2015 年 7 月 1 日発行
第 11 巻 第 1 号（通巻 44 号）
ISSN 1349-8622

SEC journal 41

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

SEC journal No.41
Vol.11 No.1 Jul. 2015

中小企業の「IT 経営」実践をサポート
～IT コーディネータ活躍の場を拡大～

特定非営利活動法人 IT コーディネータ協会 会長
播磨　崇

IT コーディネータ協会（ITCA）は、2001 年 2 月発足以来、IT コーディネータの育成・認定のため、試験とケース研修の実施並びに資格取得後の継続研修などを推進してきた。IT コーディネータ資格認定者はこれまでに累計 1 万人を超え、全国に 200 を超える ITC 組織があり、全国各地で商工団体や金融機関、自治体、関係支援機関などと連携し、中小企業などの経営革新や自治体、各団体における改革を推進する活動を展開している。

当協会が全国各地の IT コーディネータと連携して推進する主な事業活動の内容は以下の通りである。

IT コーディネータ活躍の場を広げ IT 経営を普及促進

当協会では、全国の IT コーディネータの活躍の場を広げるための取り組みとして以下のような活動を行っている。

- 全国の信用金庫等の地域金融機関と連携した中小企業の IT 利活用支援
- 東京商工会議所等と連携し中小企業の Web 導入から高度利活用までを一貫してライフサイクル型で支援
- 全国各地で地域の ITC 組織が主催する「IT 経営カンファレンス」を開催し地元経営者や関係支援機関に IT 経営成功事例等を紹介など

最近のトピックス

2016 年 1 月から「マイナンバー制度」が始まる。税、社会保障を中心に実施されるが、とくに中小企業の本制度に対する準備は十分ではない。マイナンバー（個人番号）の漏えいなどは重大な問題に発展し、処罰の対象にもなる。当協会では、これらの中小企業のマイナンバー制の導入を円滑に進めるための対応策を講じてきている。具体的には、モデル企業を例にあげ、マイナンバーへの対応を事例で示し、手を打つべきところの勘所を分かりやすく示す研修を実施し、全国の中小企業の経営者、IT コーディネータに向けて情報発信や、研修を展開していく。

また、経済産業省が推進する「攻めの IT 経営 中小企業百選」事業の事務局を担当し、全国の中小企業の中で IT を積極的に利活用し、「攻めの IT 経営」を実践している優れた企業を選定し、その事例を広く世の中に知らしめる活動を進めている。

より高度な IT 経営支援を求められる IT コーディネータ

中小企業の IT 利活用の高度化に伴い、IT の利活用のスタイルも、クラウド、モバイルなど新しい技術の活用が求められてきている。それに伴いセキュリティの確保も重要課題となっており、IT コーディネータも適切な対応を行っていかなければならない。

また、利活用分野も、製造業をはじめとし、サービス業、農業、医業などへも拡大を見せている。これらに対応するため、当協会は、全国の IT コーディネータや識者の協力を得て研究活動を展開し、その成果として人材育成プログラムを生み出す取り組みを進める。テーマとしては、マイナンバー、クラウド、セキュリティ、農業、医業などである。

更に、IT コーディネータの継続学習のほかに、当協会会員、IT ユーザや IT ベンダや個人も対象としたセミナーや研修を実施していく予定である。

IPA との連携の強化

より一層高度な支援を求められる当協会にとって、IPA と様々な活動で連携・協力をさせて頂いていることに大いに感謝している。とくに、IT コーディネータと連携した SEC 成果物の作成・普及活動、また中小企業などにおける IT 利活用の実態調査などの共同実施、全国の ITC 組織主催セミナーへの講師派遣などでは大変お世話になっている。

また、情報セキュリティの普及啓発事業でも、中小企業に対する指導者育成や普及のためのコンテンツの提供などに関してご支援を頂いており、更にこうした活動について IT コーディネータの参画、連携を促進していきたい。

