規律とスキルを確実に習得することで，開発時のプロセス品質のバラツキを低減するとともに，事業環境の変化に耐え得る技術者の流動性を確保することを期待して，三菱電機グループ全体の職能教育として，入社 2-5 年のソフトウェア技術者全員を対象とした「ソフトウェア技術者エントリ層教育コース (SeeD: Software engineers entry-level Discipline course)」を企画・設計 [Fujioka2014] し，2009 年から実施してきた．

コース自体の改善に加えて受講者の上司の巻き込みや修了者による改善事例発表会の実施等によって職場でも認知され，修了者は 2000 名を超えた．一方で，受講後のアンケートから役立ち度は十分高いものの，学んだことを必ずしも継続して実践できていない実態も見えてきた．

※1　三菱電機株式会社　人材開発センター (現在　三菱電機コントロールソフトウェア株式会社)
※2　三菱電機株式会社　人材開発センター

本論文では開講から 10 年の節目を迎えるにあたってこれまでのコース実績を振り返り，コースの狙いの定着状況からコース改善の事例とともに今後への課題を述べる．第 2 章ではコースの狙い，構成とこれまでの実施状況を，第 3 章では継続して実施しているコース改善の取り組みから特徴的な事例を紹介する．第 4 章では過去の受講者を対象に実施したアンケート結果をもとに，実施効果に対する評価と現状の課題について考察し，第 5 章では事業環境や技術の変化に対応するための今後の取り組みについて述べる．

2 コースの概要と実施状況

2.1 コースの狙い

システムの大規模化，複雑化による分業化の進展に伴い，一人の技術者がソフトウェア開発全工程を体験する機会が減っている．流用開発，派生開発の増加に伴い，ソフトウェアを一から分析・設計する経験を経ない技術者も増えた．結果的に若手ソフトウェア技術者が，職務に必要なスキルと規律を習得することが難しくなっている．職場環境に依存する OJT や自己啓発を効果的なものにするためにも，組織的に系統的な職能教育を展開するニーズが高まっている．

三菱電機グループでは，入社 2 ～ 5 年目までの担当者を育成対象とし，継続的改善を風土として構築するための人づくり・土台作りを狙い，ソフトウェア開発に関する体系的な知識と規律，自立した技術者としての改善意識を身につけさせる「ソフトウェア技術者エントリ層教育コース」を，三菱電機グループのソフトウェア技術者全員を対象とした職能教育として展開している．

コースの到達レベルとしては以下を設定した．

① 一連の SW プロセス（SW 要求分析～ SW テスト）を，基本手法を使用して実施できる．

② 品質を計画・監視・制御することができる．

③ 自己のプロセスの実績を測定し見積り，改善ができる．

2.2 コースの構成

本コースは表 1 に示す通り 5 つの講座から構成される．コースの最初にパーソナルソフトウェアプロセス (PSP) [Humphrey2001][Panasonic2017] を配して若手技術者にプロとしての規律を教え込んだ後，ソフトウェア開発の一連のプロセスを順に学ぶ．各講座間は一ヶ月程度空け，受講者が講座で学んだことを職場に帰って実践できるようにして，反復学習による習慣化を狙っている .

表 1　コースの構成

講座	テーマ	期間	講座内容	演習
第1回	基礎	2日	SW開発におけるプロセス 品質の重要性と測定／評価 見積り･計画 進捗管理	
第2回	設計技法	3日	要求分析･設計の手順と仕様書作成 要求分析、設計	要求分析 品質測定 分析モデルレビュー
第3回	実装技法	3日	SW詳細設計 コーディング 実装品質の向上	内仕作成～コードレビュー 良／悪コードの評価
第4回	テスト技法	3日	テスト設計技法 単体テスト SW適格性確認テスト	コード、仕様書の不具合検出 不具合報告
第5回	総合演習	3.5日	第1回～第4回の集大成 開発プロセス全体を通したチーム演習	見積り 設計／実装／テスト 評価･分析 成果発表

顧客との関係や事業の背景から，グループ内に存在する用語の差異を吸収するため，開発プロセスとしてソフトウェアライフサイクルプロセス (SLCP-JCF2013)[IPA2013] を採用し，プロセスの記述には SLCP をベースとした IPA SEC の組込み向け開発プロセスガイド [IPA2007] を採用した．

グループ全体で毎年数百人を育成する必要に対応するため，同一の教材で 3 種類の講座実施形態を提供している．

① 自主開催講座

事業所あるいは関係会社が企画し，実践する講座形態である．「講師養成講座」を受講し講師資格認定を受けた者が講師として講座を展開する．

② 定期開催講座

三菱電機の研修センターでの集合講座である．2 ～ 3 回／年の頻度で受講者を募集して講座を開催する．主として自主開催や出前開催が困難な組織に向けた位置づけである．

③ 出前講座

事業所あるいは関係会社の要請に基づき講師が出向いて講座を実施する．事業所内の講師が充足するまでの過渡的な施策として，初期に多く実施していたが，現在は事業所間の講師の交流目的で活用している．

2.3 コースの実施状況

コースの実施状況を図 1 に示す．

2009 年度にまず定期開催講座を開始，翌 2010 年度から講師養成講座を実施して，2011 年度からは各事業所，関係会社での自主開催講座を開始，2017 年度末までに約 150 名の講師を養成し，定期開催講座と合わせて 300 名／年以上を育成できる体制を確立，2017 年度末時点で，累計 2000 余名がコースを修了した．育成体制の充実に伴い受講修了者が増加した結果，データを取り始めた 2011 年初に 700 余

名いた対象者 (入社 2 ～ 5 年目の未受講者) は年を追うごとに減少し，2018 年以降は新たに対象になる人数を育成能力が上回って定常状態となる見込みである .

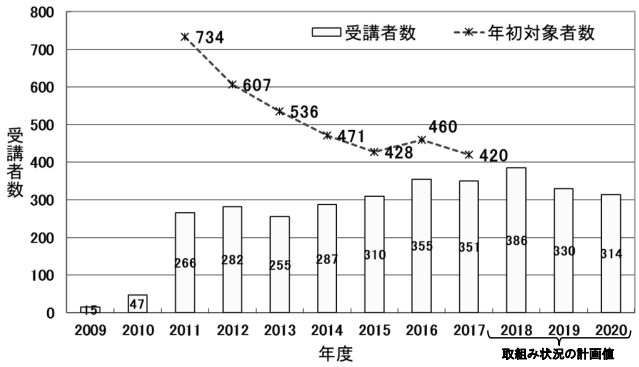


図 1. 受講者数推移

3　コース改善の取り組み

本コースは，開講から毎年継続して教材改訂や運用改善を実施してきた．以下に，3 例を紹介する．

3.1　受講者上司のコース理解施策

本コースでは各回講座で受講者に終了アンケートを実施し，受講者の理解度把握と講座改善を行ってきた．

これと並行して，コース開講から数年が経過した定期講座の受講者の上司 25 人に対して，受講者の職場での実践状況をアンケート調査した．

① ソフトウェア開発基本手法の実践状況

② ソフトウェア品質管理の実践状況

③ プロセス改善の実践状況

上記アンケート結果と受講者の終了アンケートの理解度と突き合わせを行った結果，図 2 ～ 4 となった．受講者が 5 段階評価した理解度の平均点を 4 未満，4 以上 4.5 未満，4.5 以上に分けて，上司のアンケート①～③の回答数を棒グラフに表した．

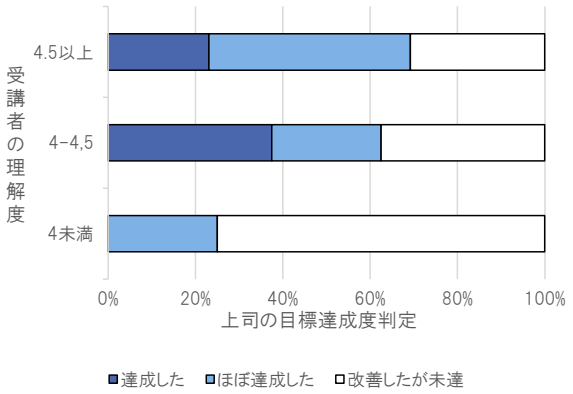


図 2.　SW 開発基本手法実践と受講者理解度の対応

図 2 から，SW 開発基本手法については，受講者の理解度が 4 以上になると，上司の 60% 以上が達成と判断している .

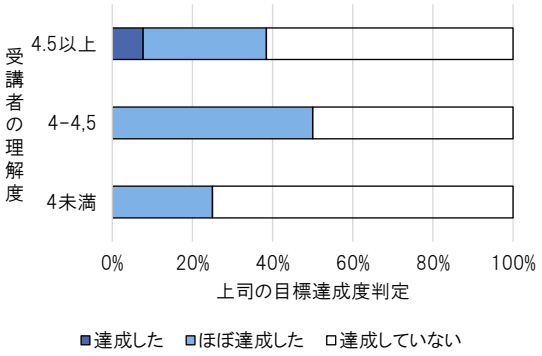


図 3.　SW 品質管理実践と受講者理解度の対応

図 3 から，SW 品質管理実践については，受講者の理解度が 4 以上になると上司が達成と判断しているが，50% 以上は達成していないと判断している．

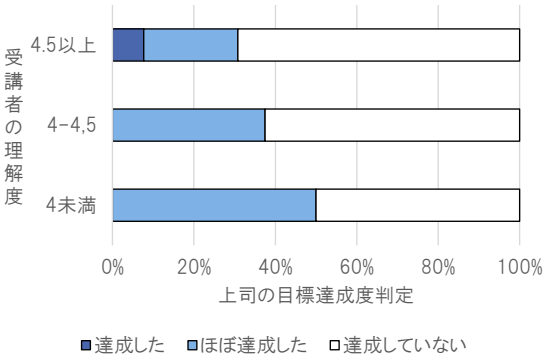


図 4.　プロセス改善実践と受講者理解度の対応

図 4 から，プロセス改善について，受講者の理解度が 4 以上でも，上司は達成していないと判断している．

アンケート①は，受講者の理解度が高くなるにつれ，上司の評価も高まるが，②や③は，受講者の理解度の高さと上司の評価の高さが連動しない傾向にある．①は受講者の成果物に変化が現れやすいため，上司が達成度を評価しやすいためと考える．

一方，②や③は，

- 受講者はコース内容を理解したが，職場での実践はできていない
- 受講者はコース内容を理解し職場での実践をしているが，上司がその取り組みを把握できていない

というケースが考えられ，上司が達成していないと判断した可能性がある．現在の受講者の上司は本コースを受講していない世代であり，コースの趣旨や目的，指導内容を

十分に理解しておらず，コース修了後の部下に適切な指導ができないという可能性も考えられる．

そこで，受講者の上司に本コースの内容を理解してもらうために，以下の取り組みを行った．

- 受講者の募集時に，受講者の上司にはコースの趣旨説明を徹底する
- 講座の終了アンケートやレポート課題は，必ず上司がコメントを記載することとし，その際には受講者が上司と直接対話して，内容の説明を行う（図 5．参照）
- 第 5 回「総合演習」最終日の成果発表会に上司が出席し，受講者の成長を実感してもらう．

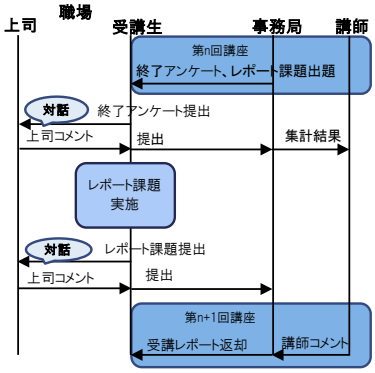


図 5．終了アンケート，レポート課題の流れ

この取り組みにより上司のコース理解が進み，終了アンケートやレポート課題のコメントが，具体的なコースの内容に則したものになってきた．成果発表会参加の上司からは，活発に質問や指導コメントが出るようになった．また，成果発表会の後に上司と講師が意見交換する時間を設け，受講者の職場での取り組み状況や講座の要望などを確認しているが，大半の上司が部下の取り組みを理解し，職場でも指導できるレベルになってきた．

3.2　第 5 回講座「総合演習」の改善

第 5 回「総合演習」は，チームによるプログラム開発を行うことで，これまでの各回で学んだ技法を，ソフトウェア開発プロセス全体を通して体感する講座である．受講者は課題プログラムの「ソフトウェア要求分析～適格性確認テスト」を 3 日間で行い，そこで得たデータを元に成果をまとめ，受講者自身のプロセス改善策を発表する．しかし，上記のやり方では開発を完了したチームは「すべての欠陥を除去した」という報告をしていた．

製品の品質評価は，客先など第三者による受入れテストや，出荷後の欠陥発生状況を含めて総合的に判断されるべきものである．そこで，定期講座では講座日数を半日増やし，最終日には開発したプログラムをチーム間で交換し合い，共通のテストケースを使った受け入れテストを行うことにした．ここで検出された欠陥は，擬似的に「市場流出欠陥」として，製品出荷までに欠陥除去ができたのかを判断するものとなる．混入された欠陥件数をより正確に計測することで，品質意識の向上と，成果発表会の内容が実体に近いものになった．

また，発表では計画と実績が定量的に比較できる資料の雛形を準備し，「計画値の策定理由」「実績値の状況と計画値との差異分析」「欠陥混入と除去の状況」「欠陥の原因分析と対策」といった項目を，チームメンバが議論できるようにした．これにより，自己のプロセス改善提案もより深みのある内容となった．

3.3　事例発表会の開催

本コースは，自主講座を開催する事業所が年々増加し，いまでは自主講座の受講者が全受講者の 9 割までになっている．一方定期講座は色々な事業所から受講者が集まるため事業所間の交流も生まれるが，自主講座では事業所以外の技術者と交流できない．

そこで，受講者が本コースで学んだ内容を自身の業務に展開した活動事例を発表する「事例発表会」を 2015 年度から開始した．受講者のいる事業所に会場を借りて，年 2 回のペースで開催し，受講者同士が業務改善に繋がる情報交換や相互研鑽する場としている．発表テーマは，第 1 回基礎講座で学んだ PSP の職場での実践状況や，開発プロセスの改善事例紹介など多岐に渡り，質疑も活発に行われている．事例発表会の前後には事業所見学や特別講演，受講者同士のディスカッションなどを企画し，参加者数も増加傾向にある．

また，この発表会を通して，事業所のコース運営窓口や講師が受講者の活動状況を把握でき，自主講座運営や講師の指導方法改善にもつながっている．

4　コース実施効果の評価と考察

本コースは第 3 章に示したような改善を継続しながら 10 年近くが経過した．そこで，本コースの実施効果について評価を行うこととし，「講座内容の職場での実践状況」「業務に対する講座の役立ち状況」をこれまでの受講者にアンケート調査した．

調査は表 2 に示す項目とし，依頼した受講者の 51% から回答を得た．

表 2. 受講者アンケート項目

大項目	調査項目	調査方法
A. 現在の担当業務	「システム管理」～「ソフトウェア適格性確認テスト」「その他」での主担当／担当業務	選択（複数可）
B. 自己のプロセス状況	①自己の時間管理 ②自己の作業見積り／測定 ③自己のスケジュール管理 ④自己の欠陥予測／記録 ⑤上記データを元に作業計画 ⑥自己のプロセス改善	下記3択と自由意見 「現在もしている」 「途中でやめた」 「受講後からしていない」
C. 業務に対する講座の役立ち状況	①品質面の能力向上 ②生産性面の能力向 ③工程管理面の能力向上	下記3択と自由意見 「非常に役立った」 「役立った」 「あまり役立っていない」
	④講座で学んだことで、現在使用／役立っているもの	技法などのキーワード等を記載

4.1 受講者（回答者）の業務種別の内訳

表 2 の「A. 現状の担当業務」から，回答のあった受講者の業務種別を以下のように分類した．

・システムエンジニア（以下 "SE" と略す）

システム要求分析～ソフトウェア要求分析，ソフトウェア適格性確認テスト～システム適格性確認テスト

・プログラマ（以下 "PG" と略す）

ソフトウェア方式設計～ソフトウェア結合テスト

・システムエンジニア兼プログラマ（以下 "SE/PG" と略す）

システム要求分析～システム適格性確認テスト

・その他

上記以外の業務（品質管理やプロセス改善など）

上記の業務種別の割合を図 6 に示す．当社では比較的開発規模の小さい組込みソフトウェア開発の従事者が多く，開発プロセスの上流から下流まで担当する SE/PG が半数近くを占めていた．

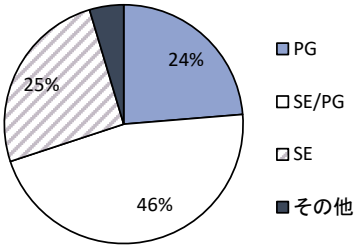


図 6. 回答者業務種別の割合

以下，各項目についての評価と考察を述べる．

4.2 講座内容の職場での実践状況

表 2 の「B. 自己のプロセス状況」の回答を「受講経年比較」と「業務種別比較」でグラフ化したものが，図 7，8 である．各グラフには 80% に補助線を入れ，超えていれば十分に定着していると判断した．

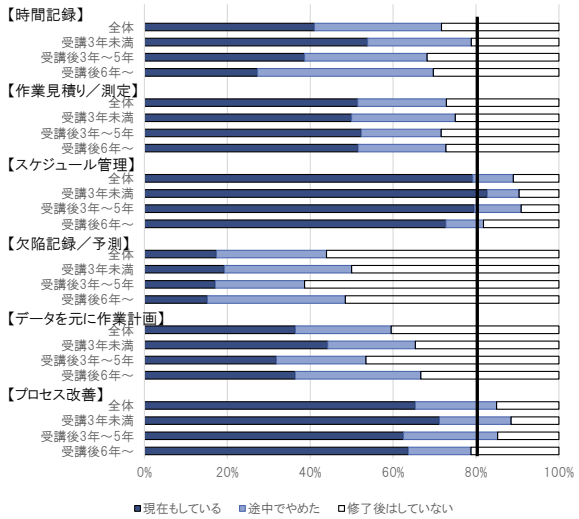


図 7. コース修了後のプロセス実践状況（受講経年比較）

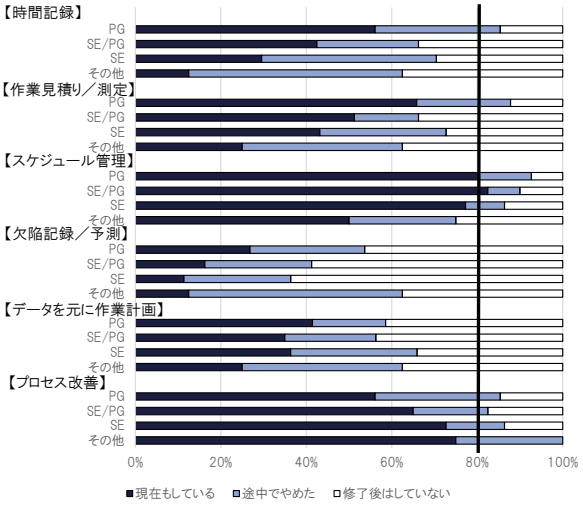


図 8. コース修了後のプロセス実践状況（業務種別比較）

図 7「受講経年比較」から，以下の傾向があった．

① 実践割合が比較的高く，受講経年の変化がない項目

「作業時間見積り／測定」

「スケジュール管理」

「プロセス改善」

② 受講経年で実践の割合が減少する項目

「時間記録」

「データを元に作業計画」

③ 受講直後から実践の割合が低い項目

「欠陥記録／予測」

また，図 8「業務種別比較」から，以下の傾向があった．

④ PG → SE/PG → SE →その他の順で増加する項目

「プロセス改善」

⑤ PG → SE/PG → SE →その他の順で減少する項目

「時間記録」

「作業時間見積り／測定」

「欠陥記録／予測」

⑥ 業務種別による変化がない項目

「スケジュール管理」

「データを元に作業計画」

（1）時間記録

②より受講経年による減少傾向が見られる．時間記録は第 1 回基礎講座（PSP）でパーソナルな活動としての実践を強調しているが，以下の様な理由で継続が難しいという意見があった．

・記録に手間がかかる，面倒

・記録できない状況が発生した後，止めた

⑤より SE が定着できていない傾向にある．SE は上流設計や進捗管理，品質管理などの他にも割り込み作業が多い．また定型的な繰り返し作業が少ないため，データを記録しても活かせないという意見があった．

（2）作業時間見積り／測定

①から受講経年による減少はなく，5 割程度が継続している．⑤より SE が定着できていない理由は上記（1）と同様に，定型的な作業が少なく，割り込み作業が多いためと考えられる．

（3）スケジュール管理

全体で唯一 8 割が実践している項目である．①⑥から受講経年や業務種別での変化も少ないことから，本項目は業務として十分に定着していると判断できる．

（4）欠陥記録／予測

③より受講直後から実践者が 5 割と低く，その後 2 割程が途中で止めている．その理由としては，流用／派生開発が多く，個人としての欠陥予測をしなくても開発ができているためと思われる．⑤より SE の割合が低いのは，個人として欠陥記録／予測の作業機会が少ないためと考えられる．

（5）データを元に作業計画

②から受講経年により 2 割程減少となっているが，上記（4）と同様のことが言える．⑥より業務種別による変化は見られない．

（6）プロセス改善

①より受講経年による変化はない．受講直後 8 ～ 9 割と高く，その後 2 割程が途中で止めているが，7 割弱が実践しており，プロセス改善の必要性は指導できていると考える．④より SE やその他の割合が高いのは，チーム全体の事を考える立場の人が多いためと思われる．