今後も、IPA 関係の皆様方のご支援、ご協力の程よろしくお願いしたい。

IoT 時代のイノベーションを支えるエンジニアリング

－IESE 所長　ディーター・ロンバック氏を迎えて－

IESE 所長
ディーター・ロンバック　×　IPA 理事
立石　譲二

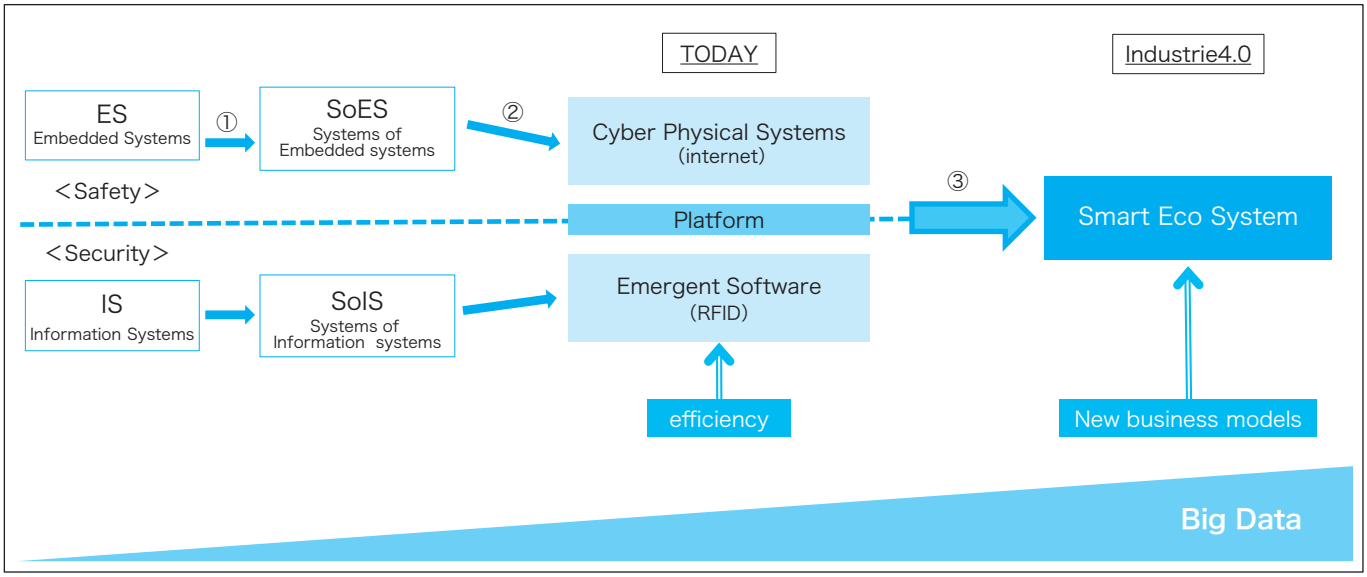
IPA/SEC 設立 10 周年にあたり、ソフトウェア・エンジニアリングの世界的権威である、ドイツ フラウンホーファー研究機構 実験的ソフトウェア工学研究所（IESE）の所長ディーター・ロンバック博士をお招きし、現在ドイツが取り組んでいる『インダストリ 4.0』の IoT（Internet of Things）、IoS（Internet of Service）構想やソフトウェア・エンジニアリングの最新動向についてお話を伺った。本対談は 2015 年 2 月に開催した SEC 特別セミナー「IoT 時代のソフトウェア・エンジニアリングとビジネスイノベーション」会場にて公開形式で実施したものである。

ディーター・ロンバック
カイザースラウテルン大学コンピューターサイエンス領域講座教授。
フラウンホーファー研究機構 実験的ソフトウェア工学研究所（IESE）設立から 2014 年まで所長、その後同所のビジネス開発ディレクターとなり、同時にカイザースラウテルンの The Board of the Science Alliance の議長に就任。Gesellschaft für Informatik (GI) のメンバーであり、ACM と IEEE のフェローも務める。主要な研究テーマはソフトウェアエンジニアリング分野である。200 以上の科学分野に関する著作を持ち、アメリカ国立科学財団の Presidential Young Investigator Award、ドイツ Rhineland-Palatinate 州のメリット勲章など多数の章を受章していて、アメリカやヨーロッパなど企業、業界、国家等に対し科学的な助言をしている。

■ IESE のプロフィール

立石：日本でも昨年あたりから「IoT」「IoS」という言葉が盛んに聞かれるようになりました。物のネットワークが急速に人々の関心を集めるようになっています。ある予測によれば、2020 年には世界中で数百億個の機器あるいはセンサがインターネットにつながると見られています。こうした IT の変化を受けて私どもの足下でも農業、医療、製造業など幅広い分野でスマート化によるイノベーションが広がりつつあります。本日のセッションでは、こうした時代に向かって従来私どもが取り組んできたソフトウェア・エンジニアリングやこれから新たに出てくるシステムズエンジニアリング、こうした工学的な手法が果たす役割について考えていきたいと思います。まず、ディーター・ロンバック博士から IESE についてご紹介いただければと思います。

立石 譲二（たていし　じょうじ）
1985 年東京工業大学大学院総合理工学研究科システム科学専攻修士課程修了（1998 年同大学院社会理工学研究科社会工学専攻博士課程修了）。同年通商産業省（現経済産業省）入省。2004 年より内閣官房副長官補（安全保障・危機管理）付内閣参事官として内閣官房情報セキュリティセンター（NISC）の新設に参画し、重要インフラ防護、政府機関のセキュリティ対策促進を主導。2008 年独立行政法人情報処理推進機構（IPA）技術本部ソフトウェア・エンジニアリング・センター（SEC）副所長、2013 年同機構理事（技術本部長）として情報セキュリティ対策、情報システム基盤の高信頼化及び電子政府における相互運用性基盤の整備等の取組みを統括。2009 年東京工業大学非常勤講師。2015 年東京大学非常勤講師。