上記（1）～（6）の総括と対策案をまとめる．

・「スケジュール管理」「プロセス改善」は概ね定着している

・「欠陥記録／予測」「データを元に作業計画」は実践を意識させる指導が必要である

・経年で減少する「時間記録」「作業時間見積り／測定」は，定着までは上司や職場でのフォローが必要である．

4.3 コースの役立ち状況

表 2 の「C．業務に対する講座の役立ち状況」の回答を「受講経年比較」と「業務種別比較」でグラフ化したものが，図 9，10 である．

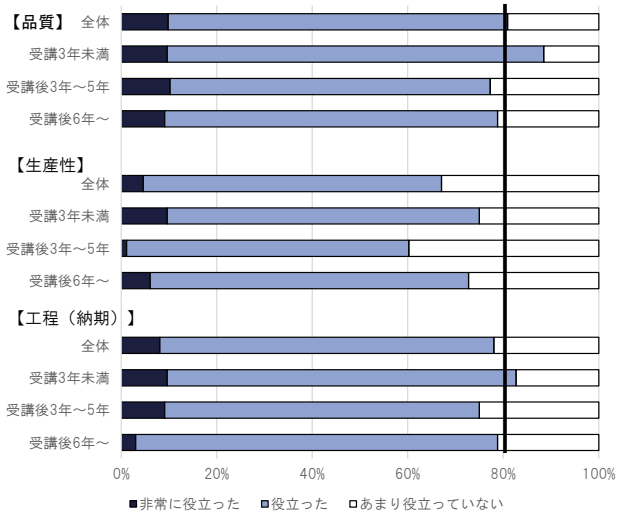


図 9．コースの能力向上役立ち状況（受講経年比較）

図 9「受講経年比較」から①～③のような傾向があった．

① 各項目ともに受講経年による役立ち度に変化は，あまり見られない

②「品質」「工程」は 8 割程度が役立っている

③ 品質→工程→生産性で役立ち度が低下している

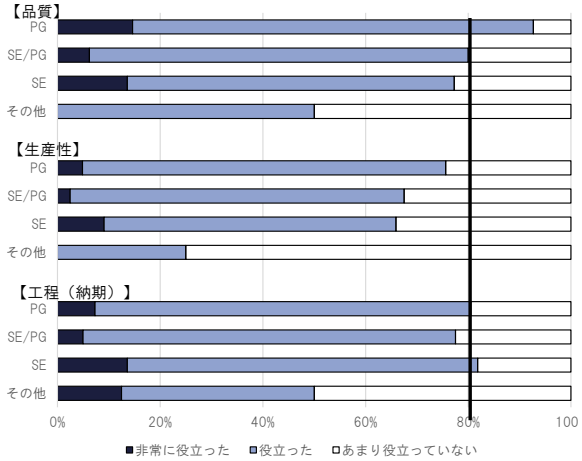


図 10. コースの能力向上役立ち状況（業務種別比較）

図 10「業務種別比較」から，④⑤の傾向があった．

④ 品質と生産性は PG が他の業務種別に比べて役立ち度の割合が高くなっているが，工程では業務種別での関係性は見られない

⑤ 業務種別「その他」はすべての項目が 5 割以下の役立ち度となっている

（1）品質面の能力向上

全体の 8 割以上が「役立った」との回答である．これは「品質は第一であり，納期，価格に優先する」とする品質重視の考えを各回講座で常に意識させた教材と指導を行っているためと思われる．

（2）生産性面の能力向上

全体の 7 割弱が「役立った」との回答である．これは，UML 設計とその支援ツール，テスト項目の自動作成ツールによる演習実施などによるものと思われる．また，SE の役立ち度が低いことが影響して全体が 8 割を下回っている．

（3）工程（納期）管理面の能力向上

全体の 8 割弱が「役立った」との回答である．ツールを活用した工程管理などの指導は講座内で特に行っていないが，PSP で指導した「時間計測」と「自己パフォーマンス理解」が影響していると考えられる．

（4）講座で役立った技法／手法

表 2 の C ④の回答での上位 5 項目が表 3 である．

表 3 から，3 項目がテスト技法に関するものであった．アンケート A の「現在の担当業務」から 76% がテスト業務を担当していることが分かり，受講者が共通的に活用できるテスト技法に回答が集まったものと考えられる．

表 3. 講座で役立った技法／手法

キーワード	割合	指導した講座
同値分割	23%	第4回：テスト技法
UML	10%	第2回：設計技法
PSP	10%	第1回：基礎
境界値分析	10%	第4回：テスト技法
テスト設計技法	9%	第4回：テスト技法

PSP は受講者全員に実践してもらう目的から第 1 回講座で指導し，講座期間中の職場や各回演習で実践をさせていたが，やや低い回答となった．これは，4.2（1）（2）で述べた理由で，継続できていないことが影響していると考えられるため，講義では「3.3 事例発表会」で先輩が取り組んだ実践例や時間計測ツール紹介等の時間を増やすことにした．また，ある自主講座では，新入社員研修で PSP を指導し，本コース受講時点（入社 3 年目以上）で継続していれば，第 1 回講座を免除するなど，継続に向けた取り組みも行い始めている．

設計技法や実装技法は，流用／派生開発の人が多く，本コースで習得した技法を使う機会が少ないため，役立ったと感じられなかったものと思われる．

本コースでは標準的なプロセスモデルの一連を理解することを主眼に指導しており，講座日数の関係上，プロジェクト管理や品質データ分析技法等は十分に指導できていない．今回の結果を踏まえて，生産性向上や職場で役立つ技法を盛り込むことを検討する．

5 今後の取り組み

5.1 スキル判断テストの実施

本コースの第 2 ～ 4 回講座では，個人による設計～テストの演習を行っているが，定時内では終わらず定時後も作業する受講者がいる．その原因は，受講者の設計～テストのスキルが低いことにある．

本コースの受講条件は以下の 3 項目で，受講募集案内にも明記されている．

①入社 2 ～ 5 年目のエントリ層技術者である

② UML の基礎知識がある

③ C，C++，Java のプログラムのコードレビューができる

本コースはソフトウェア技術者の全員教育として位置づけられており，実際は①の条件だけで受講させるケースが多い．これまでの業務で設計～テストを経験していない受講者は，②③の受講条件に達していない場合がある．この

状態で講座を修了しても，講座の内容が身に付かず受講後の職場実践ができないなど，有意義なものにならない可能性がある．

そこで，受講予定者には受講前に受講条件②③に関するスキル判断テストを受けてもらうことを考えている．そのテスト結果で弱点箇所が見つかれば，受講までに自己学習や別に開催している UML，C 言語講座等を受講してもらうこととする．

また，テスト結果が著しく低く，短期間でのスキルアップが困難と上司が判断した受講予定者は，業務経験を積みスキルアップできた段階で受講する運用にする．

5.2 開発環境・ツールの見直し

コースで使用する開発環境・ツールは多くの受講者が使うことを想定して，フリーソフトウェアや自社内製ツールを採用したが，開講から 10 年が経過し，以下のような問題が発生している．

①フリーソフトウェアがバージョンアップされた段階で有償となり，旧バージョンを使用している

⇒ サポート対象外のソフトウェアは，業務用 PC で使用できない

②自社内製ツールが保守対象外になった

⇒ 使い方を習得しても業務では使用できない

この対応として，以下を検討予定である．

・定期講座：当社推奨の市販開発環境・ツールの採用

・自主講座：職場で推奨された開発環境・ツールの使用を許可

職場で使い慣れた開発環境・ツールを講座でも使えれば，演習時間の短縮も見込まれ，メリットは大きい．

また，ある自主講座では仮想環境サーバにソフトウェア開発環境を構築して講座を実施しているところもある．仮想環境を導入するメリットには，以下が考えられる．

・講座に必要なリソースの端末準備／削除が容易になり，事務局の準備作業が効率化される

・仮想ネットワークによる閉じた環境を構築することで，旧バージョンの OS やツール動作が可能となり，逐次アップデートの必要がない

現在はグループ会社も含めた共通の情報インフラ環境が整備されつつあり，将来的には当社グループが共通的に利用できる仮想環境の提供も検討していく．

5.3 教材の e-learning 化

本コースの教材は，各回講座の 1 ケ月前に受講者に配布し予習するよう指導しているが，講座の終了アンケートでは予習実施者は半数にも満たない状況である．また，講座では演習時間が足りないという意見も多いが，講座時間を増やすことは業務とのバランスから難しい．

そこで，教材の重要なポイントを e-learning 化し，講座の前に受講させることを検討中である．受講時に教材内容を理解できていれば，講師が説明を省く箇所ができ，短縮された講義時間をグループ討議や演習に割り当てることが可能となる．

また，本コースの受講対象ではない中堅社員や中途採用の社員にもこの e-learning を受講してもらうことを検討していく．

6 おわりに

本論文では，「ソフトウェア技術者エントリ層教育コース」の改善事例と実施効果の調査結果を述べた．まだ，十分に定着していない項目もあったが，コース受講を通して，ソフトウェア技術者としての能力向上やプロセス改善に取り組む社員が増えつつあることがわかった．

今後も事業環境／技術変化に対応する講座の改善を継続し，開発の実務を担うエントリ層のレベルを向上させ，高品質のソフトウェアを継続的に開発できる組織形成を目指していく．

【参考文献】

[Fujioka2014] 藤岡 卓，田村 直樹，中島 毅，久野　倫義，真野 哲也，ソフトウェア技術者エントリ層に対する職能教育コースの設計と実装，工学教育，Vol.62, No.1, pp.54-60, 2014

[Humphrey2001] Watts S.Humphrey 著，PSP ネットワーク訳，「パーソナルソフトウェアプロセス入門」，共立出版，2001

[Panasonic2017] パナソニック株式会社人材開発カンパニー著，「パーソナルソフト開発作法（ＰＳＰ）初級者研修 第 1.9 版」，組込みシステム産業振興機構，2017

[IPA2007] IPA/SEC 編著，「組込みソフトウェア向け開発プロセスガイド改訂版」，翔泳社，2007

[IPA2013] IPA/SEC 編，「共通フレーム 2013」，オーム社，2013

[IPSJ2017]「特集 ソフトウェア工学の最前線～ソフトウェアが社会のすべてを定義する時代～」，情報処理，Vol.58, No.8, pp.670-707, 2017

SEC 2017年度活動概要

SEC 次長 日下 保裕　企画グループリーダー 江野村 亮輔　企画グループ主任 川原 翔
企画グループ 永井 秀直

IPA は、「社会全体を支える情報処理システムの信頼性向上に向けた取り組みの推進」を第三期中期目標の一つとして掲げ、重要インフラ分野の情報処理システムにかかわるソフトウェア障害情報の収集・分析やソフトウェア信頼性の見える化の促進に向けた活動を通して、我が国のソフトウェアの信頼性向上に寄与すべく活動を推進してきたところである。
近年、IoT、人工知能などの実用化に伴う第四次産業革命と呼ばれる産業構造の転換が世界規模で進みつつあり、今後、技術革新のスピードや、それに伴う社会経済情勢の変化がより一層加速していくことが見込まれている。そこで第四期中期目標期間（2018 年度～ 2022 年度）においては、「ICT に関する新しい流れを常に捉え、発信していく機能の強化」と題して最先端の技術動向や課題をいち早く捉え、新技術の活用法や社会実装上の課題・解決策を速やかに社会の各層に展開してイノベーションを加速していく機能を強化することにより、社会変革の基盤作りへ貢献すべく、ICT の新たな技術などに関する調査･分析及び基準･指針の整備、並びにそれらにかかわる情報発信を推進していく。

1 重要インフラ分野の情報処理システムにかかわるソフトウェア障害情報の収集・分析及び対策

① 重要インフラ分野などにおける情報処理システムの類似障害の再発防止や影響範囲縮小につなげるため、障害情報共有体制の構築を開始し、2014 ～ 2017 年度の 4 年間で 8 分野※1 13 グループの障害情報共有体制を構築した。

② 民間では収集が困難な障害事例情報を収集・分析し、普遍性・一般性のある教訓を導き出し、「情報処理システム高信頼化教訓集」として公開した。更に、障害情報の普遍化を自律的に実施するための「情報処理システム高信頼化教訓作成ガイドブック」、「障害未然防止のための教訓化ガイドブック」などを公開した。

2 ソフトウェア開発データの活用による情報システムの品質･信頼性向上

① 民間では収集困難な機微情報であるソフトウェア開発データを年度平均 245 プロジェクト収集・分析し、「ソフトウェア開発データ白書」として発行した。更に、同白書に掲載したデータを分析して得られた知見を、「ソフトウェア開発データが語るメッセージ」として公開した。

② 組込みソフトウェア分野では世界で初めてプロジェクトデータを収集・分析し、「組込みソフトウェア開発データ白書」として発行した。

3 組込みソフトウェア産業の構造転換に向けた取り組み

組込みソフトウェア産業の構造転換を図るため「組込みソフトウェア産業の動向把握等に関する調査」を経済産業省と協力して実施した。同産業が直面する技術面･人材面･産業面の課題について調査・分析を行うと共に、今後の施策の方向性などを取りまとめて公表した。また、政府の組込みソフトウェア産業技術戦略の企画・策定・実施の PDCA サイクルを回すために設置された、関係省庁・機関による「組込みシステム司令塔会議」に同調査・分析結

※1　情報通信、金融、航空、鉄道、電力、政府・行政サービス（地方公共団体含む）、クレジット、地域団体の計8分野。

果を提供し、「組込みソフトウェア産業戦略」策定に貢献した。

4 利用者視点でのソフトウェア信頼性の見える化の促進

(1) IoT 時代のシステム開発におけるセーフティ・セキュリティの実現

① IoT 製品の利用者や製品のセーフティ・セキュリティを脅かすリスクの発生が懸念されるため、IoT 製品開発者が開発時に最低限考慮すべきポイントを業界横断的に利用可能な全 17 指針として明示した「つながる世界の開発指針」とその解説書などを発行した。同開発指針は、IoT 推進コンソーシアム※2・総務省・経済産業省が策定した「IoT セキュリティガイドライン」に採用されると共に、約 2 年間で 8 つの産業分野・団体※3の標準仕様・ガイドラインなどに反映された。また、IoT 製品・サービスの安全性・信頼性向上に向けた取り組みを継続するべく、「つながる世界の品質確保に向けた手引き」、「『つながる世界の開発指針』の実践に向けた手引き [IoT 高信頼化機能編]」、「つながる世界の利用時の品質」などを公開した。

② IoT 製品やシステムのセーフティやセキュリティを確保するために、日本の主導による国際規格の策定に向けて、IPA の働きかけにより、「IoT セキュリティガイドライン」などの国際標準化にかかわる検討体制を構築した。更に、国際規格の素案を作成し、国際標準化委員会にて新規規格提案の概要を報告した。

(2) ソフトウェア信頼性の見える化促進のための環境整備

高度化・複雑化する IT システムの高信頼化を達成するため、ソフトウェアの上流工程での先進的な設計方法及び検証技術の効果的な適用事例を 58 件収集・分析して公開すると共に、「事例に見る先進的な設計・検証技術の適用分析」として発行した。

5 ソフトウェアの信頼性に関する海外有力機関との国際連携

これまで連携をしている海外代表的機関の米国 NIST※4、米国 SEI※5、米国 MIT※6、独国 IESE※7、英国 MISRA※8 との関係を更に強化した。

① NIST とは毎年定期協議を開催した。2017 年度は、AI※9 にかかわる両者の取り組みを共有すると共に、意見交換を実施した。また、我が国が推進している「IoT セキュリティガイドライン」をベースとした考え方の国際標準化に向けた活動を NIST に紹介した。

② SEI とは連携の一環として、ソフトウェア開発データのメトリクス分析について共同研究を実施している。2017 年度は、最近の IPA のメトリクス分析に関する活動紹介及び共同研究に関する意見交換を実施した。また、共同研究についても、引き続き情報交換を行った。

③ MIT とは、2016 年度及び 2017 年度に IPA が主催した Japanese STAMP※10 Workshop に、STAMP の第一人者である John Thomas 氏を招聘した。また、MIT が主催する STAMP Workshop at MIT に定期的に参加し、協力関係の維持に努めた。

④ IESE とは 2016 年度に欧州におけるシステムズエンジニアリング適用事例の調査契約を締結した。IPA は本調査にて収集した事例 7 件を分析し、ベストプラクティスを含む調査・分析結果を公開すると共に、IESE 専門家を招聘してシステムズエンジニアリングの普及セミナーを開催した。更に、2017 年度はシステムズエンジニアリングの実践を成功させたドイツ企業の事例 2 件を収集した。

⑤ MISRA とは IPA が作成した「【改訂版】組込みソフトウェア開発向けコーディング作法ガイド［C++ 言語版］Ver.2.0」の英語版書籍及び英語版 PDF を MISRA に送付し、改訂にかかわる相互レビューを実施した。また、2015 年度は、MISRA の Andrew Banks 氏と Chris Tapp 氏を招聘し、MISRA C 及び MISRA C++ などを紹介するセミナーを開催した。

※2 産学官が参画・連携し、IoT推進に関する技術の開発・実証や新たなビジネスモデルを創出・推進するために平成27年に設立された組織。
※3 車載器、IoTゲートウェイ、金融端末（ATM）、決済端末（POS）、エネルギー・アグリゲーション・ビジネス、利用時の品質、オープンシステムディペンダビリティ、ORiN3仕様書（仮称）
※4 NIST（National Institute of Standards and Technology）：米国商務省国立標準技術研究所
※5 SEI（Software Engineering Institute）：米国カーネギーメロン大学ソフトウェア・エンジニアリング研究所
※6 MIT（Massachusetts Institute of Technology）：米国マサチューセッツ工科大学
※7 IESE（Institute for Experimental Software Engineering）：独国フラウンホーファー研究機構実験ソフトウェア・エンジニアリング研究所
※8 MISRA（The Motor Industry Software Reliability Association）：自動車メーカ、部品メーカ、研究者からなる欧州の自動車業界団体
※9 AI（Artificial Intelligence）：人工知能
※10 STAMP（System Theoretic Accident Model and Processes）：マサチューセッツ工科大学（MIT）のNancy Leveson教授が提唱した「アクシデントはシステム構成要素間の相互作用から創発的に発生する」という理論

IoT 時代の安全安心に向けて

SEC ソフトウェアグループリーダー　中尾 昌善

1　はじめに

IoT（Internet of Things）時代の到来を迎え、新ビジネス創出の動きが活発化している。一方で、IoT 時代の機器／システム／サービス（以下、これらを総称して「IoT」と呼ぶ）は、従来のような単一的で閉じた範囲で扱われるものではなく、複合的でオープンな環境を対象とするため、その安全安心にかかわるリスクの増大が懸念される。そこで、IPA/SEC では、IoT 開発時のリスクを低減するための活動を推進している。

2　IoT時代の安全安心に向けた活動

(1)　つながる世界シリーズの策定

IoT は、一つの企業単独でなく、異分野の企業の協業によって発展を遂げる可能性がある。その開発においては、品質の保持やリスクへの備えなど、分野を超えて互いに安全安心のための共通認識が必要となってくる。そこで、2016 年 3 月に、IoT 開発時の留意点を分野横断的に取りまとめた「つながる世界の開発指針」を策定した。

2017 年度は、上記の開発指針をベースとして、IoT の品質確保に着目し、開発のみならず、検証及び運用の観点から 13 個の視点を示した「つながる世界の品質確保に向けた手引き」を作成した。これは、IoT に関する取り組みが進む欧米でも見当たらないガイドであり、高い製品品質を売りとする日本特有の先行的なものと位置付けられる。

図 1　つながる世界シリーズ

これを含め、つながる世界シリーズ（図 1）は、欧米とも遜色ないレベルでのガイド類の品揃えが進んでいる。

(2)　産業界への展開

つながる世界シリーズは、業界や各種団体での活用が進んでおり、2017 年度は新たに 4 団体で採用された。また、組込み産業系では、およそ 20% 程度の個別企業での活用、あるいは活用予定という調査結果も出ている。

2017 年度に作成した「つながる世界の品質確保に向けた手引き」では、「つながる世界の品質確保チェックリスト」も併せて公開しており、更なる具体的活用を期待している。

(3)　標準化への提案

「つながる世界の開発指針」を母体とした「IoT セキュリティガイドライン」を、世界の標準規格とすべく提案活動を推進している。ISO/IEC JTC 1 では、テーマ提案の採用に向けて投票が行われた段階である。

(4)　システムズエンジニアリングの推進

IoT による新サービス創出は、ビジネス構造を劇的に変化させる可能性を秘めている。一方で、従来の開発方法論に捉われていると、その変化に柔軟に対応できず、逆にビジネスリスクをもたらす危険がある。そのパラダイムシフトに備えるための開発アプローチとして注目されているのが、システムズエンジニアリングである。

2017 年度は、それを自らが体験するために、パイロットプロジェクトを実施し、そこで得られた知見を「システムズエンジニアリング導入実施の一事例　報告書」として公開した。また、日本国内においてシステムズエンジニアリングを実践し成功した事例を集め、その成功ポイントを分析した「成功事例に学ぶシステムズエンジニアリング」を公開した。具体的事例として参考活用いただければ幸いである。

(5)　先進設計事例の収集と公開

開発現場における先進的な手法適用などの取り組みを、延べ 93 事例収集し、公開してきた。2017 年度は、とくに IoT の開発事例に着目した。

3　おわりに

IoT によるビジネス革新は、安全安心に支えられつつ進行していくことが望ましい。当機構で作成したガイド類がその一助となることを期待している。

IoT時代の安全安心に向けて

「つながる世界の開発指針」の展開状況

SEC 調査役 宮原 真次　SEC 研究員 小崎 光義　SEC 研究員 丸山 秀史
SEC 研究員 西尾 桂子　SEC 研究員 河合 和哉　SEC 研究員 山田 朝彦

1 はじめに

近年、IoT（Internet of Things）の新しい製品・システムやサービスが創出され、IoT を活用した企業の生産性向上や国民生活の利便性が向上している。このような状況の中で、IPA/SEC は IoT の安全安心を確保するための考え方を示した「つながる世界の開発指針」を 2016 年 3 月に公開した。本稿では、この「つながる世界の開発指針」の産業界への展開の状況とその関連施策について報告する。

2 「つながる世界の開発指針」の概要と普及活動

2.1 「つながる世界の開発指針」の概要

IoT では自動車や家電、ウェアラブル機器など様々な「モノ」がネットワークに接続されるが、このような「つながる世界」では利便性は高いものの、遠隔からの攻撃や故障の影響がほかのモノに波及するなどのリスクも高い。そこで IPA/SEC は IoT ならではのリスクに着目し、開発者向けにリスク対策に資する 17 の指針をまとめた（表 1）。

※詳細は、SEC journal 45 号（2016 年 7 月）を参照。

更に、IoT の使われる環境や、様々な利用者に焦点を当て、実際に使うときの「利用時の品質」に着目し、考慮すべき事項を追記した第 2 版を発行した。（2017 年 6 月）

表 1　開発時に考慮すべき 17 の指針

大項目		指針
方針	つながる世界の安全安心に企業として取り組む	指針 1 安全安心の基本方針を策定する
		指針 2 安全安心のための体制・人材を見直す
		指針 3 内部不正やミスに備える
分析	つながる世界のリスクを認識する	指針 4 守るべきものを特定する
		指針 5 つながることによるリスクを想定する
		指針 6 つながりで波及するリスクを想定する
		指針 7 物理的なリスクを認識する
設計	守るべきものを守る設計を考える	指針 8 個々でも全体でも守れる設計をする
		指針 9 つながる相手に迷惑をかけない設計をする
		指針 10 安全安心を実現する設計の整合性をとる
		指針 11 不特定の相手とつなげられても安全安心を確保できる設計をする
		指針 12 安全安心を実現する設計の検証・評価を行う
保守	市場に出た後も守る設計を考える	指針 13 自身がどのような状態かを把握し、記録する機能を設ける
		指針 14 時間が経っても安全安心を維持する機能を設ける
運用	関係者と一緒に守る	指針 15 出荷後も IoT リスクを把握し、情報発信する
		指針 16 出荷後の関係事業者に守ってもらいたいことを伝える
		指針 17 つながることによるリスクを一般利用者に知ってもらう

2.2 展開状況

(1) IoT 政策への展開

国の IoT 政策の一環として設立された IoT 推進コンソーシアムの IoT セキュリティ WG に「つながる世界の開発指針」を提案した。その結果、IoT 機器・システムのセキュリティの指針として「IoT セキュリティガイドライン」に採用された（2016 年 7 月公開）※1。

また、資源エネルギー庁が推進している ERAB（Energy Resource Aggregation Business）のサイバーセキュリティ WG に参加し、開発指針の考え方の重要性を訴えた結果、そのガイドラインに盛り込まれた（2017 年 4 月公開）※2。

(2) 産業界への展開

IoT のセキュリティ強化を目指す業界団体などで「つながる世界の開発指針」の重要性を認識され、その業界でのガイドラインに反映された。

重要生活機器のセキュリティ強化を目指す一般社団法人重要生活機器連携セキュリティ協議会（CCDS）では、「つながる世界の開発指針」をベースにして、車載器、IoT ゲートウェイ、ATM、POS の 4 つの分野のセキュリティガイドラインを作成し公開した（2016 年 6 月公開）※3。

OSD（Open Systems Dependability）の普及を目指す一般社団法人ディペンダビリティ技術推進協会（DEOS 協会）が、開発指針を OSD の視点で解説した報告書をまとめ公開した（2018 年 3 月公開）※4。

FA 機器などをつなぐミドルウェアを提案している ORiN 協

※1　http://www.meti.go.jp/press/2016/07/20160705002/20160705002.html
※2　http://www.meti.go.jp/press/2017/04/20170426001/20170426001.html
※3　https://www.ccds.or.jp/public_document/index.html
※4　http://deos.or.jp/link/obj/pdf/DEOS-TR-20180125.pdf

議会が、開発指針の考え方を現在策定中の ORiN3 仕様にセキュリティ要件として盛り込んだ（2018 年 7 月公開予定）。

(3)　セミナー開催

「つながる世界の開発指針」及び、関連成果に関して、IPA 主催のセミナーや業界団体と協調したセミナー、外部の展示会・セミナーなどで、講演を実施した。これまでの 2 年間で合計 57 回の講演に延べ 782 社に参加していただいた。また、IPA Web サイトからのダウンロード数は約 16,000 回、書籍の配布数は約 8,400 冊の実績となった。

3　関連施策

3.1 開発指針の実践に向けた手引きの策定

「つながる世界の開発指針」を開発現場で実践するために、指針のうち技術面での対策が必要になる部分を更に具体化し、「『つながる世界の開発指針』の実践に向けた手引き［IoT 高信頼化機能編］」を策定した（2017 年 5 月公開）。

この実践に向けた手引きでは、IoT の高信頼化の実現に向けて、ライフサイクルで検討が必要となる、「開始」「予防」「検知」「回復」「終了」の視点で、12 の機能要件と 23 の高信頼化機能をまとめた（表 2）。

※詳細は、SEC journal 49 号（2017 年 7 月）を参照。

3.2 IoT の品質確保に向けた手引きの策定

(1)　背景と概要

IoT は、以下の 3 つの大きな特徴がある。

- システムが日々変化！　接続される機器の種類や個数が膨大で、システムが日々刻々と変化する
- 様々な環境で利用！　屋内／屋外、高地や寒冷地など様々な環境、幼児から高齢者まで幅広い層で利用される
- 10 年以上の長期利用！　自動車・家電製品・工場のシステムなどは、長期に利用される

IoT の品質は、これらの特徴を捉えて、IoT 製品・システムの開発時点での品質確保、更に、出荷後の運用における品質の維持・改善の視点が重要となる。

IoT の品質を検討するにあたり、産業界からの意見や IoT テストの動向、開発指針からの考慮事項など、約 100 件の IoT の品質にかかわる意識を収集した。これらの品質意識に対して、品質を検討する場面（検証計画、妥当性確認、検証、運用計画、運用実施）と品質特性（ISO/IEC 25000：SQuaRE）などを考慮した観点で分析し、13 の品質視点としてまとめ、「つながる世界の品質確保に向けた手引き」として、公開した（2018 年 3 月公開）。

(2)　品質視点の解説例

IoT の品質確保の 13 の視点について、幾つか内容を紹介する。

表 2　IoT の機能要件と高信頼化機能

IoT 高信頼化要件		IoT 高信頼化のための 12 の機能要件	実装に向けた 23 の高信頼化機能
開始	導入時や利用開始時に安全安心が確認できる	1．初期設定が適切に行われ、その確認ができる	初期設定機能、設定情報確認機能
		2．サービスを利用する時に許可されていることを確認できる	認証機能、アクセス制御機能
予防	稼働中の異常発生を未然に防止できる	3．異常の予兆を把握できる	ログ収集機能、時刻同期機能、予兆機能、診断機能、ウイルス対策機能
		4．守るべき機能・資産を保護できる	アクセス制御機能、ログ収集機能、時刻同期機能、暗号化機能
		5．異常発生に備えて事前に対処できる	リモートアップデート機能
検知	稼働中の異常発生を早期に検知できる	6．異常発生を監視・通知できる	監視機能、状態可視化機能
		7．異常の原因を特定するためのログが取得できる	ログ収集機能、時刻同期機能
回復	異常が発生しても稼働の維持や早期の復旧ができる	8．構成の把握ができる	構成情報管理機能
		9．異常が発生しても稼働の維持ができる	診断機能、隔離機能、縮退機能、冗長構成機能
		10．異常から早期復旧ができる	リモートアップデート機能、停止機能、復旧機能、障害情報管理機能
終了	利用の終了やシステム・サービス終了後も安全安心が確保できる	11．自律的な終了や一時的な利用禁止ができる	停止機能、操作保護機能、寿命管理機能
		12．データ消去ができる	消去機能

表 3　IoT の品質確保の 13 の視点

活動	品質の確保、維持・改善の視点
V&V マネジメント	IoT の品質確保のための検証・評価計画立案
	1. IoT の社会的影響やリスクを想定する
妥当性確認	利用者視点での要求の妥当性確認
	2. つながる機能の要求仕様が利用者を満足させるか確認する
	3. 実装した機能が利用者の要求を満たしているか評価する
検証	IoT の特徴に着目したテスト設計
	4. 多種多様なつながり方での動作と性能に着目する
	5. 多種多様な利用環境や使い方に着目する
	6. 障害や故障、セキュリティ異常の検知と回復に着目する
	7. 長期安定稼働の維持に着目する
	8. 大規模・大量データのテスト環境構築とテスト効率化を検討する
	9. テストのしやすさと実施可能性を検討する
	IoT の効率的なテスト実施
	10. テストを効率的に実施し、エビデンスを残す
運用マネジメント	IoT の品質を維持・改善するための運用計画立案
	11. 運用中の環境変化による影響やリスクを想定する
運用実施	長期利用での品質維持と改善
	12. 運用中の環境変化を捉え、品質が維持されているか確認する
	13. ソフトウェアの更新時はつながる相手への影響を確認する

例 1．V&V マネジメント（検証・評価計画の立案）

【視点 1】IoT の社会的影響やリスクを想定する

検証や評価計画を立てるときには、対象製品やシステムの適用分野を理解し、問題が発生したときの社会的な影響やリスクを考慮し、品質の説明責任が果たせる計画の策定が重要である。また、IoT は様々なベンダや構築業者がかかわると想定され、構

築にかかわる関係者間の責任範囲などの合意形成も必要となる。

この視点 1 では、考慮ポイントとして、以下の 4 つの検討事項を説明している。

【1-1】IoT の特徴を考慮した検証・評価の方針を策定する

【1-2】つながる範囲を明確化してリスク・コストを意識しながら検証・評価計画を策定する

【1-3】つなぐ相手や利用者に対して品質を説明できるようにする

【1-4】検証・評価の範囲を明確化し、関係者間の合意を促す

ここでの留意点としては、

IoT 製品・システム開発では、多種多様の購入品が使われるが、仕様とは異なる意図しない動きをする場合もあり（悪意を持ったバックドアーの内包や輸送ルートや倉庫での改ざんなど）、購入品のサプライチェーンでの品質をどのように確認するかが重要なポイントとなる。品質の説明責任をきちんと果たすためにも、購入品に対する品質の確認方針を決めておくことが重要である。

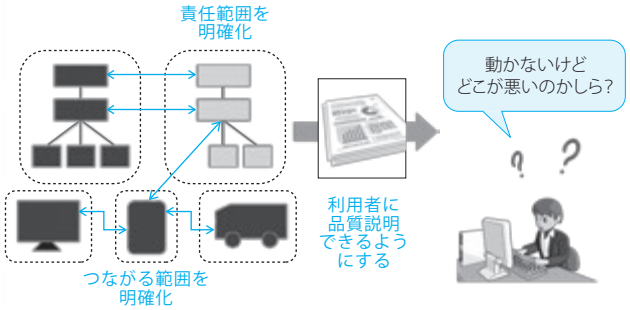


図 1　検証・評価計画の立案

例 2. 妥当性確認（要求仕様のレビュー）

【視点 2】つながる機能の要求仕様が利用者を満足させるか確認する

IoT では利用者や利用環境の想定が難しく、想定外の利用者や利用環境で使われる可能性がある。多様な利用者や利用環境の変化に対して、本来提供したい価値を継続して提供できるか、要求仕様そのものの妥当性を確認することが重要となる。また、要求仕様に明確に書かれていない暗黙的な要求（非機能要件など）も対象として、妥当性の確認が必要である。

この視点 2 では、考慮ポイントとして、以下の 4 つの検討事項を説明している。

【2-1】IoT 特有の機能や性能、互換性や拡張性に着目する

【2-2】利用環境や利用者の使い方に着目する

【2-3】IoT のライフサイクルでの安全安心に着目する

【2-4】長期利用のための保守・運用に着目する

ここでの留意点としては、

IoT 製品・システム開発では、そもそも要求仕様が IoT の特徴を十分考慮しているかに着目して、具体的な設計に着手する前に仕様の妥当性確認が重要となる。とくに、つながることによるリスクを考慮し、安全安心を維持するための機能（故障やセキュリティ異常の検知、ログ収集、縮退・回復やリモートアップデートなど）が、要求仕様として明確になっており、それらの想定が妥当であることの確認が重要である。

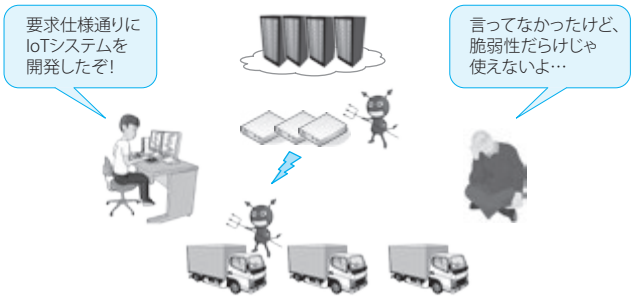


図 2　利用者視点で要求・要件の妥当性を確認

3.3 開発指針の国際標準化活動

(1) 概要

これまで述べた IPA/SEC の成果は世界に先がけたものであり、IoT 社会の安全安心の確保のためのセーフティ・セキュリティの基準は、国際的にもそのニーズが認識されていることから、「つながる世界の開発指針」とこれを母体とした IoT 推進コンソーシアムの「IoT セキュリティガイドライン」の国際標準化に取り組んでいる。具体的には、一般社団法人情報処理学会に設置された対応委員会の活動を通じて、情報技術の国際標準を開発している ISO/IEC JTC 1 の SC 27（IT Security techniques）と SC 41（Internet of Things and related technologies）に国際標準規格として提案活動を推進している。

(2) SC 27 における標準化活動

「IoT セキュリティガイドライン」を基に SC 27/WG 4 に対して提案活動を推進している。規格開発項目として "Guidelines for security and privacy in Internet of Things（IoT）" を提案して行われた投票の結果、新規規格開発項目としての成立条件を満足した。今後は、規格開発項目として正式に成立させ、国内対応委員会において積極的に提案開発を推進していく。

(3) SC 41 における標準化活動

「つながる世界の開発指針」と内閣サイバーセキュリティセンターが 2016 年 8 月に策定した「安全な IoT システムのためのセキュリティに関する一般的枠組」を基にして SC 41 に、規格開発項目として "Methodology for implementing and maintaining trustworthiness of IoT systems and services" を提案し、2018 年 3 月時点で投票期間中である。今後は、規格開発項目として正式に成立させ、国内対応委員会において積極的に提案開発を推進していく。

4 今後の活動

2018 年度は、つながる世界シリーズを各地域や中小企業へ展開を加速すると共に、国際標準化の提案活動を推進し、我が国の IoT の安全安心に向けた基盤を確立させたい。

IoT時代の安全安心に向けて

システムズエンジニアリングの推進

SEC 主任研究員　端山 毅　SEC 研究員　齊藤 善治　SEC 研究員　齋藤 毅

1 はじめに

近年の製品／サービスは、多数の装置が接続されたり、利害関係者が多くなったりして、多様性と不確実性が増し、個々の技術だけでは実現が困難になっている。そのような状況では、システム全体を俯瞰し、相互作用に着目しつつ統合管理するシステムズエンジニアリングのアプローチが有効である。IoT などの新しい製品／サービスの開発を成功に導くことに寄与し、産業界の競争力強化を図ることを目的として、2015 年度にシステムズエンジニアリングを推進する活動を開始した。

まず、国内外の事例調査や関連技術調査を行い、2016 年度にシステムズエンジニアリングの認知と重要性の認識を促すための啓発書を公開した。

2017 年度は、日本企業の事例を収集し、システムズエンジニアリングの主要なアプローチや考え方がどう活用されているのか分析した。この分析結果を用いて、絡み合った事情の中で解決策にたどり着く過程を紹介し、システムズエンジニアリングのアプローチがどのように役立つのか、実践的な解説書を作成した。

加えて、システム開発の現場でシステムズエンジニアリングの導入を目指してパイロットプロジェクトに取り組んだ。

2 システムズエンジニアリングの有用性の発信

経営者と開発現場の課題意識を喚起し、新しい開発アプローチの必要性を説明するため、啓発資料を作成した。

「経営者のためのシステムズエンジニアリング導入の薦め」（2017 年 3 月）

「開発者のためのシステムズエンジニアリング導入の薦め」（2017 年 5 月）

これらの文書では、多岐にわたるシステムズエンジニアリングの知見の中から、とくに重要な 4 つのポイントを抽出した。

① 目的指向と全体俯瞰

② 多様な専門分野を統合

③ 抽象化・モデル化

④ 反復による発見と進化

3 事例に基づくシステムズエンジニアリングの解説

システムズエンジニアリングは、欧米の航空・宇宙分野など高度な専門性を要する複雑なシステム開発で培われた知見を体系化したものである。機械工学、電気工学、ソフトウェア工学など、多数の技術領域を横断する体系として抽象的に整理されている（図 1）。IoT 化の潮流の中で、様々な製品／サービスの開発にシステムズエンジニアリングのノウハウを活用していくことに焦点を当てる場合、厳密さを多少失うとも、事例ベースで実務者の共感を得つつ、実践的な理解を広めることにした。

この趣旨に沿って、昨今の技術や社会の変化の中で生じた日本企業の事例に基づき、システムズエンジニアリングの主要な視点やアプローチの効果を、実践的な観点で分析・解説して、

「成功事例に学ぶシステムズエンジニアリング～ IoT 時代の

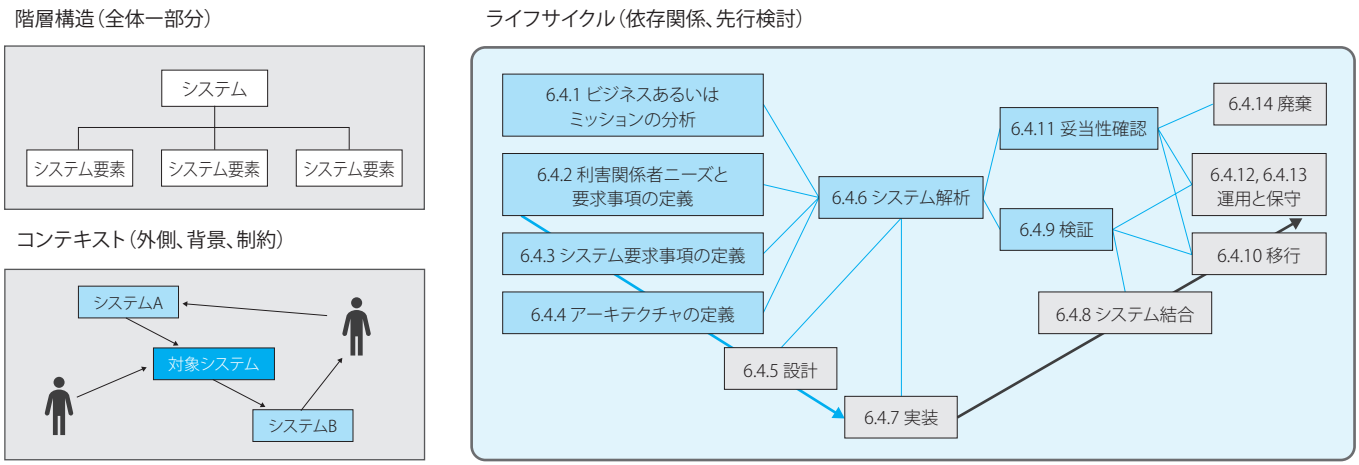


図 1　システムズエンジニアリングの基本的な概念

システム開発アプローチ～」（2018 年 3 月）

を公開した。本書は、システムやサービスの企画・開発に取り組もうとするマネジメント層、リーダー、担当者向けに作成した。

本書で取り上げた事例は、事業的観点から目的を達成した成功事例であるが、意図的にシステムズエンジニアリングを適用した案件ではない。しかし、その課題解決の過程では、システムズエンジニアリングが唱道しているアプローチが実践されており、その有効性の理解を促進するにふさわしい内容であった。また、事例の中で解決策が模索された課題は、今後のシステム開発に共通するテーマを含んでいる。

システムズエンジニアリングを構成する一般的な定義や活動については、システムライフサイクルプロセスの国際規格である ISO/IEC/IEEE 15288:2015 をよりどころとした。この中から、とくに本書の内容と関連性の高い 7 プロセスを抽出し、各事例において実施された活動が、どのプロセスに該当するかを確認した。

表 1 の縦の項目は、ISO/IEC/IEEE 15288:2015 のプロセスであり、横の項目は我々が抽出した 4 つのポイントである。この表に事例の活動を当てはめると、一般的なシステムズエンジニアリングのプロセスに対比しつつ、4 つのポイントで挙げた観点での意義を確認できる。

本書で取り上げた事例の概要を表 2 に示す。

4 システムズエンジニアリングのアプローチの試行

三菱重工機械システム株式会社と、「システム開発の上流工程にシステムズエンジニアリングの考え方を導入する取り組み」を共同で実施した。

今回の主な取り組み内容を以下に示す。

① 開発対象の全体をシステムとして捉え、「目的指向と全体俯瞰」の考え方を徹底する

②「システムライフサイクルのプロセス」の重要なポイントを理解し、自組織の開発標準と比較する

③ 開発の上流工程で「妥当性確認」を行う

④ パイロット活動の取り組みを整理し、今後の開発に向けた提言をまとめる

これらの活動は、開発チームが新たな取り組みを導入する活動につながると共に、以下の知見が得られた。

A)「システムライフサイクルのプロセス」を理解し、開発標準と比較することで、自部門の有識者の暗黙知を整理評価して、新たな開発標準の運用に活用できる

B) 上流工程で、プロジェクト本来の目的とシステムの設計内容の整合性を確認することで、問題の早期発見につなげられる

表 1　ポイント×プロセス対応表

注）下記 6.4.1~6.4.11 は、ISO/IEC/IEEE 15288:2015 の章節番号である。		ポイント			
		目的指向と全体俯瞰	多様な専門分野を統合	抽象化・モデル化	反復による発見と進化
プロセス	6.4.1 ビジネスあるいはミッションの分析				
	6.4.2 利害関係者ニーズと要求事項の定義				
	6.4.3 システム要求事項の定義				
	6.4.4 アーキテクチャの定義				
	6.4.6 システム解析				
	6.4.9 検証				
	6.4.11 妥当性確認				

表 2　事例一覧

地域活性化イベント向け情報共有基盤の開発
多様な関係者を巻き込み、ステークホルダのニーズと要求を明確化し、全体を俯瞰して段階的に集客イベントを支える情報共有基盤を開発、拡張し、継続的な地域活性化活動につなげた。 （株）富士通総研
電子お薬手帳システムに適用したセキュリティ設計
目的指向と全体俯瞰によりセキュリティの課題を抽出し、医療と IT にまたがる複雑な問題に対して、抽象化・モデル化を活用した系統的なアプローチでセキュアなシステムを実現した。ソニー（株）
多様な要求を満足させる自動車エンジンの開発
自動車エンジンをシステムと捉え、全体最適の観点から機能目標を定義し設計した。部品の物理設計に先行して機能開発を行い、検証も含め効率的に開発を進め、大幅な燃費向上などの目標を達成した。マツダ（株）
首都圏の高密度鉄道輸送を支えるデジタル ATC の開発
2 世代先まで見通し段階的に開発し、デジタル ATC（Automatic Train Control）を実現した。移行／運用も視野に入れて課題を見据え、要件や設計に反映し、試験時間帯の制約などを克服した。東日本旅客鉄道（株）
Web スキャンシステムの企画開発
一段高い視点から俯瞰し、ビジネス分析及びステークホルダ要求分析を行って、スキャナーの新たなクラウド連携サービスを実現した。キヤノン電子（株）

このパイロット活動の結果は、

「システムズエンジニアリング導入実施の一事例　報告書」（2018 年 3 月）

として報告書を公開した。

5 今後の活動予定

今後、システムズエンジニアリングのアプローチは、多くの産業分野において役立つと考えられる。これまでに収集、整理した事例及び解説を広める活動を、IPA 自身のみならず、関連する団体、大学、企業などの協力も得ながら進め、引き続き産業界から事例や課題を収集しつつ、実践的な情報提供を継続する。

IoT時代の安全安心に向けて

先進的な設計・検証技術の適用事例収集・公開

SEC 研究員　遠藤 秀則　　SEC 研究員　佐々木 方規

1 はじめに

システム及びソフトウェアの開発現場では、「品質確保」「生産性向上」「安心・安全な運用」など様々な課題を解決すべく先進的な開発技術の適用に積極的に取り組んでいるプロジェクトがある。それらのプロジェクトの事例から、課題解決のための独自の工夫など、実践的な取り組み内容を紹介することにより、我が国のシステム及びソフトウェア開発の高信頼化に寄与したいと考える。

IPA/SEC では 2013 年度より先進技術や手法を活用したベストプラクティスの収集を開始し公開してきた。昨年度は 2018 年 2 月に「先進的な設計・検証技術の適用事例報告書 2017 年度版※1」として新たに 16 事例を公開、2013 年度からの公開事例数は、延べ 73 社、93 件となった。紹介している事例の内容として、「課題解決に向けて、どのような視点で先進技術を導入したのか」「導入する際、どのような工夫や苦労をしたのか」が記述され、いずれも開発現場で実践されたものばかりである。こうした事例を参考にすることで、読者は自社の事業領域や開発課題などと照らし合わせ、自社の導入に役立てることが可能になると考えている。

2 事例の収集

2017 年度公開した事例 16 件を表 1 に示す。2013 年の収集開始時点では、「高信頼化」のお手本になる事例が主であったが、2017 年度は「高信頼化」のみならず、「変化する要求への柔軟な対応」「開発速度と品質の両立」「開発者視点」だけでなく「利用者視点」での取り組みなどの事例も積極的に収集した。近年、企業における IT システムの役割は、利便性を求めるものから、IT システムを駆使したイノベーションでビジネスを創出するものへと変化している。このような環境では、実現したい要件を模索しながら開発を行う技術や、要件の変化を素早く開発にフィードバックする手法が求められている。更に、マイクロサービスやハイブリッドクラウドなど、従来よりも複雑な環境が登場している。このような時代の変化に合わせ参考となる先行事例を収集した。

3 活用実績

本活動も 7 年目となり、公開事例を見て自社に先進技術や手法などを適用した実績、また自社へ導入した際に工夫したノウハウなどを更に事例として提供していただき、再び公開するといった好循環も現れてきた。更に、グループ会社や業界団体にて公開事例の活用方法を検討する勉強会や一般財団法人日本科学技術連盟でスタートした ODC 分析研究会※2のように、本活動の公開事例をトリガとしたコミュニティ発足など自律的な導入の機運は高まってきている。また、セミナーやイベントなどの紹介機会を設けたところ多くの方々に参加いただけたこと、参加者から事例活用の紹介があったことなどから、開発現場でも参考情報として利用されていることが推察される。

4 今後について

これまでの収集公開活動で、手法、技術は当初計画したすべての項目を網羅したこと、普及活動により業界や企業単位で自律的な導入の動きが進んできたことから、独立した施策としての先進事例収集公開活動は 2017 年度で終了とすることにした。今後は IPA/SEC のそれぞれの活動の中で事例収集、公開活動を実施していく。開発現場で抱える問題に対してソフトウェア・エンジニアリングの活用で解決しているベストプラクティスを SEC 各施策の中で収集公開していく。

※1　先進的な設計・検証技術の適用事例報告書2017年度版　　https://www.ipa.go.jp/sec/reports/20180228.html
※2　一般財団法人日本科学技術連盟ODC分析研究会　　https://www.juse.or.jp/sqip/odc_workshop/index.html

表１　先進的な設計・検証技術の適用事例一覧（2017 年度版）

事例番号	タイトル	事例提供元	キーワード
78	ソフトウェア開発組織の効率的な品質改善に向けたプロセスデータの活用	日本電気株式会社	プロセス改善 CMMI
79	開発プロジェクト期間中での利用時品質の評価プロセスによるソフトウェア開発へのプロアクティブなフィードバックプロセスの提案	ウイングアーク１ｓｔ株式会社	JIS X 25000 シリーズ規格（SQuaRE） 利用時品質 ソフトウェア品質評価
80	GSN を活用した技術者能力計測手法の提案	国立研究開発法人宇宙航空研究開発機構	GSN IV&V
81	統合モデル技術によるユーザ主体のソフトウェア開発手法 「モデル指向開発（MOD）」の紹介 MOD:Model-Oriented development	株式会社日立産業制御ソリューションズ	MBD MDD XDDP SPL
82	楽しいシステム開発、失敗しないシステム開発を実現する要件定義工程の進め方	株式会社ウイング	超高速開発 プロトタイピング 自動生成ツール 要件定義 ユーザ開発・保守
83	エンタープライズシステムへのマイクロサービスアーキテクチャー適用の実践	東芝デジタルソリューションズ株式会社	マイクロサービス API エコノミー
84	業務アプリケーション改修時の XDDP 適用事例 ～品質の見える化による、効果的なプロセス改善の実践～	株式会社両備システムズ	XDDP メトリクス プロセス改善
85	派生開発での時間効率性劣化を変更要求から検出する方法	日本科学技術連盟　ソフトウェア品質研究会	派生開発 品質保証
86	ヒューマンエラーによる失敗・事故の分析手法 ～医療分野の分析手法を基にした SE 事故分析プロセスの提案～	株式会社日立製作所	ヒューマンエラー 事故分析
87	レビュー会議の可視化により目的の曖昧さを明確にする手法 ～ソフトウェア開発現場への TMBRI 法の導入～	株式会社モバイルインターネットテクノロジー 株式会社インテック 株式会社東光高岳 GE ヘルスケア・ジャパン株式会社 ソーバル株式会社	レビュー
88	Session Based Test Management による探索的テストの実践 ～受託開発でも探索的テストを管理し活用できる～	株式会社エヌ・ティ・ティ・データ	Session Based Test Management 探索的テスト
89	要因組合せによる大量のテスト項目実施における障害の早期検出および工数削減の取り組み	富士通株式会社	テストオートメーション テスト項目自動生成 障害分析
90	「影響波及パス分析法」の適用事例 ～統合テストでの影響範囲に対するテスト漏れ防止～	株式会社デンソークリエイト	派生開発 影響分析 ソースコード解析
91	利用時品質を高めるための開発プロセス ～デザインエンジニアリング～	エスディーテック株式会社	利用時品質 UX 人間中心設計 HMI 検証
92	大規模・長期的な製品シリーズの開発に有効なソフトウェア開発手法 「モデル指向 SPL 開発」の紹介	株式会社日立産業制御ソリューションズ	MBD MDD XDDP SPL
93	中大規模エンタープライズシステムに適用可能なアジャイル開発手法	JBCC 株式会社	アジャイル開発 自動生成ツール ユーザ開発・保守

重要インフラ分野等システム / 製品の障害対策

SEC 調査役　三縄 俊信　SEC 研究員　目黒 達生　SEC 研究員　村岡 恭昭
SEC 研究員　齋藤 毅　SEC 調査役　三原 幸博　SEC 研究員　松田 充弘
システムグループリーダー　山下 博之

2013 年度から 2017 年度まで、重要インフラ分野などのシステム / 製品の障害事例からヒアリングなどにより情報を収集し、その根本原因の分析と再発防止策の検討を行った。その結果をもとに、IT サービス分野については産業分野横断で活用可能な普遍的な教訓を 50 件、組込み分野については多様な組込み製品や制御システムに適用可能な教訓を 35 件、それぞれ作成し、教訓集に取りまとめて公開した。また、IT サービス分野において障害事例情報を共有する仕組みの構築に向けた支援活動を行い、7 つの産業分野・地域で計 13 件の情報共有の仕組みが運用を開始した。

1　背景

情報処理システムは、銀行や証券などの金融サービス、各種手続きのための行政サービス、ソーシャルネットワークなどの情報通信サービス、交通機関の運行制御など、私たちの生活や社会・経済基盤を支える重要インフラ分野などの IT サービスに深く浸透しており、ひとたび障害が発生するとその影響は非常に広範囲に及ぶ。私たちが安全で安心な生活や社会・経済活動を続けるためには、重要インフラなどを支える IT サービスの一層の信頼性向上が求められている。

IPA では、社会インフラに影響を与えマスコミなどで報道された IT サービス障害情報を継続的に収集している。そのデータによれば、IT サービス障害の発生状況は調査を開始した 2009 年から増加傾向にある（図 1）。

また、家電製品や自動車をはじめ、機能の大半がコンピューターを利用してソフトウェアで実現されている組込み製品や制御システムにも、社会インフラの重要な役割を担うものが多くあり、これらは、実現する機能の増加と共に複雑化する傾向にある。組込みシステムが急速に IoT に発展しつつある今日では、様々なシステムとの相互接続を前提に信頼性を確保するための技術面や運用管理での更なる工夫が求められている。

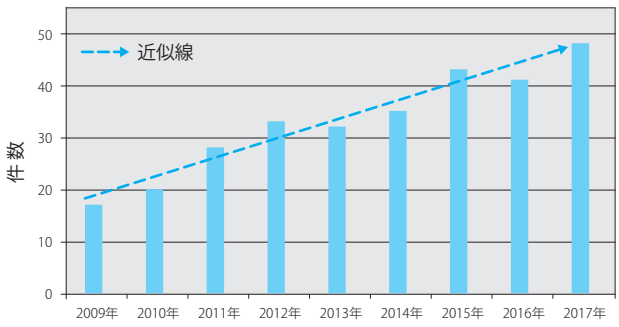


図 1　報道された IT サービス障害の発生件数の推移

従来、IT サービスや組込み製品などの情報処理システムの障害に対する原因分析と再発防止策の実施は、多くの場合、当事者においてのみ行われ、その情報はほとんど公開されてこなかった。そのため、当事者以外の同種システム、あるいは他業界・分野のシステムにおいて、類似の障害が発生することがあった。

情報処理システムの開発 / 構築・運用やその管理は、社会や技術の進展につれて複雑化・多様化しており、一個人や一企業でカバーできる範囲には限界がある。そして、その複雑性・多様性が今後ますます拡大していくことは明らかである。このような情報処理システムの構築・運用及びその管理にかかわる信頼性面での課題を解決するためには、より多くの人々・企業の経験を社会全体で共有・伝承することが求められてきている。

このような背景から、IT サービスや組込み製品・制御システムの障害事例情報の分析や対策手法の整理・体系化を通して得られる「教訓」を業界・分野を超えて幅広く共有し、類似障害の再発防止や影響範囲縮小につなげる仕組みの構築に向けた活動を 2013 年度から実施している。

2　障害事例の収集と教訓化

「重要インフラ IT サービス高信頼化部会※1」及び「製品・制御システム高信頼化部会※2」（以下、部会）の活動を通じて障

※1　重要インフラITサービス高信頼化部会：銀行、保険、証券、電力、鉄道、情報通信、政府・行政などの重要インフラ分野における情報処理システムの有識者・専門家で構成する委員会
※2　製品・制御システム高信頼化部会：重要インフラ分野における製品・制御システムに関する有識者・専門家で構成する委員会

害事例を収集し、障害発生の原因分析を行い普遍化した上で、前者で計50件、後者で計35件の教訓を取りまとめ、「情報処理システム高信頼化教訓集（ITサービス編）2017年度版※3」（以下、教訓集2017）、及び「同（組込みシステム編）2015年度版※4」を公開した。

また、2010年から2017年まで収集蓄積している報道されたシステム障害の一覧及び教訓集2017の各教訓を利用者が検索しやすいように、99件の「注意すべき観点」を整理し、これに基づく分類を行った一覧表を公開した。また、上記の結果をもとに「注意すべき観点」に基づいた障害事例の10種の分類を教訓集2017に掲載した。

3 障害未然防止のための設計知識の整理

組込みシステムを開発する企業の多くは、過去の障害事例（過去トラブル）を記録し障害情報データベースとして蓄積している。一般に「過去トラ（DB）」と呼ばれており、そこには障害再発を防止するためのノウハウが含まれている。「製品・制御システム高信頼化部会」では、このノウハウを設計知識の形で取り出すことができれば、開発に直接携わる組込みソフトウェア設計者のために役立てられると考え、「障害未然防止のための設計知識の整理手法ガイドブック」として公開した。

4 普及展開活動

① 教訓集ダイジェストを用いた普及

システム障害情報共有の普及活動を推進するために、ITサービス編及び組込みシステム編の情報処理システム高信頼化教訓集に収録された教訓を一覧化して、収録内容を容易に検索できるようにした小冊子「情報処理システム教訓集ダイジェスト2017年度版」を作成・公開した。

② 教訓集活用メールマガジンの配信

情報処理システム高信頼化に関する情報「教訓集活用メールマガジン」の配信希望者（約1,300名）に対して、部会委員が寄稿した障害情報共有の取り組み記事（表1参照）や、新たな教訓の解説記事などを毎月配信することにより、教訓活用を促進した。

表1　配信された委員からのメッセージ

記事タイトル	NO.
システム障害のマネジメントについて	10号
プロジェクトのゴールはサーバの火を落とすとき！	11号
サービスを継続的に提供するためには	12号
重要インフラITサービスを支えるための政府の取組	13号
過ちて改めざる、是すなわち過ちという	15号
トラブル対策にも新たなチャレンジを	18号
「情報処理システム高信頼化教訓集」に学ぶ	19号
トラブルは品質改善の種	20号

5 システム障害情報共有の仕組み構築

各業界団体などにシステム障害情報の共有の仕組み構築を働きかけ、7つの分野・地域で、計13の情報共有グループが構築され（図2）、その運営を開始した。また、幾つかのグループについて意見交換会を開催した。

図2　2017年度までに構築された情報共有グループ

6 今後の予定

2018年度以降も引き続き情報処理システムの高信頼化に向けて有益な情報発信を行う予定である。また、システム障害事例情報の共有については、自律的な情報共有の促進に向けた各種の支援を継続する予定である。

※3　URL: https://www.ipa.go.jp/sec/reports/20180326.html
※4　URL: https://www.ipa.go.jp/sec/reports/20160331_2.html

定量的管理による信頼性・生産性向上

SEC 研究員　峯尾 正美　SEC 専門委員　佐伯 正夫　SEC 調査役　三原 幸博
SEC 研究員　松田 充弘　SEC 研究員　田代 宣子　システムグループリーダー　山下 博之

SEC 設立以来、定量的に管理されたソフトウェア開発データを業界から広く収集・蓄積し、ソフトウェアの信頼性・生産性などの観点で統計分析した結果を「ソフトウェア開発データ白書」として公開している。白書は毎年、エンタプライズ系と組込み系とを交互に発行してきた。また、定量的管理の推進に向け、データの深掘り分析に基づく知見の発信やガイドブックの発行、セミナーなどを実施した。

エンタプライズシステム分野

近年、ソフトウェアの大規模化／複雑化が進む一方、信頼性向上、生産性向上、開発期間短縮などの要求はますます高まっている。このようなニーズに適切に応えるには、開発プロジェクトの定量的管理が基本であり、とくに、自組織で蓄積した定量的データ（ベンチマーク）をもとに、見積もりや開発計画の妥当性を評価したり、他組織の良い点を学び、自プロジェクト、組織の改善を図ったりする「ベンチマーキング」が重要になる。

このような要求に応えるべく、IPA/SEC では、公開ベンチマークとして利用してもらえるよう、国内の主な開発ベンダ約 30 社から収集したデータを統計分析し、「ソフトウェア開発データ白書」（以下、白書）として 2004 年度より定期的に刊行してきた。また、白書データをより深堀りして得られた知見を、「ソフトウェア開発データが語るメッセージ」の形で公開してきた。

1 「ソフトウェア開発データ白書」

1.1 白書 2016-2017

図 1　ソフトウェア開発データ白書 2016-2017

【発行】 2016 年 10 月 1 日

【収録データ件数】 4,067 件

【特長】

① 生産性、信頼性の変動要因候補を多面的に分析

② 工程ごとの成果物量を分析

③ 生産性、信頼性の主要な変動要因である「業種」ごとに データを分析した「業種編 3 種」を発行

1.2 白書 2018-2019

【発行】 2018 年 10 月（予定）

【収録データ件数】 4,564 件

[ただし、統計データは近年 6 年間のデータ（1,475 件）から算出]

【特長】

① 生産性や信頼性が経年推移している状況を考慮して、最近の開発状況に応じたベンチマークを提供するため、掲載する統計値は、近年 6 年間のプロジェクトデータを対象に算出

② 主なデータの経年推移を掲載

③「業種編（金融・保険業編、情報通信業編、製造業編）」の統計情報掲載項目を本編と同等とし、拡充

2 「ソフトウェア開発データが語るメッセージ」

2.1 メッセージ 2015 ～プロジェクトや組織のマネジメントの指針と信頼性・生産性の傾向～

【分析結果から得られた知見例】

① 設計レビュー工数比率が低いと、信頼性※1 が低くなる傾向がある。とくにレビュー工数比率 2%未満ではその傾向が顕著である。一方、レビュー工数比率が 7%以上の場合、信頼性が低いものはわずかである

② 設計文書化密度が高くなるにつれて、設計レビュー指摘密度が高くなる傾向がある

※1：信頼性は、出荷後の発生不具合密度で示す。

③ テスト密度が高くテスト検出不具合密度が低いことは、相対的に信頼性が良い兆候の一つである

【ソフトウェア開発者に向けたメッセージ例】

設計レビュー工数比率が 7%以上となるよう設計レビューを強化しよう。ただし、大きくし過ぎても効果／コストは頭打ちとなる

ドキュメントを増強して、効果的な設計レビューを実施しよう。具体的には、文書化密度の中央値（白書データでは、15.8[頁／ KSLOC※2] 以上となるよう増強しよう

テスト密度を中央値（白書データで 30.8[ケース／ KSLOC]）以上に、テスト検出不具合密度を中央値（白書データでは 1.60[件／ KSLOC]）以下に設定してテストを評価しよう

2.2 メッセージ 2016 ～設計レビュー・要件定義強化のススメ～

【分析結果から得られた知見】

① 上流工程（基本設計～製作）での不具合摘出比率※3 を高めることによって信頼性向上が期待できる

② 要件定義を強化することによって信頼性向上が期待できる

【ソフトウェア開発者に向けたメッセージ】

①-1 プロジェクト計画や品質マネジメント改善などのシーンにおいて、上流工程（基本設計～製作）での不具合摘出比率を、目安として 85%程度に高めるような開発スタイルを目指そう

①-2 上流工程での不具合摘出比率を高めるには、特に設計レビューを質・量で充実させることを目指そう

②-1 プロジェクト計画／再計画や品質マネジメント改善などのシーンにおいて、要件定義を質、量共に強化することを目指そう

②-2 ユーザの協力を得る（ユーザ担当者の要求仕様関与を高める）ことによって、要求仕様をより明確にしよう

2.3 メッセージ 2017 ～生産性・信頼性の経年推移の分析から～

【分析結果から得られた知見】

① ソフトウェア生産性（SLOC 生産性）は全体的に低下傾向にある

② ソフトウェアに対する品質要求は高まっている

③ 生産性を低下させないポイントは上流工程強化にある

【ソフトウェア開発者に向けたメッセージ】

① 定量的管理を推進し、生産性の経年推移を踏まえて生産性目標を設定しよう

② 定量的管理を推進し、品質要求レベルに見合った生産性目標を設定しよう

③ 業務分野経験などのスキルが高い要員を育成しよう

※2：Source Lines of Code　ソースコードの行数
※3：上流工程でのレビュー指摘件数÷（開発工程全体でのレビュー指摘件数＋摘出不良件数）

組込みシステム分野

IPA/SEC は、組込みソフトウェア開発企業が保有するプロジェクト管理データを分析して、組込みソフトウェアの信頼性や生産性の指標を公開する活動を 2013 年に開始した。活動の狙いは、見えないと言われてきた組込み開発の現場に「見せる化」する文化を醸成し、スキルの高い技術者が個人で抱え込むリスクを共有して組織的な開発に変えていくことにあった。一方で、プロジェクト管理データの収集が定着している企業にとっては、企業内で個別に分析するだけでなく、公的な機関にプロジェクト管理データを提供して、将来を見据えた産業界の知識データベース作りを期待する思いも感じられる。

図 2　組込みソフトウェア開発データ白書 2017

1 「組込みソフトウェア開発データ白書 2017」

2015 年の初版発行に続き、416 件のデータを分析した「組込みソフトウェア開発データ白書 2017」を 2017 年 11 月に発行した（図表 2）。2017 年版の分析では、リアルタイム性要求の強弱の違いなど製品特性によって、生産性やバグ密度の傾向が変わることを定量的に裏付けることができた。この分析内容の詳細については、SEC journal51 号で紹介している。

2 プロジェクトマネジメントガイド［定量データ活用編］の発行

組込みソフトウェア開発業界に、定量データ活用の文化を広めるためには、管理データの活用方法や活用してどんなメリットがあるのかを伝える必要がある。それを伝えたのが本冊子であり、2015 年 11 月に「組込みソフトウェア開発データ白書 2015」と併せて発行した（図表 3）。

図 3　組込みソフトウェア向けプロジェクトマネジメントガイド［定量データ活用編］

システム構築能力の強化

SEC システムグループリーダー　山下 博之

IT の利用拡大と IT システムの複雑化・大規模化が進展し、IT システムの信頼性・安全性は一層重要となっている。しかしながら、システム構築プロジェクトの失敗はなくならないばかりか、深刻さを増すセキュリティ上の課題に新たな対応を迫られている。こうした背景から、IPA/SEC では、システム構築能力の強化に取り組み、過去 5 年間に次の活動を行った：要件定義などシステム構築上流工程の強化、システム理論に基づく安全性解析手法の普及展開、制御システム向けのセーフティ・セキュリティ対策の検討、コーディング作法ガイドの整備。

システム構築能力の強化

上流工程の課題解決に向けて

SEC 研究員　山本 英明　SEC 研究員　村岡 恭昭　SEC システムグループリーダー　山下 博之

システム構築上流工程の作業不備による開発プロジェクトの失敗や運用後のシステムトラブルがなくならないという背景から、要件定義と再構築の 2 つの課題に取り組んだ成果をまとめガイドブックと小冊子を発行した。また、非機能要求の考慮漏れによる手戻りをなくすために、非機能要求グレードを 8 年ぶりに改訂した。

1 背景

当初、業務支援であった IT システムは、その技術自身の進展と時代の要請から、その利用が質、量共に拡大してきた。新たなビジネス価値を創出するための攻めの分野と、基幹業務を確実に遂行する守りの分野を区別し、それぞれに強化することが経営に直結する課題となった。

攻めの分野では、要求のすべてが開発初期に分からず、IT システムのサービス開始後に徐々に明らかになる要求への対応が常に求められる課題がある。また、攻めの分野を推進するためには守りの分野を着実に整備／更新し、適切に連携する必要がある。守りの分野では、長年保守開発を続けたシステムの再構築に「特有の難しさ」があり、下流工程のリスクであるにもかかわらず把握が難しい課題がある。実際、再構築プロジェクトの失敗事例報告は少なくない。

また、公開済の非機能要求に関しては、新たなセキュリティ脅威の台頭や、システム基盤技術の進展など、社会や技術の変化により、非機能要求グレード※1 初版公開時から非機能要求に変化が生じていることが課題である。

以上の上流工程に関する課題を解決するために、「要件定義」「システム再構築」「非機能要求」をテーマとした取り組みを行った。

2 要件定義

要件定義は、立場が異なる人とのコミュニケーションが最も多く、抜けや漏れが発生しやすい。図 1-1 に示す通り、5 つのリスクに対する対策が求められる。

「ユーザのための要件定義ガイド～要求を明確にするための勘どころ～」（以下、要件定義ガイド）を 2016 年度に出版※2 し、主にユーザ企業で IT システムの要件定義を実施する方を対象に、要件定義において発生する抜け、漏れなどの問題と、その解決方法をまとめた（図 1-2）。

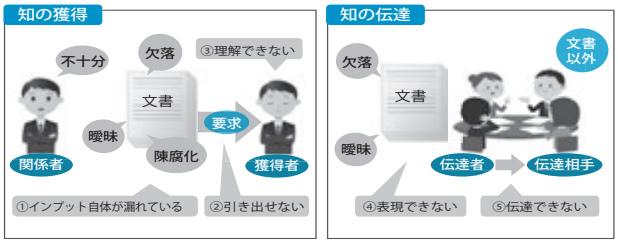


図 1-1　コミュニケーションギャップを生む 5 つのリスク

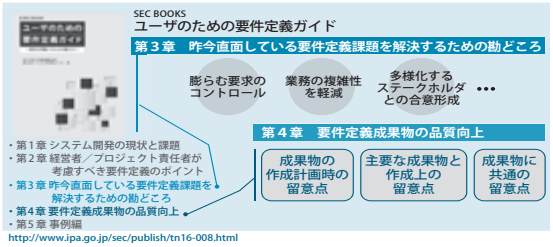


図 1-2　要件定義ガイドの構成

3 システム再構築

「システム再構築を成功に導くユーザガイド　第２版　～ユーザとベンダで共有する再構築のリスクと対策～」（以下、再構築ガイド）を 2017 年度に出版※3 し、ユーザ企業がシステム再構築の企画／計画工程で留意すべきポイントを、実践に即した形式で紹介した（図 1-3）。

とくに、品質保証における最も重要なテーマである「業務継続性の担保」については、読者の実践につながるような具体的な内容を示した（図 1-4）。

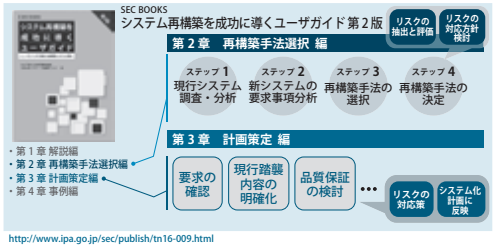


図 1-3　再構築ガイドの構成

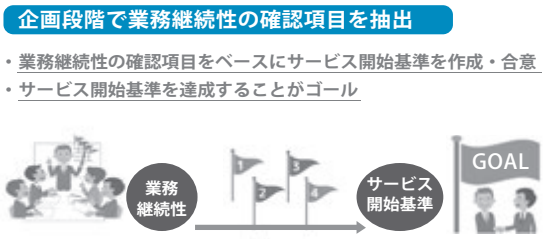


図 1-4　業務継続性を上流工程から明確化するイメージ

4 非機能要求

非機能要求グレードの初版を 2010 年度に公開したが、非機能要求の変化に伴う定義漏れを防止するため、非機能要求グレードのスコープは維持したまま、「非機能要求グレード 2018」を 2018 年 4 月に公開した。主な改訂対象は、「セキュリティ」と「仮想化」に関する要求である。

なお、主要な改訂は、図 1-5 に示す通り「非機能要求グレード本体」である。本体のうちの利用ガイド（解説編、利用編）、及び周辺資料の利用ガイド（活用編）や小冊子、各種研修教材は、メトリクスの総数など、改訂内容と整合させる必要がある部分のみを更新した。

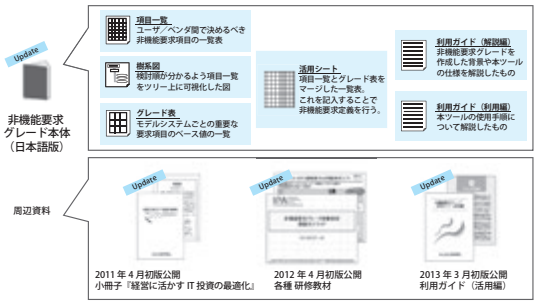


図 1-5　改訂した「非機能要求グレード」の成果物一覧

5 今後の予定

ガイドブックや非機能要求グレードの更なる普及のため、セミナーやイベントでの発信を行う。とくに、作成した小冊子を用いて、経営層など企業の上位層や、経験が浅い読者層にも内容をご理解いただきたい。

なお、要件定義ガイドは第２版や中小企業向けを作成する。第２版では、要件定義工程のより前半における課題解決の勘どころを充実させる。中小企業向けでは、比較的小さな規模に適用する際の勘どころをまとめる。

※1　URL: https://www.ipa.go.jp/sec/reports/20180425.html
※2　URL: https://www.ipa.go.jp/sec/publish/tn16-008.html
※3　URL: https://www.ipa.go.jp/sec/publish/tn16-009.html

システム構築能力の強化

システム理論に基づく安全性解析手法 STAMP/STPA の普及促進

SEC 調査役　石井 正悟　SEC 調査役　三原 幸博
SEC 調査役　十山 圭介　SEC 研究員　金子 朋子　SEC 研究員　向山 輝

STAMP/STPA は、現代の複雑システムに適した新しい安全性解析手法としてマサチューセッツ工科大学（MIT）が 2012 年に提唱した手法であり、欧米において活用が進展している。IPA/SEC では 2015 年度から、国内での STAMP/STPA の普及活動を行っており、この分野の有識者などが参加する「IoT システム安全性向上技術 WG」での検討を通し、これまでにガイドブックの発行、STAMP 支援ツールの開発・公開、ワークショップの開催を行ってきた。

1 はじめてのSTAMP/STPA（活用編）

IPA/SEC では、2015 年度に入門書として「はじめての STAMP/STPA」を発行し、2016 年度には、実際に適用する際のヒントやコツを解説した「実践編」を発行した。

2017 年度は、「理解する」「やってみる」と段階を踏んできたシリーズの第 3 弾として、「当たり前に実施する」ことを目指し、「はじめての STAMP/STPA（活用編）」を公開した（図 2-1）。

図 2-1　「はじめての STAMP/STPA」シリーズ

活用編では、鉄道、ロボット、自動車、電力網の分野の 4 つの具体例で、人と機械の協調による安全制御の分析や安全とセキュリティの統合分析などへの適用を解説した。いずれも、産業界において STAMP/STPA を役立てる際に参考となる先進的な事例である。

(1)　鉄道踏切の分析例

既存の鉄道踏切制御システムの課題を整理した上で、その解決方法として列車と踏切制御装置が情報交換を行いながら制御を行う「クローズドループ型」の安全性を STAMP を用いて評価した事例を解説した（図 2-2）。

(2)　二輪倒立ロボットの分析例

人と機械（ロボット）の協調によって転倒や衝突を避ける安全制御に対して、指示の競合などによる事故要因を STAMP を用いて分析した事例を解説した。

(3)　自動車の電動パーキングブレーキ（EPB）の分析例

一般社団法人 JASPAR（Japan Automotive Software Platform and Architecture）が仮想的な EPB を取り上げて STAMP/STPA 分析を実施した例を紹介した（図 2-3）。ISO 26262（自動車向け機能安全の国際規格）に規定される安全分析に STAMP/STPA を適用するための工夫や留意点が示されている。自動車分野以外の産業分野における安全規格や既存開発プロセスとの整合を考える上でも参考となる事例である。

(4)　電力網のセキュリティ分析への応用例

米国における広域送電網とローカル送電網の接続に関して、安全性とセキュリティを統合して分析するための STPA 拡張手法である "STPA-SafeSec" を適用した分析事例に関する文献を紹介した。

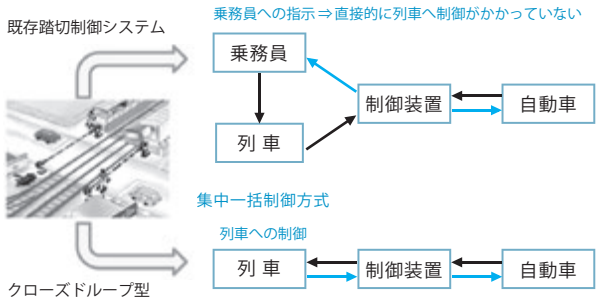


図 2-2　クローズドループ型踏切の STAMP モデル

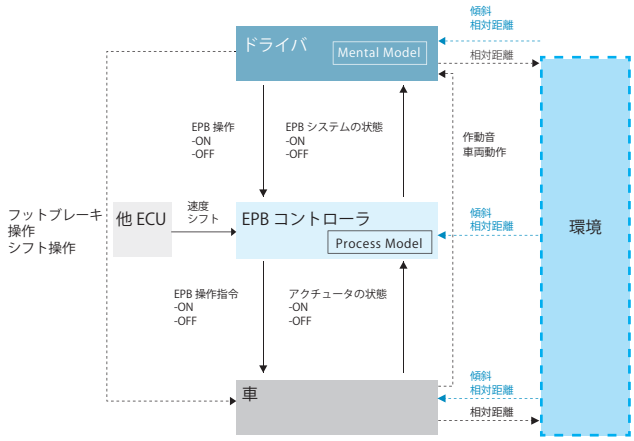


図 2-3　電動パーキングブレーキの STAMP モデル

2 STAMP Workbench

STAMP の導入を容易にするモデリングツール STAMP Workbench を 2018 年 3 月に無償で公開した。

STPA は分析において自由な発想を引き出すことを意図した手法であることから、分析の途中で新たな気づきを得て前の Step に立ち戻ることが少なくない。そうすることによって、分析の質が向上するので、新たな気づきを得ることは喜ばしいことである。しかし、喜ばしいとは言っても手戻りであることに違いはない。既に作成した図表を修正するのは、面倒な上に、修正ミスを犯しやすい作業である。更に、図表修正作業のために分析のための思考が中断されることが困る。こういうときに有効なのがモデリングツールである。しかし、STPA 分析という目的に合ったモデリングツールは海外に 2,3 あるものの、いずれも研究目的に開発されており、分析作業を支援するツールとしては物足りなかった。そこで、「産業界で STAMP を実適用する多くの人の役に立つ機能」を備えたツール STAMP Workbench を IPA/SEC が開発し、広く普及するよう、OSS として公開した。同ツールの開発思想は次の通り。

■手順を誘導する。しかし、使い方を限定しない

■専門用語、表記法を知らなくても解析できる

■可能な限り自動化し、分析者が思考に専念できる

STAMP　Workbench の画面例を図 2-4 に示す。

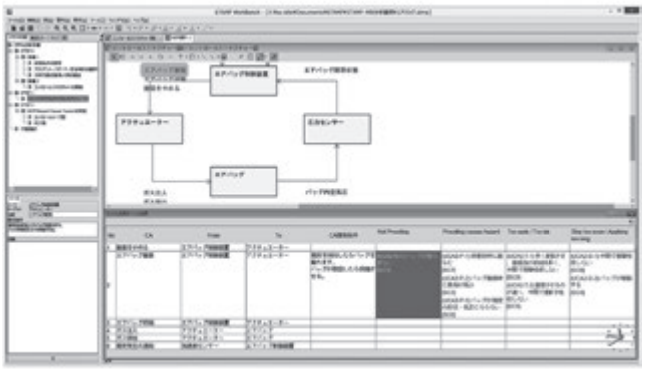


図 2-4　STAMP　Workbench の画面例

3 STAMPワークショップ in Japan

2016年に福岡市で開催した第1回に引き続き、第2回STAMPワークショップ in Japanを、2017年11月27日から3日間にわたって慶応義塾大学で開催した。

4カ国から延べ181名（第1回は117名）の参加者が集まり、初日に米国MITからの基調講演／チュートリアル、欧州STAMPワークショップ（ESW）からの招待講演が順に行われ、その後、一般講演として産業界から13件、学術界から11件、合計24件（第1回は16件）の発表が行われた（表2-1）。

表2-1　STAMPワークショップ（第2回）のスケジュール

開催日	プログラム
11/27（月）	チュートリアル・招待講演・一般講演（Overseas & Tools Session）：3件・STAMPツールデモ
11/28（火）	一般講演：16件
11/29（水）	一般講演：5件

併せてポスター展示を2件（第1回は0件）行い、2017年度に開発中であったSTAMP支援ツールSTAMP Workbenchを紹介し、期間中、デモ展示した。また、講演資料をIPA/SEC Webサイトに掲載した。（https://www.ipa.go.jp/sec/events/20171127.html）

一般講演の内容を分析した結果、以下の傾向が読み取れた。

①学術界からの発表は、手法解説に偏っているが、産業界からはどの種別にもほぼ同じ件数の発表がある

②試行事例の発表件数は産業界と学術界でほぼ同じである。産業界は、開発済みの制御システムを題材としているが、学術界は、制御システム以外にも目を向けている

③活用事例と手法改善は、産業界から多く発表されており、開発中又は今後のシステムに適用してみようという産業界の意欲が感じられる

④産業界から手法改善の発表が多かったのは、適用プロセスの標準化を目指して、手法の定型化を求めていることによると考えられる

4 今後の取り組み

今後は、これまでの活動の積み上げを生かし、STAMPを適用した効果をより多くの方に実感いただき、複雑化するシステムの安全解析の手法として広く有効活用されていくことを目指していく。

システム構築能力の強化

制御システム向けのセーフティ及びセキュリティ対策

SEC 調査役 石田 茂　SEC 調査役 久野 倫義
SEC 研究員 細目 紀子　SEC 専門委員 中谷 博司

クローズした環境で運用されてきた重要インフラを担う制御システムに対し、IoT化の進展や事業系システムとの連携など相互接続が進む中、セキュリティ対策が急務となっている。このような状況に対応するため、「制御システムセーフティ・セキュリティ要件検討ガイド」を発行し、セーフティとセキュリティのそれぞれの要件を連携させるための基本的な考え方と手順を示した。

1 背景

プラント、鉄道、電力など社会の重要インフラを担う制御システムは、稼働の連続性と共に、安全性や環境への影響などにも配慮を要することから、その多くは独自システムによるクローズした環境で運用されてきた。しかし近年では、IoT化の進展や制御系システムの事業系システムとの連携など、制御システムの稼働環境にも大きな変化が生じてきた。こうした相互接続が進む中、重要インフラを狙ったサイバー攻撃の増加は大きな問題となっており、セキュリティ対策が急務となっている。

一方、開発・運用に携わる現場には「セーフティ※1とセキュリティ双方に精通した技術者が極めて少ない」「安全性を確保しながらセキュリティ検討をどのように進めたら良いのか分からない」「情報セキュリティ技術者は、機密漏えいやシステムに対する改ざん・攻撃が、健康や安全性、環境に重大な影響を及ぼすなど、制御システム特有の被害イメージをつかみにくい」などの課題がある。

そこでIPAでは、2015年より関係企業や大学機関の方々から成る制御システムセーフティ・セキュリティ検討WGを設置し、これらの課題の解決に向けた検討を進めてきた。その成果を「制御システム セーフティ・セキュリティ要件検討ガイド」として取りまとめ、2018年3月に公開した。

2 本ガイドの目的と特徴

2.1 目的

システム障害発生時、人命や環境など社会活動に影響を及ぼすような制御システムを想定し、既存のセーフティシステム※2に対し、セーフティとセキュリティのそれぞれの要件を連携させるための基本的な考え方と手順を示すことを目的としている。

2.2 特徴

本ガイドの特徴を以下に示す。

【セーフティファースト】

セーフティシステムを含む制御システムによって稼働中の工場、プラントがあり、安全性の確保を実現済のシステムにセキュリティ対応を行うケースを想定。

【グローバル対応と国際規格】

事業のグローバル化の実情に鑑み、セーフティ・セキュリティの国際規格・標準を参照。

セーフティ：IEC 61508※3, Functional safety of electrical/electronic/programmable electronic safety-related systems

セキュリティ：IEC 62443※4, Industrial Automation and Control Systems Security

【二編構成による解説】

基本編では基本となる考え方と検討手順を示し、より理解を深めることができるよう、ケーススタディ編で抽象化されたシステムによる詳細を解説。また、脅威分析を実施する際の分析シートテンプレートも添付。

3 ガイドの構成

3.1 基本編

基本編ではセーフティとセキュリティの検討を進める際の考え方と検討プロセスについて図表を用いながら説明している。この際、記述内容に関連のある情報やトピックスをコラムとして要所に掲載した。

セーフティファーストの考え方に基づき、セキュリティ検討を行う際の検討プロセス概要は図 3-1 の通りである。

セーフティは構築済という前提なので、Step0 として過去に行われた安全設計経緯の確認から開始し、事業者が実施するセキュリティ検討とその結果を受けてインテグレータが実施するセキュリティ検討という手順として示した。

※1 「セーフティ」：安全
※2 「セーフティシステム」：国際機能安全規格などに適合した安全関連システム
※3 IEC 61508: IEC（国際電気標準会議）が制定した基本安全規格。プロセス産業における電気・電子・プログラマブル電子（E/E/PE）機能安全に関する国際規格
※4 IEC 62443: 制御システムセキュリティの事業者、インテグレータ、装置ベンダを対象とした汎用的な国際標準規格。

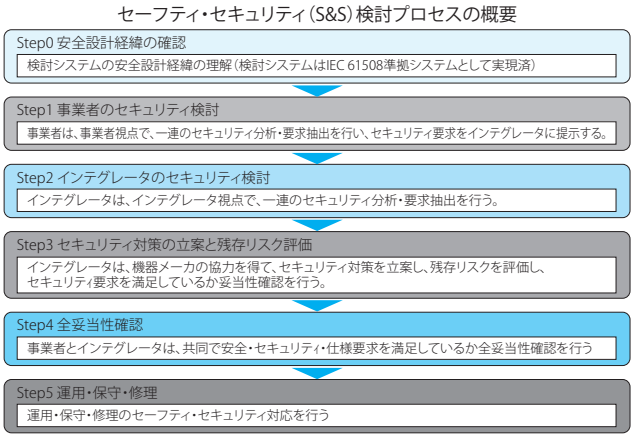


図 3-1　検討プロセス

3.2 ケーススタディ編

ケーススタディ編では、基本編の内容を具体的に示し、理解を促進できるよう、現実のシステムを抽象化した「検討システム」を用いて解説した。この「検討システム」は産業用ロボットを含む FA（ファクトリーオートメーション）システムで、実在するものではなく、あくまでも解説用のものである。

ケーススタディ編ではこの検討システムに対して、基本編で示したプロセス手順に従って、検討が進む一連のストーリーとして記述した。図 3-2 に検討例を示す。

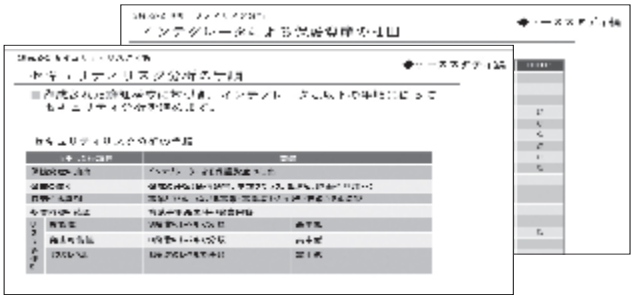


図 3-2　ケーススタディの検討例

3.3 脅威分析シート

本ガイドでは、手順に従って系統的に脅威分析を行うために、図 3-3 に示す「脅威分析シート」を用いている。

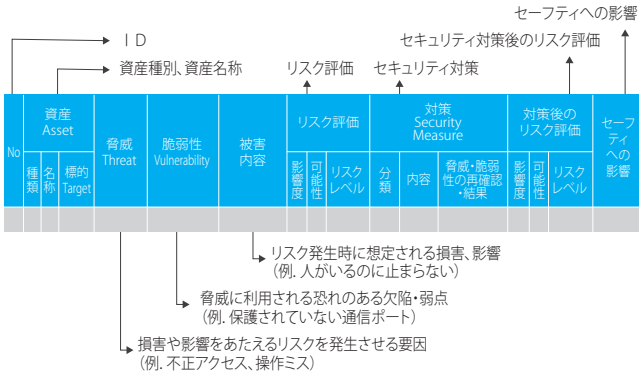


図 3-3　脅威分析シートの構成

システム構築能力の強化

コーディング作法ガイド（ESCR※1）の整備

SEC 調査役 十山 圭介　SEC 調査役 三原 幸博　SEC 調査役 久野 倫義

ESCR は 2006 年の C 言語版 Ver. 1.0 発行以来、言語規格の更新などに追随して改訂を行っている。2015 ～ 2016 年度には、C++ 言語版の改訂（Ver. 2.0）を実施し、2017 年度には、セキュアコーディングに向けた C 言語版の改訂に取り組み、C 言語版 Ver. 3.0 を公開し、2018 年 6 月 20 日に書籍化し発行した。

1 コーディング作法ガイドの改訂

SEC ではコーディング作法ガイド改訂 WG において、組込みソフトウェアのソースコード品質の向上を目的に、ESCR としてコーディングの際に注意すべき事柄やノウハウを取りまとめ、公開している。 ESCR では、コーディングにおける基本的な考え方（作法）と対象言語に合わせて作法を具体化した個々のルールとをソフトウェア品質特性の観点で整理している。組織でコーディング規約を決める際やコーディング時の参考、また個人のプログラミング学習のために、書籍や PDF 版などこれまで 3 万部を超えて多くの方々に ESCR を利用いただいている。2017 年度は以下の改訂を行い、その結果を公開した。

● セキュアコーディングに向けた ESCR の改訂

この改訂では、CERT C※2 ルールの一部や IPA セキュリセンター（ISEC）からの提案を新規ルールの追加や解説の拡充といった形で ESCR の中に取り込む作業を進めた。CERT C ルールと ISEC 提案ルール 6 個をリストアップし、それらに対応する ESCR ルールがあるか、ない場合に新規作成するか、解説を追加するかといった点を検討して ESCR [C 言語版] の改訂版（Ver. 3.0）を作成した。

● ESCR ルールと CERT ルール対応表の改訂

「現状の ESCR のルールにはセキュリティの観点から重要なものが含まれており、それらと CERT C ルールとの対応付けを行ってコーディング規約の作成段階からセキュリティを念頭に置く」（図 4-1）ことが重要であるとの認識から、暫定版ではあるが 2016 年 6 月から ESCR C ルールと CERT C ルールの対応表を公開している。

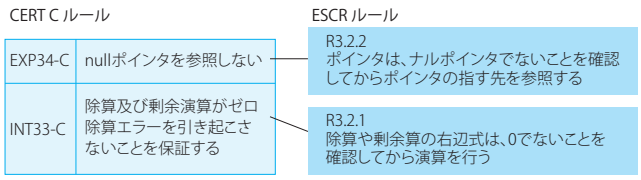


図 4-1　ESCR と CERT の典型的なルール対応関係

2017 年度には、ルールや解説の ESCR 本編への追加に加え、この対応表を改訂・拡張して C++ 言語のルール対応も含めたものとして更新した。表 4-1 に対応表の一部を示す。

表 4-1　ESCR と CERT などのルール対応（一部）

ESCR ルール	MISRA ルールとの関係			CERT C	CERT C++	CWE
	C:2004	C:2012	C++:2008			
［信頼性 1］R1 領域は初期化し、大きさに気を付けて使用する。						
R1.1.1	9.1	R9.1	8-5-1	EXP33-C	EXP53-CPP	CWE-119 CWE-456 CWE665
R1.1.2						CWE-456
R1.2.1				ARR02-C STR11-C STR31-C		ARR02-C STR11-C STR31-C
R1.2.2	9.3	R8.12	8-5-3	INT09-C		CWE-665
R1.3.1	17.1	R18.1	5-0-15 5-0-16	ARR30-C ARR37-C ARR39-C	ARR30-C ARR37-C ARR39-C	CWE-119 CWE-122 CWE-129 CWE-468 CWE469 CWE-788
	17.4	R18.4				

2 コーディング作法ガイドに関する海外連携

MISRA C と MISRA C++ は MISRA※3 が策定しているコーディングガイドラインであり、安全で信頼性あるソフトウェアの開発のため、自動車業界を中心に広範に運用され標準技法としての地位を築いている。 SEC では、ESCR と MISRA C とで相互引用や改訂時のレビューを行うなど、MISRA と連携して活動している。

MISRA からは、MISRA C++ の改訂スケジュールとその適用プロジェクトに関する情報を得ている。また、SEC の WG では 2016 年度の ESCR［C++ 言語版］改訂に対する MISRA WG のメンバからのコメントに対応し、C 言語版にも適用される項目については、その Ver. 3.0 に反映した。

※1　ESCR：Embedded System development Coding Reference
※2　CERT C：C 言語を使ってセキュアコーディングを行うためのルールを定めたもの（カーネギーメロン大学ソフトウェア工学研究所による）
※3　MISRA：The Motor Industry Software Reliability Association

ソフトウェア工学分野の先導的研究支援事業について

SEC 調査役　小沢 理康

IPA/SEC では我が国におけるソフトウェア工学・システム工学分野の研究の促進及びその成果の産業界への展開を図るため、「ソフトウェア工学分野の先導的研究支援事業」を 2012 年度から 2016 年度まで実施してきた。この 5 年間に 20 件の研究を支援しており、支援終了後も各大学などでは研究が継続され、また、研究成果が企業などで試用されている。本稿では事業終了後の研究の継続状況や研究成果の利活用状況についての調査結果を報告する。

1　研究支援事業の概要

IPA はソフトウェア工学やシステム工学にかかわる研究、また、ソフトウェアの経済的効果に関する研究についての一層の振興をねらいとして本研究支援事業を実施してきた。本研究支援事業による 17 大学 20 件の研究成果（表 1 にテーマ一覧を示す）の概要は以下の URL にて公開している。

https://www.ipa.go.jp/sec/rise/index.html

2　研究支援後の成果発表と継続研究

研究成果は論文や講演等を通じて外部へ公表され、周知される。研究は支援終了後も継続され、更に発展することが期待される。IPA は研究成果の普及や研究の継続・発展状況を把握するため、各大学への調査を実施している。

20 件の研究の論文が論文誌や書籍などへ掲載された件数の累計は 104 件であった。同じく学会、セミナー、展示会などにて発表や講演が行われた件数の累計は 134 件となっている。研究成果が広く公表されていることが分かる。

また、他の大学や企業などと共同研究に結び付いた件数は 24 件となっている。複数の大学と共同研究を進めるだけでなく、企業からの支援を得て研究を進めている事例も見られる。

3　研究成果の企業での利活用

本研究支援事業では研究成果が産業界へ移転し、利活用されることを事業の目的の一つとしていることから、研究成果に対する企業などからの問い合わせ状況や、企業などでの利活用状況も調査している。

研究に対する企業からの問い合わせは 56 件以上であった。自社事業に関連する部分についての意見交換や相談、研究成果を社内に導入するための問い合わせ、研究成果であるツールのインストール方法など問い合わせ内容は様々である。

研究成果が企業などで利活用されているとの回答は 24 件であった。研究成果であるツールが適用できるかを企業が検証したり、企業が有する実データを適用して分析や評価を実施したりするなどの例があり、企業での試用が進められている様子が窺える。IPA は引き続き研究成果の産業界への移転状況などについてフォローしていく所存である。

表 1　研究テーマ一覧

＜ソフトウェア高信頼化＞
- 要件定義プロセスと保守プロセスにおけるモデル検査技術の開発現場への適用に関する研究（芝浦工業大学）
- モデルを含む設計成果物の集積とその活用方法に関する研究（九州大学）
- 実用性が高い形式工学手法と支援ツールの研究開発（法政大学）
- 次世代ソフトウェア信頼性評価技術の開発とその実装（広島大学）
- 抽象化に基づいた UML 設計の検証支援ツールの開発（岡山県立大学）
- 形式仕様とテスト生成の部分的・段階的な活用（情報・システム研究機構）
- 保守プロセスにおけるモデル検査技術の開発現場への適用に関する研究（芝浦工業大学）
- データマイニング手法を応用した定性的信頼性／安全性解析支援ツールの開発（広島大学）
- 要求定義の高品質化のための要求仕様の整合性の検証知識の形式知化と一貫性検証支援ツールの開発（工学院大学）

＜ソフトウェア品質評価＞
- ソフトウェア品質の第三者評価のための基盤技術ーソフトウェアプロジェクトトモグラフィの開発ー（奈良先端科学技術大学院大学）
- コードクローン分析に基づくソフトウェア開発・保守支援に関する研究（大阪大学）
- ソフトウェア品質の第三者評価のための基盤技術ーソフトウェアプロジェクトトモグラフィ技術の高度化ー（奈良先端科学技術大学院大学）
- 測定評価と分析を通じたソフトウェア製品品質の実態定量化および総合的品質評価枠組みの確立（早稲田大学）

＜保証ケース＞
- オープンシステム・ディペンダビリティのための形式アシュランスケース・フレームワーク（神奈川大学）
- 保証ケース作成支援方式の研究（名古屋大学）
- D-Case に基づく議論構造可視化支援ツールの開発と、スマートコミュニティにおける合意形成の実証（電気通信大学）

＜プロジェクト管理＞
- IPA EPM-X の機能拡張によるプロアクティブ型プロジェクトモニタリング環境の構築（和歌山大学）

＜システム工学＞
- システムモデルと繰り返し型モデル検査による次世代自動運転車を取り巻く System of Systems のアーキテクチャ設計（慶應義塾大学）

＜その他＞
- 携帯端末用アプリケーションソフトウェアが地方経済に与える効果の実証実験評価に関する研究（福井大学）
- 日本のソフトウェア技術者の生産性及び処遇の向上効果研究：アジア、欧米諸国との国際比較分析のフレームワークを用いて（同志社大学）

プロモーション活動

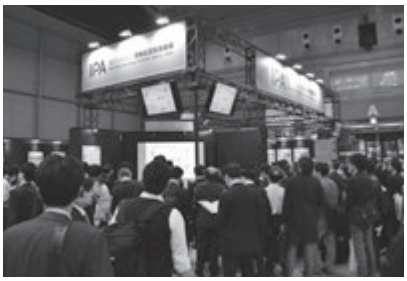
SEC プロモーショングループ　佐藤 康彦

IPA ではソフトウェアの信頼性、安全性に関する事業成果を、技術者層及び経営層に向けて、展示会、イベントへの出展、セミナーの開催などにより、普及・啓発活動を行っている。

1 展示会への出展

事業成果普及の一環として、一般社団法人組込みシステム技術協会の主催により 7 月 12 日、13 日に大阪で開催された組込み総合技術展関西（ETWest2017）、及び 11 月 15 日～ 17 日に横浜で開催された組込み総合技術展（ET/IoT2017）に出展し、パネル展示、資料配布及びセミナーを実施した。展示では近年普及が進む IoT 機器・システムの安全性確保を推進する「つながる世界の開発指針」や「組込みソフトウェア開発データ白書」「STAMP/STPA」に関する関心が高かった。

両展示会での IPA のセミナーは、産学界からの著名な講師の講演が聴講できると例年好評であり、2017 年度も多くの来場者に聴講いただいた。ET/IoT2017 では展示会開催日に合わせて「組込みソフトウェア開発データ白書 2017」を発行し、同白書の活用など定量的管理を推進するセミナーを実施した。また、IT 系の若手ベンチャー企業代表による IoT・AI をテーマにしたトークセッションも注目を集めた。そのほか、IPA の成果物を解説するブースプレゼン、IT の先端技術を有識者がコンパクトに紹介する先端技術入門ゼミも、連日多くの聴講者で賑わった。



ET/IoT 2017　IPA ブース

2 イベントの共催

ソフトウェア関連学会や団体が実施するイベントの共催、出展を行っている。2017 年 11 月には STAMP に関する国際会議、「第 2 回 STAMP Workshop」を開催した。STAMP Workshop は米国、欧州で毎年開催されているが、日本では 2016 年度の福岡での開催に続き、慶應義塾大学（東京）で第 2 回目を開催した。Workshop では欧米での適用事例の紹介に加え、IPA からは 2018 年 3 月に公開となった STAMP 支援ツールである STAMP Workbench の紹介も行った。また、国立研究開発法人宇宙航空研究開発機構（JAXA）と共催しているクリティカルソフトウェアワークショップ（WOCS[2]：Workshop on Critical Software System）も 2017 年度は同会場で続けて開催し、両イベントのシナジー効果を狙った。

同じく 11 月には特定非営利活動法人横断型基幹科学技術研究団体連合（横幹連合）との共催で、複雑化する現代社会で重要さを増すシステム化への理解と推進を目指して「システム・イノベーション」シンポジウムを開催し、産学官の有識者によるディスカッションを行った。

「システム・イノベーション」シンポジウム

3 セミナーの開催

ソフトウェアの信頼性、安全性に関する技術の普及・啓発のため、首都圏のみならず、地域団体や民間団体と連携し、事業成果をテーマに職員や外部有識者を講師としたセミナーを実施している。

2017 年度は東京で 36 回、地域で 7 回、計 43 回のセミナーを開催し、2,335 名の方に参加いただいた。セミナーのテーマとしては、ソフトウェア開発で大切とされる「上流工程」強化の取り組みに関するものを 10 回開催（参加者 637 名）、「IoT 推進」に関するものを 6 回開催した（参加者 469 名）。また、「STAMP」に関するセミナーも 3 回開催し、251 名の参加者を得た。

また、日本のソフトウェア産業は首都圏に集中しているため、IPA のセミナーも首都圏に集中していたが、地域でも首都圏同様のセミナーの開催を要望される声もあり、近年地域でのセミナーの共催あるいは講師の派遣にも力を入れている。

4 成果物のダウンロード、購入

当機構の報告書などの事業成果は IPA Web サイトからダウンロードができる。また発行書籍に関しては直販、amazon による通販での購入が可能となっている。書店によっては取り寄せも行っているので詳細はお近くの書店にお問い合わせいただきたい。

2018 年度は IPA 第四期中期計画がスタートし、当センターも昨今の急速な変化を続ける IT 社会の動向に対応すべく、社会基盤センターとして活動を開始した。今後は更に社会動向、業界ニーズに呼応した活動の普及推進に向けた取り組みを進めていきたい。

SAPの成功：ドイツの制度環境からの一考察

同志社大学客員教授、フランス国立労働経済社会研究所（LEST）、ドイツ日本研究所（DIJ）客員研究員 山内 麻理

アメリカ企業の独壇場である汎用ソフトウェア市場でなぜドイツのSAPが統合基幹業務システム（ERP）における世界首位の地位を築くことができたのだろうか。本稿においてはERPの商品特性とドイツの教育訓練制度や企業間コーディネーションなど制度的特徴との関連からその答えを導く。

1 はじめに

日本のソフトウェア産業からなぜ世界標準となる商品が開発されないのか。この疑問についてはこれまで幾度となく議論されてきた。シリコンバレーに代表にされるアメリカ、高度な開発も人海戦術も可能なインドや中国、オープンな開発環境を重視する北欧諸国など、各国が正にそれぞれの制度や文化、環境を活用したイノベーションや商品戦略を展開している [Lundvall 1992; Nelson 1993]。他方、大規模なソフトウェアパッケージは、マイクロソフトやオラクルなどアメリカ企業の独壇場だが、そんな中で、ドイツSAPのERP（統合基幹業務システム）市場における存在感は際立っている。現在世界の大企業向けERP市場におけるSAPのシェアは23％程度であり、2位のオラクルやインフォア（それぞれ16％程度）を引き離している [Statista 2015a]。

90年代以降主流となった資本主義経済の類型化に従えば、ドイツは日本と同様に調整型市場経済として分類され、労使や政府、金融機関など主要ステークホルダによる協調や調整が重視される国とされ、（協調や調整より）市場における標準設定を重視し、ITやバイオテクノロジーなど急進的イノベーションを得意としてきたアングロサクソン諸国と対比される [Hall and Soskice 2001; ドーア 2001]。そのような国であるドイツにおいて、なぜ世界的なソフトウェアパッケージが開発されたのだろうか。本稿では、SAPの世界的な躍進について、ドイツの制度的枠組みに照らして議論し、なぜ、同じく調整型市場経済として分類される日本において同様のイノベーションが起きなかったかについて再考する。

2 SAPの歴史

SAPはIBMを退職した5人のプログラマーが1972年にドイツ南西部のバーデンヴュルテンベルグ州に設立した企業である。現在は、130以上の国々に84,100人の従業員を雇用、アメリカ人CEOが指揮する多国籍企業である。売り上げの約85％が海外であり、「フォーブスグローバル2000」に登場する2,000社のうち、実に87％（1,737社）がSAPの顧客とされ、世界の商取引の74％がSAPのアプリケーションによって処理されている [日経BP 2014] と言う。

多くのERPベンダは、もともとある機能に特化したソフトウェアの開発から始める傾向があるが、SAPの場合、財務と購買を初期のシステムとして開発、次第にほかのモジュールを追加し統合システムを完成している。現在では、会計が62カ国、人事が94カ国の国別バージョンを提供、世界のGDPにおいて前者が94％、後者が96％に相当する [日経BP 2014]。最近ではクラウド、ミドルウェア、ビッグデータなど新分野にも力を入れ、鉄道網の整備やスポーツ競技の成績向上のためのデータ解析などでも実績を上げている。また、2009年までCEOを務めたカガーマンはドイツのインダストリ4.0の提唱者の一人としても知られている。

設立当時、コンピューターは非常に高価であったため、SAPは1980年までコンピューターを購入できず、開発はもっぱら顧客企業のサイトで行われた。SAPは、当初から標準化されたコードを製作しほかの顧客にも再利用可能なソフトウェアの製作を目指した。しかし、ドイツにおいても当時のソフトウェア産業では、「顧客のために特別に仕立てたソフトを提供する」ことがベンダ企業に期待された規

範であったため、汎用性の高いものを作るという目標は顧客企業には伏せておかれることもあった。しかし、SAP は各社との契約の中で構築したコードを活用し標準化された中核コードを拡張していった [Lehrer 2006]。

SAP のシステムはもともと IBM のメインフレーム用に設計されていたが、80 年代にはシーメンスをはじめとするほかのハードウェア上でもインストール可能となった。1981 年に完成した最初の多機能装備型システム R/2 は、こうしてメインフレーム上で稼働する大型業務用ソフトウェアの市場を静かに独占していった [Lehrer 2006: 195]。次世代パッケージの R/3 では、より拡張性の高い、Unix を中心としたクライアントサーバー型にいち早く変更、企業の IT インフラ刷新と共に、顧客基盤を拡大していった。R/3 はもともと中堅企業向けに設計されていたが、海外オペレーションの統合を考えていた大企業から熱心な引き合いを受けた [Lehrer 2006]。

ドイツにおいても、SAP の ERP が開発される以前は、特定の用途の IT インフラは外注するものの、日々のオペレーションに関するソフトウェアは各企業が独自に製作することが主流だった。1990 年代に R/3 が普及するにつれて、そのコスト競争力や多機能性から自前のソフトウェア開発をあらためアウトソース化する動きが急速に進展、それと並行して、ドイツの大規模・中規模 IT 企業の多くが市場から退出した。例えば、Siemens Nixdorf は、まずハードウェアから、次いで、ソフトウェア市場から撤退している [Lehrer 2006:197]。こうやって SAP は大型基幹業務システムの国内市場を静かに征服していった。ERP のインストールには通常幾つものパラメータ設定が必要となり、かつ、企業内の業務プロセスを大幅に体系化する必要があるため、多くの顧客企業は外部の IT コンサルタントを活用した。そのため、90 年代には、R/3 のコンサルタントは、IT コンサルタントの中で最も高報酬と言われる存在となった。

SAP は、80 年代からソフトウェアのインストールに関する業務を IT コンサルタントや IT ベンダに委任し、自らはパッケージ開発に注力したため、Big 6 会計事務所（当時）などその後の SAP の海外展開に重要な役割を果たす IT サービス企業との関係を早期に構築している [Lehrer 2006]。同時に、SAP の海外展開から恩恵を受けた IT ベンダも多く存在する。SAP 商品を専門に取り扱う CSC Ploenzke and Plaut は SAP の海外展開を活用して国際市場でのプレゼンスを高めることに成功した。Debis（Daimler-Chrysler の IT サービス部門）や Siemens-Nixdorf の SAP サービス部門も同じく急速に海外展開を果たした。Ixos Software AG や IDS Prof。 Scheer と言った関連機器を取り扱う企業も 1990 年代に急速に成長している [Lehrer 2006:197-198、企業の名称は当時のもの]。

3 政治経済の仕組みと産業政策

ドイツ型市場経済は、労使や産官学による緊密なコーディネーションを基盤に成り立ってきた。その仕組みは、日本型の企業グループを中心とするコーディネーションと異なり、産業別の経営者連合や労働組合を主軸とするより広範囲な単位で構成される。また、各組織の役割、権利、義務は明文化された法令に依ることが多く、慣習に基づくことが多い日本型の労使関係やコーディネーションと対比される [ドーア 2001]。ドイツの伝統的産業別賃金交渉やデュアルシステムに代表される職業訓練制度（後述）は、そのような強力な労使関係や産官の連携をもとに行われてきた。

例えば、ドイツの賃金や労働条件の決定においては、産業別労働組合と産業別雇用者連合による集団交渉が重要な役割を果たしてきた。毎年の賃金交渉は産業ごと、地域ごとに行われ、金属加工業であれば、バーデンヴュルテンベルグ州がほかのすべての地域の賃上げ率の決定的な指標となる [ドーア 2001: 271]。代表的な団体交渉に基づく賃金協定が、同じ雇用者連合に加盟するすべての企業に適用されてきたため、ドイツにおいてはこれまで最低賃金が存在しなかったほどである（2015 年 1 月に導入、時給 8.5 ユーロ）。従って、日本に見られるような企業規模間の賃金格差は限定的であり、賃金格差を利用した下請け構造はそもそも成立しにくかった。

ドイツではまた公式な職業訓練が発達し職業資格が重視されている。最近でも勤労者の 8 割程度が保有する資格と何らかの関連がある業務に就いている [Bosch 2010]。従って、同一労働同一賃金の原則が定着しながら、保有する職業資格などによる賃金の差別化が行われてきた。そして、賃金協約や職業資格が職種や業種を基準としている点は、職種・業種別に構成される SAP、ERP の構造と符合する。

職業訓練においては、デュアルシステムと言われる、企業内 OJT と公立の職業学校における座学を組み合わせた若者向け訓練がとくに発達している。デュアルシステムにおけるカリキュラム策定や運用においては、やはり産業別の雇用者連合や各地の商工会議所などの企業側代表、産業別労働組合など労働者側の代表、更に、職業学校や連邦・州政府などのソーシャルパートナーが深く関与している。従って、ドイツの同業者たちは日本よりもはるかに頻繁に会う

機会がある [ドーア 2001]。そのような背景から、インダストリ 4.0 のような国を挙げた産業政策や自動車業界における部品標準化のイニシアティブなどが高い実効性を持つことも理解できる。

企業のイノベーションや新製品開発を支援する機関としては、フラウンホーファー研究所やマックスプランク研究所などが知られているが、中でも、フラウンホーファー研究所は、欧州最大の応用研究や結果重視の研究体制で圧倒的な存在である。同研究所の所長の多くは、企業への勤務経験を有し、教授の国家資格も保有している。そのため、大学と企業間の知識交流の懸け橋となり [中村 2015]、企業は自社だけで解決できない課題を乗り越えるために研究所の扉をたたく。フラウンホーファー研究所のミッションの一つは、R&D 機能を持たない中小企業に自身が有するイノベーションノウハウを提供、産業界のために働き、最終的に売れる製品化に貢献することである [岩本 2015:16]。従って、各研究員は、論文や特許の数ではなく、企業からの受託研究基金をいかに集めたかということで評価されるため、積極的に企業のニーズをくみ取るように行動するインセンティブを持つ。年間予算の 3 分の 1 は企業からの受託研究費であり、政府からの資金援助も企業から獲得した受託研究額に応じて支払われる [中村 2015]。

その背景として、ドイツではミッテルシュタンド（Mittelstand）と呼ばれる中堅・中小の同族企業が機械産業を中心に経済発展に大きな役割を果たしている。例えば、ドイツ機械工業連盟（VDMA）は約 3,000 社に及ぶ機械・プラントメーカが所属する欧州最大の生産財の工業会連合体であり、インダストリ 4.0 においても主導的役割を演じている。日本型市場経済では、中小企業は大企業の下請けとなりやすく表舞台に立つことは多くないが、ドイツにおけるミッテルシュタンドは、ドイツ経済の「隠れたチャンピオン」として地域経済の活性化に貢献すると共に、堅調な輸出にも大きく寄与している。例えば、日本の中小企業（従業員数 300 人以下）のうち直接輸出を行っている企業は 2.8％に過ぎないが、従業員 250 人未満のドイツ企業のうち 20％が直接輸出を行っている [METI 2012]。フラウンホーファーなどの研究所がそうした中堅・中小企業のイノベーションを積極的に支援している。

ドイツはまた貿易立国であり、同規模の国の中でも輸出や対外直接投資の GDP に対する比率が相対的に高く、GDP 当たりの輸出額は日本の約 4 倍である [岩本 2015]。日本同様、少子高齢化による国内市場の低迷、東欧の民主化による近隣諸国への投資機会の増大、EU 統一市場の拡大などもあり、とくに 90 年代後半は対外直接投資が増加、日本の投資額が GDP 比 1％以下だった当時、ドイツでは少ない年で 2％台、多い年は 5％を超える対外直接投資を行っていた。

コーポレートガバナンスは、戦後銀行を中心とする内部型モニタリングが主体であったが、社会民主党のシュレーダー政権下で企業税制改革の一環として銀行や企業の保有する株式に対するキャピタルゲイン課税が廃止されたことなどから、金融機関を中心とする株式の持ち合いは一気に解消されている※1。その結果、企業はより株価を意識した経営を行い敵対的買収など外部からの脅威に自ら備える必要も生じている [Streeck 2009]。また、海外展開を資金的に賄うため直接市場から積極的に資金調達したこともあり、株式市場における海外投資家のプレゼンスは高く、現在ではドイツ株価指数 DAX の外国人持株比率は 60％を超えている。

他方、ドイツはフランスやイタリアと同様に同族ファミリーによる企業支配が強い国でもある。2006 年時点で売上高上位 1,000 社のうち 34％以上の企業が 25％以上の議決権を同族によって保有されている [吉森 2015]。すなわち、ミッテルシュタンドのような中堅・中小企業だけでなく、フォルクスワーゲン、BMW、ボッシュ、ヘンケルなど名だたる大企業がこうした同族企業に該当する。そのため、上場株式時価総額の対 GDP 比は、アメリカはもとより日本をはるかに下回る。同族企業は長期志向でありながら迅速な意思決定が可能であり革新能力が高い。メルケル首相は同族企業を「ドイツ経済の牽引車」と呼び称賛している [吉森 2015:3]。このような企業群を支えるため、ドイツ企業の形態は多様で合計 20 ほどの会社形態や公益財団が円滑な事業承継を可能としている [吉森 2015]。また、企業の法的整理や M&A などコーポレートリストラクチャリングを促進する法制度や税制は日本よりはるかに整備されている [木下 2013]。

同族企業の存在に加え、シュレーダー政権下の税制改革は、ドイツ経済の立て直しのため、アメリカ型の企業経営や資本市場の活性化を意識して行われたこともあり、ドイツのコーポレートガバナンスは、米英ほどではないものの、日本よりはるかに株主重視の方向へ変化している [ドーア 2001; Streeck 2009]。従って、企業価値を重視する経営が

※1　その後2008年の税制改革で非課税措置は廃止され、現在では法人所得として課税されている [野村総合研究所 2014:11]。

より浸透している可能性が高く、ユーザ企業においては、洗練された統合システムを活用した効率経営が重視されたこと、ベンダ企業においては不採算事業からの撤退が促進されたことなども、SAP 商品の急速な普及に寄与したと考えられる。

ドイツ国内の ERP 市場のシェアを見ると、SAP が 55％程度、2 位以下はマイクロソフト（7.7％）、SAGE（6.4％）、インフォア（6.1％）、オラクル（1.9％）と外資が続く [Statista 2015b]。CRM については、SAP が最大のシェアを持つものの、30% 程度と業務系のシェアに比べると低水準である。CRM についても、SAP に次いで、オラクル、セールスフォースドットコムなど米系企業が後を追う形となっている。すなわち、大規模統合基幹業務システムの国内ベンダは現在では SAP だけであり、企業グループや分野ごとにベンダが乱立した日本とは極めて異なる産業構造となっている。

4 ドイツ型市場経済の特徴とSAPの躍進

ここで、ドイツの政治経済の仕組みと SAP の成功との関係について議論したい。最近の比較制度分析やイノベーション論においては、各国の商品戦略やイノベーション創出は、教育訓練制度、コーポレートガバナンス、企業間コーディネーションなどその国の制度的枠組みに大きく依存する [Lundvall 1992；Nelson 1993; Hall and Soskice 2001]。ERP のような統合基幹業務システムがドイツで最も発達した背景として、多くの制度的特徴が関連している。

まず、ドイツ型市場経済においては、様々な基準や手続きが明文化されていることが多く、各社の制度や業務フローが標準化されやすいという特徴がある。SAP 商品の急速な普及はその点から大いに裨益（ひえき）したとされる [Lehrer 2006]。例えば、（とくに伝統的産業における）ドイツ被用者の賃金や給与は、基本給についてはそのほとんどが、業績給についてもかなりの部分が企業を超えた産業別の労使交渉で決められており、産業や業種ごとに参考となる職務等級や報酬テーブルも存在する [大塚 2010: 338, Lehrer 2006]。

また、産業ごとの企業間コーディネーションは賃金だけでなく研修制度やワークシステムなど広範囲にわたり、国際比較調査においても英米企業と比べてドイツ企業の業務プロセスがより企業間で類似していることが明らかになっている [Lane and Bachmann 1996; Lehrer 2006]。このような標準化や企業間コーディネーションは、戦後のドイツが漸進的変化を特徴とする産業に特化してきたため達成されやすかったという産業構造上の特徴とも関連している .

教育訓練制度については、上述の通り、デュアルシステムを中心とする職業訓練制度が整備され、従業員の技能や知識が平均的に高く共通性が高いことが挙げられる。デュアルシステムにおける初期職業教育訓練は、BIBB（Bundesinstitut für Berufsbildung）と呼ばれる連邦職業教育訓練機構、商工会議所、労働組合、職業学校などのソーシャルパートナーが共同で策定する訓練規定に準拠して行われる。そのため、訓練生の習得する技能は企業間で一定の共通性が確保される。日独の技能の違いに注目した Streeck [1996] に言わせれば、ドイツの技能は職種による共通性が高く移転可能だが、日本の技能はより企業特殊的である。

また、職業訓練は顧客企業もベンダ企業も同一の職種であれば同一の訓練規定に沿って行われるため、顧客側の知識が低いという状況も避けられる。ドイツとイギリスのソフトウェア開発プロジェクトを比較した実証研究 [Grimshaw and Miozzo 2006] を見ると、ドイツのほうが顧客企業担当者の IT 知識が押しなべて高く、そのため、ベンダとユーザ（顧客）のコーディネーションが円滑であった。また、長期的で円滑な労使関係により、アウトソーシングに伴う IT スタッフの移籍も計画的に行われる。同調査によると、ドイツでは、IT スタッフの移籍について、労使協議会※2 を含めた包括的な交渉が、実際のアウトソーシングより 6 カ月程度も先駆けて丁寧に行われるため、従業員側の安心感が高いとされる。また、そのプロセスは企業間で共通性が高く、個別企業の方針や資本国籍による労働慣行の差異もより限定的である。比較して、イギリスでは、顧客企業側担当者の IT 知識のバラツキが大きく、IT スタッフの異動も唐突に行われる傾向があるため、スタッフの抵抗や不満がより頻繁に報告されている。

このような背景から、ドイツにおいては共通のソフトウェアを活用する素地が他国よりはるかに整っていると想定される。SAP 創設者の一人であるプラットナーによれば、複数機能を統合したビジネスソフトウェアを 90 年代以前に開発できたのは SAP をはじめとするドイツのベンダだけであった [Lehrer 2006: 202]。

最近の比較制度分析によれば、技術進化の著しい IT 分野は、一般的に自由な労働市場やリスクマネーの集まりやすいアングロサクソン諸国でより発展しやすいとされる。他

※2　ドイツの事業所で労使が広範な労働条件について協議する場。

方、IT やバイオテクノロジーなど新分野の技術やリスク特性をより詳細に分析した Casper and Whitley [2004] の調査によれば、ERP のような業務系ソフトウェアは、IT 分野の商品の中でも、スタンダードソフトウェア（アプリケーションソフトウェアなど）やミドルウェアなどのように、不確実性が高く技術の陳腐化が起こりやすい商品と異なり、より漸進的なイノベーションや知識の集積を必要とする商品に該当する。従って、技能や業務フローの標準化が発達していることに加え、長期的雇用や労使関係によって複雑な組織内外のコーディネーションや知識や技術のすり合わせが可能であったことが、ドイツ国内で高度な ERP 製品やほかの業務系ソフトウェアが数多く開発されたこと関係していることが予想される。

表 1　ソフトウェア企業のサブセクター別分布

	ドイツ	イギリス	スウェーデン
業務系ソフト	54（90）	23（26）	20（44）
スタンダードソフト	3（5）	58（66）	16（34）
ミドルウェア	3（5）	7（8）	10（22）
合計	60（100）	88（100）	46（100）

数値は各国の上場企業数（カッコ内は比率）.
出所 :Casper and Whitley（2004）

これらの制度的特徴については日本にも当てはまる点があるが、ドイツと比べると、従業員の技能形成は基本的に各企業内の OJT が中心で技能や業務フローの標準化が低いこと、配置転換を多用する人事制度により職務知識の専門性も高まらなかったことなどから、日本においては産業や職種による技能の共通性は低水準である。低い雇用流動性と企業特殊技能の組み合わせは、むしろ、汎用性のない企業独自のソフトウェア開発とその利用を助長する結果となり、共通の ERP の広範囲な普及を妨げてきた要因の一つとなっているのではないだろうか。更に、日本型コーディネーションにおける賃金交渉は、春闘などタイミングを合わせて行うものの、最終的な妥結額は各企業に任されており、同様の産業や職務における企業間、或いは、企業規模間の賃金格差は歴然としている。自動車など一部の加工組立型産業においては、そのような賃金格差や系列・下請け構造から奏功した面もあるが、ソフトウェア産業においては、作業者と最終顧客の距離が拡大することに起因する非効率性、また、アジャイル開発が困難になりやすいなど不利な点が多く、日本の制度環境から裨益したとは言い難い。

ちなみに、スウェーデンはドイツや日本と同様に、長期的な労使関係や企業間コーディネーションを有する市場経済として分類されてきたが、90 年代から 2000 年代にかけてエリクソンなどの代表的企業が率先して、プログラミング言語をオープンソース化し、人事制度もより柔軟性の高いものに変革することで、ミドルウェアなどより技術変化の激しい分野で多くの企業を誕生させている [Casper and Whitley 2004]（表 1）。アメリカについて言えば、高い雇用流動性やリスクキャピタルの豊富さがあるものの、長期的な労使関係や企業間コーディネーションの欠如から技術や知識のすり合わせや標準化は得意分野とは言えない。

5 ドイツの教育訓練制度

5-1 教育制度の概要

ここで、制度の中でも技術者の養成と最も関係が深い教育訓練制度についてより詳細に議論する。ドイツの教育制度は複線型であり、通常 4 年の基礎学校を卒業すると（年齢にして 10 歳、または、州によっては 2 年の観察期間を経て 12 歳）、学業成績や適性に基づき、基幹学校、実科学校、ギムナジウムという 3 つの進路のどれかを選択することになる（70 年代に総合学校と呼ばれる進路包括的な学校も設立されたが通学者は限定的である）。

基礎学校において成績優秀な学生はギムナジウムに進学し卒業試験に合格すると大学進学資格であるアビトゥーア（Abitur）を取得し、大学など高等教育機関に進学する。ギムナジウムは、日本の中高一貫教育に相当し、戦前はエリート輩出のための教育機関と位置付けられていたが、戦後は教育の大衆化と共に増設され、進学者数が増加している。そのため、ドイツでは職業教育が発達していたこともあり、大学進学率は低水準であったが、2000 年以降は 30％を超え、OECD 諸国の平均に近づいてきている。

5-2 職業教育訓練制度

基幹学校や実科学校を卒業する生徒の多くは、その後、職業訓練を受ける。18 歳まで定時制の通学義務があるため、彼らの多くは、職場で OJT を受けながら近隣の職業学校に通うデュアルシステムと呼ばれる制度に参加する。デュアルシステムは中世以来の徒弟制度と学校における職業教育を結び付けることで 19 世紀から制度化が進展し、戦後は 1969 年の職業教育法に基づき運営されている。

昨今、急速な技術変化や産業構造の変化、少子化や高学歴化を受け、デュアルシステムにおいて企業が求める人材や若者が望む職種が変化している。その結果、職業訓練は、高度な専門知識や理論を必要とする分野と従来通りの OJT

を中心とする分野に二極化してきている [Bosch 2010，山内 2016]。前者においては、大学進学資格を研修生受け入れの条件とする企業が多く、IT や金融など成績上位者が集まる業種においては、6 割から 7 割が大学入学資格を保有している。更に、そのような産業においては、ギムナジウムを優秀な成績で卒業した若者を職業訓練に惹き付けるため、初期職業訓練資格と大学の学位を同時に取得できるデユアルスタディプログラムと呼ばれる制度を提供する企業もある。

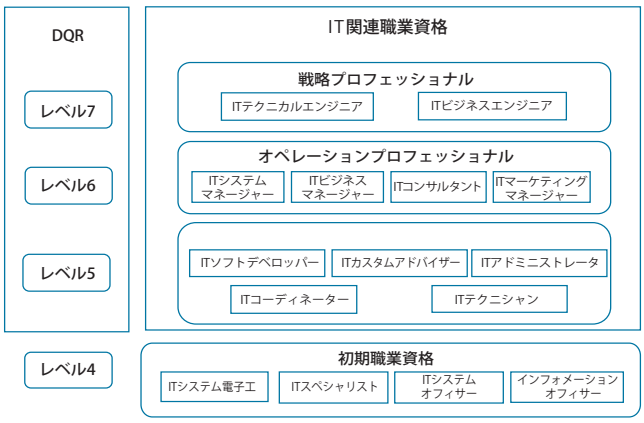
デユアルシステムは、通常 2 ～ 3 年、IT 関連職種では 3 年を要し、修了後は一人前の技術者として扱われる公式の職業資格が付与される。また、とくに大企業においては、79％の訓練修了者（前述のデュアルスタディプログラムでは 90％）が訓練先企業に就職することから、企業にとっては将来の社員を、研修生にとっては将来の就職先を見極めるための研修期間とも言える。

職業訓練制度の対象となる公認職種は 60 年代に 900 程度あったが、その後大幅に整理統合され、現在は 350 程度となっている。技術の急速な変化を受け新規職種の追加については 2 年、既存職種の改訂については 1 年以内に行うようプログラム更新のスピードを早める決定が労使などソーシャルパートナーの間で合意されている。労働者側も自らの雇用を守るため新技術の獲得には余念がない。1980 年代には、金属、電気など、技術変化の大きい分野の職種が統合され､IT など重要分野が新たに追加されたが､とくに、IT 関連の研修カリキュラムはその重要性から 9 カ月という異例の早さで開発され、3 年間の初期職業訓練プログラムを修了した研修生は、イギリスであれば大学卒業者が就くような職種に採用されている [Steedman et al.2003]。訓練内容も時代の要請に応えるためにタイムリーに改訂されており、筆者が 2015 年 7 月に訪問した BIBB の IT 部門責任者によれば、当時、ビッグデータやクラウドなどを追加することが検討されていた。

IT 関連職種は、デュアルシステムによる初期職業訓練制度において大変人気の高い職種であり、現在約 15,000 人の研修生が参加している。この分野では、4 つの職種（IT システム電子工、インフォメーションオフィサー、IT システムオフィサー､IT スペシャリスト ）が認定されているが、そのうち IT スペシャリストについては、研修科目の 2/3 が共通科目、残りの 1/3 はアプリケーションデベロップメントとシステムインテグレーションのどちらかを選択できる .

研修のカリキュラムは、前述の通り、訓練規定に基づき、ベンダ、ユーザにかかわらずすべての参加企業が同じプランに従うため、異なる企業で訓練を受ける研修生の間で一定水準の技能が共有され、技能や作業プロセスの標準化に一役買っていることが予想される。他方、企業ごとのニーズも反映できるようカリキュラム編成は柔軟でもあり、例えば、コンピューター言語を学習させる時間は規定されていても、どの言語を選択するかについては、各企業に任されている。また、デユアルシステムの研修生は企業で働きながら、近隣の職業学校に通学するため、同業種で研修を受けるほかの若者と人的ネットワークを構築することもできる。

初期職業訓練においては、かつては、実務と理論が別々に教えられることが多かったが、最近のカリキュラムでは、より実践に近い研修を行うことが奨励されている。例えば、研修生が小さなプロジェクトを仕切るようなプログラムが組み込まれており、まず、顧客の要件を聞き出し、プロジェクト計画を立て、社内の開発チームと連係し、開発を行い、テストやシステム導入を行うなど、実務に類似した手順に従い研修が行われるそうだ。試験についても、自分が担当したプロジェクトについて、どのような背景からハードやソフトのソリューションを選択したかなど実践に沿った質問がされ、それに対する回答の妥当性などを、実務経験豊富な専門家によって評価され、合否が決定される。



出所 :[BIBB 2015]をもとに筆者作成

図 1　IT 関連職業資格とドイツ資格枠組み（DQR）

IT 分野はまた、継続訓練が最も発達した産業の一つでもある。3 年の職業訓練を修了し最初の職業資格を取得した後も、引き続き上位の職業資格を目指すことができる。初期職業訓練修了後、数年間実務をこなすとスペシャリストの資格を取得し、中間管理職に相当するポストに就くことができる（図 1）。スペシャリストは､ソフトウェアデベロッパー、カスタムアドバイザー、アドミニストレータなど 5 つのプロファイルに大別され、更に合計 14 の資格に分類

される。スペシャリストとしての研修を受けるためには、ただ実務経験があれば良いということではなく、特定の資格に関連するプロジェクトを経験しておく必要がある。

その後の職業訓練については、更に上を目指すものはオペレーションプロフェッショナルの研修を受ける。これは、8段階から成るドイツ資格枠組み（DQR）のレベル6に相当し、一般教育の大学（学士）レベルに相当する。製造業や手工業の職業資格で言うとマイスターに相当する。更にその上を目指すものは戦略プロフェッショナルと呼ばれる資格を目指す。これはDQRの7に相当する。このように職業訓練は若年層だけでなく能力とやる気のあるものには長期にわたって提供される。ただし、レベルが上がるにつれ、難易度も増すためだれでも参加できるというものではない。また、職業資格の有無が通常賃金水準やポストに反映されるため資格の管理は厳格である。

ドイツの職業訓練制度は世界的に有名であり、日本においても日本版デユアルシステムの導入が検討されたことがあるが、産官学のソーシャルパートナーの連携を基礎とするドイツ型デユアルシステムを模倣することは容易ではない。筆者が訪問した企業（ティッセンクルップ社）の話では、デュアルシステムの話を聞くために、EU各国やロシアなどからしばしば専門家や企業関係者が訪問するそうだ。

5-3 高等教育制度

次に、一般の高等教育機関における教育について概観する。ドイツの大学は大別すると総合大学と専門大学の2種類がある。前者は伝統的な大学であり、幅広い学問を網羅する。後者は、技術、経営、福祉など専門科目をカバーし、より実践的である。

ドイツの教育機関は、初等教育から、大学、大学院などの高等教育までほとんどが公立である。ギムナジウムの修了証明書であるアビトゥーアがあれば、原則どの大学にも入学できるため、これまで、大学間のランクについて語られることは少なかった。最近は、大学大衆化の影響で生徒数が増加すると共に生徒間の学力差が顕著になっており、自然科学系の科目のように実験器具に制限がある分野に加えそのほかの人気分野でも、アビトゥーアの点による足切りなどを用いた学生の選別が行われている。国際的に競争力の高い研究大学を創出する目的から、特定の大学により多額の研究費を割り当てるイニシアティブも始まり、大学間の序列が次第に顕在化しつつあるという意見がある。また、学問分野における国際競争力を高めるために、自然科学分野ではフランホーファーやマックスプランクといった先端技術を扱う研究機関と大学を連携させ、若い優秀な人材の輩出を促すようなことも図られている [坂本 2006]。

大学間のランクが明確でないドイツにおいては、優秀な若者は早期に進級し上の学位を目指すことが多い。大企業の人事部長を対象とした調査で、企業が大卒者の採用において重視する項目を見ると、「実務経験」や「卒業成績」などと並んで「在学期間」が「大学名」より上位にランクされている [吉川 2004: 193]。また、フランス、ドイツ、イギリスの上位200社のトップマネジメントの学歴に関する調査によると、ドイツでは、イギリスのパブリックスクールやフランスのグランゼコールなど特定のエリート教育機関への集中は見られないものの、トップマネージャーの45%が博士号の取得者であった [ウィッティントン 2008: 82]。すなわち、経営者の専門性が高いこともまたドイツの特徴であり、産学連携を後押しする要因の一つであろう。

ドイツ公共職業安定所 [Bundesagentur für Arbeit 2015] のレポートによれば、2013年時点で情報工学専攻の高等教育学位取得者数は21,200人であり、2003年の7,800人から10年間で3倍近く増加している。情報工学は専門大学と総合大学の両方で学ぶことができるが、専門大学の卒業者が52%と総合大学の卒業者を若干上回っている。ドイツでは高等教育が実質無料であることなどから、入学後に専攻を変更する学生や途中で退学する学生は多く、一学期の登録者数は卒業者数をはるかに上回る。2013/14年度には、前年度を7%上回る58,000人の学生（つまり、その年の修了者の2倍以上）が情報工学課程に履修登録している。

クスマノ [2004] は、プログラム言語や設計原理などコンピューターサイエンス分野のヨーロッパの大学教育を高く評価しているが、ドイツの高等教育においては、更に理論と実践の結合が重視されている。大学や専攻分野により方針は異なるが、数カ月の実習期間が課されるか、推奨されることが多く、とくに専門大学では、通常、一学期をすべて実習にあてる実務実習学期が設けられている [吉川 2004: 188]。

6 おわりに

本稿においてはSAPの世界的躍進がドイツの制度環境から大いに裨益したことを明らかにした。公式な職業訓練に基づく共通性の高い技能や業務プロセス、高度でありながら実践を重視する高等教育、集団交渉の協約適用拡大により下請け構造が困難となる低い企業規模間の賃金格差、90年代以降加速した海外進出、企業価値を重視するコーポレー

トガバナンスなど、どれを取っても統合基幹業務システムの開発・導入を後押しする背景があった。

日本企業のグローバル化が遅ればせながら進展しているが、「ネットワーク外部性」を考慮すると、既に多くの海外企業で活用されているシステムを導入する利点は大きく、日本の IT ベンダによる苦戦が予想される。日本でも製薬産業などクロスボーダーの企業買収が日常茶飯事の業界では主要企業のほとんどが 90 年代に SAP を導入している [鈴木他 2001: 31]。

すなわち、技術者個人の能力や個別企業の努力を超えたより大きな制度環境やイノベーションシステムによって商品の国際競争力は大きく規定される。国の産業政策や企業の商品戦略はそのことをより意識しなければならない。

謝辞

本研究の一部は、独立行政法人情報処理推進機構（IPA）「2014 年度ソフトウェア工学分野の先導的研究支援事業」の委託に基づいて行われた。

【参考文献】

[BIBB 2015] Bundesinstitut für Berufsbildung,Aus-und Weiterbildung in den IT-Berufen,BIBB,2015.

[Bosch 2010] G.osch,'The revitalization of dual system of German vocational training',in Bosch,G. and Charest,J.(eds.) Vocational Training: International Perspective,Chapter 6,pp.136- 161,Routledge,2010.

[Bundesagentur für Arbeit 2015] Bundesagentur für Arbeit Statistik,Der Arbeitsmarkt für IT-Fachleute in Deutschland,Mai 2015.

[Casper and Whitley 2004] S.Casper and R.Whitley,'Managing competences in entrepreneurial technology firms: a comparative institutional analysis between German,Sweden and the UK',Research Policy,33,pp.89-106,2004.

[Grimshaw and Miozzo 2006] D.Grimshaw and E. Miozzo,'Institutional effects on the market for IT outsourcing: analyzing clients,suppliers and staff transfer in Germany and the UK',in Miozzo,E.and Grimshaw,D.(eds.) Knowledge Intensive Business Services: Organizational Forms and National Institutions,Chapter 6,pp.151-186,Edward Elgar Publishing Limited,2006.

[Hall and Soskice 2001] P.Hall and D.Soskice,Varieties of Capitalism: The Institutional Foundations of Comparative Advantage,Oxford University Press,2001,（邦訳：遠山弘徳 , 我孫子誠男 , 山田鋭夫 , 宇仁宏幸 , 藤田奈々子 ,『資本主義の多様性：比較優位の制度的基礎』ナカニシヤ出版）.

[Lane and Bachmann 1996] C.Lane and R.Bachmann,'The social constitution of trust: supplier relations in Britain and Germany',Organization Studies,17 (3),pp.365-95,1996.

[Lehrer 2006] M.Lehrer,'Two types of organizational modularity: SAP,ERP product architecture and the German tipping point in make/buy decision for IT services',in Miozzo,E. and Grimshaw,D. (eds.) Knowledge Intensive Business Services: Organizational Forms and National Institutions,Chapter 7,pp.187-204,Edward Elgar Publishing Limited,2006.

[Lundvall 1992] B-Å.Lundvall,National systems of innovation: Toward a Theory of Innovation and Interactive Learning,Pinter Publishers,1992.

[Nelson 1993] R.Nelson,National Innovation Systems: A Comparative Analysis,Oxford University Press、1993.

[Statista 2015a] The Statistics Portal,Share of the enterprise resource planning (ERP) software solutions market worldwide,as of October 2015,by vender,2015。

[Statista 2015b] Das Statistik-Portal,Marktanteile der führenden Anbieter am Umsatz mit Enterprise-Resource-Planning-Software (ERP) in Deutschland von 2011 bis 2013.

[Steedman et al.2003] H.Steedman,K.Wagner and J. Foreman,'The impact on firms of IT skill-supply strategies: An Anglo-German comparison',London School of Economics,2003.

[Streeck 1996] W.Streeck,'Lean production in the German automobile industry: a test case for convergence theory' in Berger,S.,Dore,R. (eds.) National Diversity and Global Capitalism,Chapter 5,pp.138-178,Cornell University Press,1996.

[Streeck 2009] W.Streeck,Re-Forming Capitalism － Institutional Change in the German Political Economies,Oxford University Press,2009.

[岩本 2015] 岩本晃一 ,「「独り勝ち」のドイツから日本の「地方・中小企業」への示唆 : ドイツ現地調査から」,RIETI Discussion Paper Series 15-P-002, 2015.

[ウィッティントン 2008] リチャード・ウィッティントン ,『戦略とは何か？　本質を捉える 4 つのアプローチ』（邦訳：須田敏子 , 原田順子）, 慶應義塾大学出版会 ,2008.

[大塚 2010] 大塚忠 ,『ドイツの社会経済的産業基盤』, 関西大学出版部 ,2010.

[木下 2013] 木下信行 ,「「我が国企業の低収益性等の制度的背景」の模様」, 日本銀行金融研究所　ディスカッションペーパー No.2013-J-2,2013.

[クスマノ 2004] マイケル・クスマノ ,『ソフトウェア企業の競争戦略』,（監訳：サイコム・インターナショナル）, ダイアモンド社 ,2004.

[坂本 2006] 坂本明美 ,『海外・人づくりハンドブック　ドイツ　技術指導から生活・異文化体験まで』, 海外職業訓練協会 ,2006.

[鈴木他 2001] 鈴木広子 , 安田一彦 ,「医療用医薬品業界の企業情報システム化戦略 : 統合基幹業務システム ERP 導入の実態分析からの考察」, Journal of the Japan Society for Management Information, 10(1), pp.29-42,2001.

[吉川 2004] 吉川裕美子 ,「ドイツ高等教育とインターンシップ―大学生の職業への移行」寺田盛紀（編）,『キャリア形成就職メカニズムの国際比較―日独米中の学校から職業への移行過程』,pp.182-195, 晃洋書房 ,2004.

[ドーア 2001] ロナルド・ドーア ,『日本型資本主義と市場主義の衝突―日・独対アングロサクソン』,（邦訳：藤井真人）, 東洋経済新報社 ,2001.

[中村 2015] 中村吉明 ,「公的研究機関の研究マネジメント－産業技術総合研究所とフラウンホーファー研究機構のケーススタディ―」,MOT 学会 ,2015.

[日経 BP 2014] 日経 BP ビジョナリー経営研究所 ,『SAP －会社を , 社会を , 世界を変えるシンプル・イノベーター』, 日経 BP 社 ,2014.

[野村総合研究所 2014]「ドイツにおける資本市場改革及び金融機関の対応等に係わる調査　報告書」,2014.

[METI 2012] 経済産業省 ,『2012 通商白書－世界とのつながりの中で広げる成長のフロンティア』,2012.

[山内 2016] 山内麻理 ,「ドイツ職業教育訓練制度の進化と変容―二極化とハイブリッド化の兆し」『日本労務学会誌』17(2), pp. 37-55, 2016.

[吉森 2015] 吉森賢 ,『ドイツ同族企業』,NTT 出版株式会社 ,2015.

IEC 62853と「つながる世界の開発指針」
Open Systems Dependabilityの観点からの考察

DEOS 協会 技術部会／パナソニック 中川 雅通　DEOS 協会 技術部会／富士ゼロックス 山浦 一郎
DEOS 協会 標準化部会／株式会社ソニーコンピュータサイエンス研究所 森田 直
DEOS 協会 標準化部会／神奈川大学 武山 誠　DEOS 協会 標準化部会／神奈川大学 木下 佳樹

変化に対応してサービスを継続できるシステムの指針として、OSD：Open Systems Dependability の考えに基づく国際標準 IEC 62853 が今年発行された。一方、IoT 分野の開発において安全安心の確保のための指針として「つながる世界の開発指針」がある。本稿では、OSD の概要と、「つながる世界の開発指針」を OSD の観点から考察した内容を紹介する。

1 はじめに

現在のシステムは、利用者の期待、環境、技術などの様々な変化に直面している。そのためシステムが長期間サービスを提供し続けるには、運用開始後でも変化に対応し、適応、成長し続けなければならない。変化によく対応できるシステムの提供、継続のために、一般社団法人ディペンダビリティ技術推進協会（DEOS 協会）※1 は「OSD：Open Systems Dependability」※2※3 の考え方を基本とし、その実用化研究、概念の普及、標準化などを推進している。その結果を反映し、対象分野によらない汎用の OSD 要件の国際標準 IEC 62853 Open Systems Dependability ※4 が今年発行された。

一方、IoT 分野、つまり様々なモノがつながって新たな価値を創出していく『つながる世界』では、安全安心の確保が問題となっている。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（IPA/SEC）※5 は、IoT 分野の開発で安全安心に関して最低限考慮すべき事項を「つながる世界の開発指針」※6 としてまとめている。

OSD と「つながる世界の開発指針」は、つながり変化する世界で機能やサービスを継続して提供し続けるという共通の課題に取り組んでいる。本稿では、汎用の OSD の観点から、IoT 分野を対象とした「つながる世界の開発指針」を考察して得られた知見について報告する。詳細は、DEOS 協会の技術資料※7 に記載している。

2 OSDの概要

OSD では、従来別々に扱われていた開発と運用・保守を、変化に対応するための一体の活動として考える※8。運用の知見からの改善の開発なども含むこの活動は、システムライフサイクルの各ステージで独立して行われるものではない。密接に連携しフィードバックし合うステージすべてで継続して行われる、サービスを提供し続けるための活動である。OSD の要件は、合意形成、説明責任遂行、変化対応、障害対応の各目的を達成する 4 つの「プロセスビュー」のそれぞれが、ライフサイクル全体の中で実現されていることである。

2.1 OSD の 4 つのプロセスビュー

以下に OSD の核となる 4 つのプロセスビューの目的について説明する。

■ 合意形成プロセスビュー
- システム、システムの目的、目標、環境、性能、ライフサイクル、及びこれらの変化に関する共通理解と明示的合意を確立し、維持する。

■ 説明責任遂行プロセスビュー
- 合意事項違反と、違反によってステークホルダと社会一般にもたらされる帰結（説明責任者に課される救済義務を含む）との間の対応関係を確立することで、合意実現の公算を増し、システムに対する確信と信用を保ち、潜在的な被害に対する救済措置を確保する。

■ 障害対応プロセスビュー
- 障害に際してもサービス中断と損害を最小にとどめ、その状況のもとで最も適切なやり方で、可能な限りサービス提供を続ける。

■ 変化対応プロセスビュー
- 要求事項、環境、目標又は目的が変化しても、システムを「目的にかなった（fit for purpose）」状態に維持する。

2.2 4 つのプロセスビュー間の関係

これら 4 つは、独立してあるものではなく、お互いに関連し合っ

※1　一般社団法人　ディペンダビリティ技術推進協会, “DEOS協会,” http://deos.or.jp
※2　M. T. (ed.), Open Systems Dependability: Dependability Engineering for Ever-Changing Systems, Second Edition, CRC Press, 2015.
※3　所眞理雄(編), DEOS, 変化しつづけるシステムのためのディペンダビリティ工学, 近代科学社, 2014.
※4　IEC 62853 : 2018 Open systems dependability.
※5　IPA 独立行政法人 情報処理推進機構　ソフトウェア高信頼化, “IPA/SEC,” https://www.ipa.go.jp/sec/
※6　IPA/SEC, つながる世界の開発指針（第2版）, https://www.ipa.go.jp/sec/publish/tn16-002.html :IPA/SEC, 2017.
※7　DEOS協会技術部会, “IEC 62853と「つながる世界の開発指針」の比較検討,” http://deos.or.jp/link/obj/pdf/DEOS-TR-20180125.pdf
※8　DEOS協会, “はじめてみるIEC62853の実装,” DEOS協会, 2018, http://deos.or.jp/link/obj/pdf/Introduciton62853Implementation-DEOS20180605.pdf

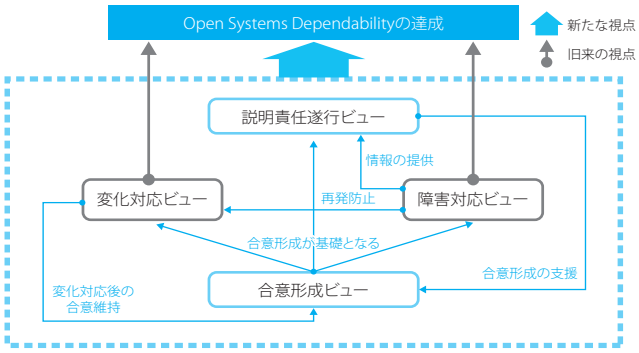


図 1　4 つのプロセスビューの関係

ている（図 1）。

合意形成はほかの 3 つのプロセスビューのベースになる。説明責任遂行はシステムに対する確信と信頼の根拠をステークホルダと社会一般に与え、また障害対応後と変化対応後の次の合意形成を支援する。障害対応は、再発防止のための変化対応を起動する。変化対応は、合意形成を再スタートさせて合意を維持し、システムを「目的にかなった」状態に保つ。合意は継続的に形成され維持されるべきものである。

3 OSDの観点からの「つながる世界の開発指針」の検討

「つながる世界の開発指針」[※6]は、5 つの大項目に分かれた 17 の指針から構成される（表 1）。

それらについて OSD の観点から検討を行った[※7]。そこで得られた知見を指針の大項目ごとに幾つか紹介する。

3.1 方針：つながる世界の安全安心に企業として取り組む

指針 1 にある「経営者が基本方針を策定し、社内に周知すると共に、継続的に実現状況を把握して見直す」は、OSD の説明責任遂行に求められる「意思決定者とほかのステークホルダに意思決定から生じた結果を知らせるフィードバックループの確立」の具体化の一つになる。

3.2 分析：つながる世界のリスクを認識する

指針 4 は「リスクの分析の結果を設計に反映する」ことを求めるが、OSD の説明責任の観点からは「抽出したリスクをステークホルダと共有する」ことも要求している。

3.3 設計：守るべきものを守る設計を考える

指針 12 の「安全安心を実現する設計の検証・評価を行う」は、OSD での「障害対応の遂行」につながる。OSD では「起きた障害の実態に即して設計時の仮定を見直す」、「なされた対応処理を評価する」ために、設計時だけでなく運用時にも検証・評価を行うことも要求している。

3.4 保守：市場に出た後も守る設計を考える

指針 13 の「自身がどのような状態かを把握し、記録する機能を設ける」は、OSD の変化対応での「環境、前提、リスクなどの変化で、システムの適応が必要となり得るものの識別」の実現につながり、同じ観点となっている。

表 1　つながる世界の開発指針一覧

	大項目	指針	
方針	3.1 つながる世界の安全安心に企業として取り組む	指針 1	安全安心の基本方針を策定する
		指針 2	安全安心のための体制・人材を見直す
		指針 3	内部不正やミスに備える
分析	3.2 つながる世界のリスクを認識する	指針 4	守るべきものを特定する
		指針 5	つながることによるリスクを想定する
		指針 6	つながりで波及するリスクを想定する
		指針 7	物理的なリスクを認識する
設計	3.3 守るべきものを守る設計を考える	指針 8	個々でも全体でも守れる設計をする
		指針 9	つながる相手に迷惑をかけない設計をする
		指針 10	安全安心を実現する設計の整合性をとる
		指針 11	不特定の相手とつなげられても安全安心を確保できる設計をする
		指針 12	安全安心を実現する設計の検証・評価を行う
保守	3.4 市場に出た後も守る設計を考える	指針 13	自身がどのような状態かを把握し、記録する機能を設ける
		指針 14	時間が経っても安全安心を維持する機能を設ける
運用	3.5 関係者と一緒に守る	指針 15	出荷後も IoT リスクを把握し、情報発信する
		指針 16	出荷後の関係事業者に守ってもらいたいことを伝える
		指針 17	つながることによるリスクを一般利用者に知ってもらう

3.5 運用：関係者と一緒に守る

指針 15、16 では、リスクや守ってもらいたいことの関係者への周知が重視されている。OSD の合意形成の観点からは、更に、何にどこまで対応するかなどについて関係者との「明示的合意」を双方向のコミュニケーションで確立することも要求している。

また、障害対応に対する、関係者、一般利用者へ、「実施した障害対応が、正しい対応であったか」の説明責任遂行も要求している。更に、合意をなぜ守らないといけないのか、守らないとどうなるかについて、普段から説明をして納得を得ることも要求している。

4 考察

「つながる世界の開発指針」は、IoT システムの開発から運用へのリニアな部分に焦点を置いている。「つながる」ことは、開発時には把握できない変化を運用時に受けるということでもある。よって、障害対応が開発時に完成することはなく、運用を通じた改善が重要となる。OSD では、運用時の予期外の障害を開発時と同様に分析し、分析結果に基づき再発防止に向けシステムを改変することにより、変化に対応することを要求している。IoT システムにもそのような OSD の障害対応、変化対応の要求事項を取り入れることで、サービスの継続性、安全・安心を向上できると考えられる。

一方、関係者（ステークホルダ）について、OSD の現規定では説明責任者か否かしか区別していないが、「つながる世界の開発指針」では、開発者、運用事業者、出荷後の関係事業者、廃棄事業者、直接ユーザ、間接ユーザ、受動的ユーザなどを想定してより具体的に指針を提示している。今後、OSD を特定分野向けに具体化する際の参考となる。

5 今後の展望

様々な分野のシステム、サービスが OSD の便益を享受できるようにするには、IEC 62853 をより具体的な分野別標準に展開し使いやすい形にしなければならない。本報告で紹介した検討は、その IoT 分野向けの一歩であり、今後、更に検討を深めて IoT 分野でのガイドライン策定や標準化に取り組みたい。

編集後記

2018年7月、IPAに社会基盤センターが誕生しました。従来のIPA組織、ソフトウェア高信頼化センター（SEC）や国際標準推進センター、IT人材育成部門の一部を統合しデジタル化がもたらす社会変革や産業・人材プラットフォームの整備を支援していく活動を進めてまいります。この新センター発足にあたり「SEC journal」誌も見直しさせていただくこととなりました。SEC journalは2004年にソフトウェア高信頼化センターの前身であるソフトウェア・エンジニアリングセンターの誕生に合わせ創刊し、この間、創刊号から本53号まで、それぞれの号の特集は変わっても、機関誌の目的や編集方針は引き継がれてきました。見直しにあたり、これまでSEC journalをお読みいただいている皆様のご意見を是非頂戴したくアンケートへのご協力をお願いいたします。（下記参照）

重ねまして、長年にわたり「SEC journal」誌をご愛読いただいた皆様、ご寄稿いただいた皆様、並びに制作にあたりご支援いただきました関係の皆様に深く感謝いたします。今後とも社会基盤センター事業へのご支援よろしくお願い申し上げます。

（編集長）

編集部より

アンケートのお願い

SEC journal 見直しにあたり読者の皆様のご意見を伺いたくアンケートにご協力お願いいたします。詳細は同封しましたご案内、または下記 Web ページをご覧ください。

https://www.ipa.go.jp/sec/secjournal/info.html

SEC journal 編集委員会

夏の石狩川（北海道旭川市）

撮影：K.Hasegawa

SEC journal 第14巻 第1号（通巻56号） 2018年8月8日発行 2020年8月1日改訂

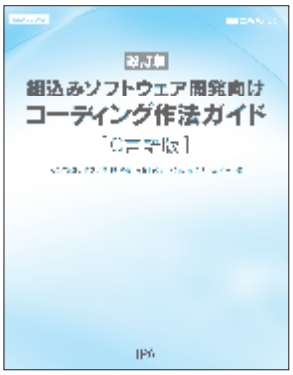
編集兼発行人　独立行政法人情報処理推進機構
社会基盤センター
センター長　片岡 晃
〒113-6591　東京都文京区本駒込2-28-8　文京グリーンコート センターオフィス16階
Tel：03-5978-7543　Fax：03-5978-7517
URL：https://www.ipa.go.jp/ikc/　e-mail：sec-journal_customer@ipa.go.jp

――[C言語版] ESCR Ver.3.0書籍化――

「改訂版 組込みソフトウェア開発向け コーディング作法ガイド［C言語版］ESCR Ver.3.0」書籍発行

本書はC言語を用いて開発されるソフトウェアのソースコードの品質をより良いものとするために、組織やグループ内のコーディングルールを決める際の参考として利用していただくことを期待して、コーディングする際の注意事項やノウハウをルール集としてまとめたものです。昨今のIoTの進展に伴い、組込み製品においてもセキュリティを意識した実装が強く求められるようになってきました。このような要請に応えるため、従来からの「C言語などを用いて組込みソフトウェアを作成する場合に、ソースコードの標準化や品質の均一化を進めること」に加えて「ソフトウェアの脆弱性作り込みを回避すること」を目的に［C言語版］ ESCR(*)の改訂版Ver. 3.0を2018年2月に公開、6月に書籍化し発行しました。これまで同様ESCRシリーズの特徴であるプログラミング初心者にも分かりやすい記述とし、規約の策定やコーディング技法、レビュー時に注意すべきポイントの説明などを盛り込んだプログラミング全般に関しても役立つガイドとなっています。

(*)Embedded System development Coding Reference



書籍版のご購入とPDF版
ダウンロードはこちらから

https://www.ipa.go.jp/sec/publish/tn18-004.html

IoT 時代に活躍する【組込みシステムの腕利きエンジニア】を目指す！

国家試験 エンベデッドシステムスペシャリスト試験

高度な実践能力の証明に！

▶ 身近な場面を想定した出題を通して、最適な組込みシステム実現のために必要となる高度な実践能力（レベル 4）を問います。

（レベル 4 の定義）：専門分野において、自らのスキルの活用によって、独力で業務上の課題の発見と解決をリードするレベル。

技術要素

プロセッサ、メモリ、バス、計測・制御、リアルタイム OS、プラットフォーム、電気・電子回路、ネットワーク、セキュリティ

開発技術

・要求分析の実行とレビュー
・設計の実行とレビュー
・テストの実行とレビュー

管理技術

・開発環境マネジメント
・知財マネジメント
・構成管理、変更管理

▶ 近年の試験では、「無線通信ネットワークを使用した安全運転支援システム」、「3 次元複写機」、「通信機能をもつ電子血圧計を用いた健康管理システム」、「非接触型 IC カードを使用した入退場ゲートシステム」などのテーマを出題しました。

▶ 自動車、家電、モバイル機器などに搭載する組込みシステムや重要インフラの制御システムを、ハードウェアとソフトウェアを適切に組み合わせて構築し、求められる機能・性能・品質・セキュリティなどを実現できる組込みエンジニアを目指す方に最適です。

試験概要

【試験区分】エンベデッドシステムスペシャリスト試験（情報処理技術者試験 高度試験の 1 区分として実施）
【日　　時】年 1 回の実施（毎年 4 月第 3 日曜日）
【申込受付】毎年 1 月中旬から 2 月下旬（予定）まで WEB・郵送で申込み受付

詳しくは、Web ページをご覧ください。https://www.jitec.ipa.go.jp/index.html
試験概要の最新情報、過去問題、活用事例などをご紹介しています。

IPA Better Life with IT

SEC journal No.53
第14巻第1号（通巻56号）
2018年8月8日発行
2020年8月1日改訂

ISSN 1349-8622