図：IT Mega-trend：Integration

※ロンバック博士が会場でホワイトボードに手書きしたものから作成

ロンバック：まずフラウンホーファー研究機構について、そして私が所長をしております IESE についてご紹介したいと思います。

フラウンホーファー研究機構は、第二次世界大戦後間もなくの 1949 年、産業界の戦後復興を手助けすることを目的に設立されました。その使命は、すべての産業界のすべての技術について様々な研究を行い、技術移転を行うことにあります。この研究機構の下には、実に 75 もの組織があり、全体の従業員数は 23,000 人、年間予算は 20 億ユーロに達します。予算の 3 分の 2 は委託業務によるもので、そのうちの半分は公的なもの、半分が民間のものとなっています。

IESE は、実験的ソフトウェア工学研究機構の中にあります。この研究機構は 1996 年に設立されました。ドイツのフランクフルトの南西約 100km にあるカイザースラウテルンというところにあります。この研究機構のフラグシップ的な研究機関と位置づけられているのが IESE です。

IESE はソフトウェアシステムズエンジニアリングを研究の対象としており、200 人以上の従業員がいます。様々な産業セクターの、ソフトウェアを活用した製品・サービスを開発している企業と協力をしています。

具体的には、自動車、航空、医療機器、エネルギーマネジメントの各業界です。

私たちが取り組んでいるのは、物の世界とデジタルな世界の統合にフォーカスした研究であり、いわゆる IoT、スマートエコシステム、サイバーフィジカルシステムなどを領域としています。また、私たちが能力を発揮している場として、システムアーキテクチャ、セーフティエンジニアリング、セキュリティのエンジニアリング、そしてビッグデータの解析などがあります。

■ IoT 時代の本質は何か

立石：はじめに、IoT 時代のイノベーションについて考えていきたいと思います。

そもそも IoT という言葉が取り上げられる背景には色々な側面があると思います。それは単に物と物がつながるという文字通りのことだけではなくて、サイバーの空間と物理、実体の空間が切れ目なくつながるということ、また IT が持つ産業構造、ビジネスモデルに対する変革力を指して言われているところもあると思います。しかしこれらは、かなり広い概念が、断片的に伝えられているような側面があるのではないでしょうか。IoT によって起こる本質的な改革力の全体像が今ひとつ明らかになっていない、つかみにくいという気がしてなりません。これらについては、インダストリ 4.0（Industrie4.0）やアメリカの取り組みである IIC（Industrial Internet Consortium）などが紹介されていますが、まず、IoT 時代の本質は何か、その全体像について博士のお考えを伺いたいと思います。

ロンバック：全体像が見えにくいというご指摘は、全く

その通りだと思います。語る人の経歴やバックグラウンドが、IT 寄りなのかエンジニア寄りなのかによっても見方が違ってきます。全体像ではなく、そのごく一部を見ているに過ぎないということがあると思います。視点が違い、使う用語も違うという状況です。IESE においてはそうした様々な視点を分析し、それをとりまとめて全体像を再構築しました。それは、色々な部分に分かれ、様々な用語が使われている中で全体的な視野で見る手助けになるのではないかと思っています。３つのレベルのシステムの統合ということを、図でご説明しましょう。

まず、最初の部分がどのように始まったのかと言えば、ソフトウェアや IT は、２つの全く異なるものとして存在していました。１つが組込みのシステム、もう一つが情報システムです。互いに離れており、それぞれの課題も異なっていました。組込みシステムにおいては安全、情報システムにおいてはセキュリティでした。

さて、次に何が起こったか――これは私が第一のレベルの統合と呼んでいるものですが（図中①）、“組込みシステムのシステム”が登場しました。物理的なシステムが相互接続され、マシンとマシンがやりとりをするようになったわけです。また、情報システムの側でも同じことが起きました。企業の中に単体で存在したたくさんの情報システムが相互接続され、やりとりをするようになっていったのです。

このシステム間ではデータ交換が可能でしたが、そこにインテリジェンスはありませんでした。誰もが情報を活用するということはできなかったのです。そして最近になって出てきたのがサイバーフィジカルシステム（Cyber Physical System）です。これはエンジニアリングの世界の方から発信してきたものです。組込みシステムが相互接続され、全体としてのシステムになっていたのですが、それがインターネットにアクセスしました。それによってもう一方の情報システムに対してインターフェースが取られました（②）。

トラックを例にご紹介すれば、トラックにセンサが登載され、センサからデータがインターネットに送られます。トラックの中のある部品が近い将来不具合を起こしそうだ、ということが見て取れるということです。製造技術者の視点でいえば、パーツが IP を持ったというということになります。

情報システムの方でも同じようなことが起こりました。例をあげれば RFID（Radio Frequency Identification）です。倉庫の中でのロジスティックシステムにおいて、物の世界とデータが RFID を通じて、より結びつきを強めました。

これらが最終的に合体したところがスマートエコシステム（Smart Eco System）です（③）。第３のインテグレーション、あるいはインダストリ 4.0 と私たちが呼んでいるものです。組込みシステムと情報システムが同じレベルで相互にやりとりすることができるようになった世界であり、先ほどのトラックの例で言えば、走っているトラックに間もなく故障が起こるようなパーツがあったとき、ロジスティック側がそれを察知し、交換部品を用意して、故障する前に交換してしまう。それによってトラックが故障せずに走り続けビジネスを継続することができます。

トラックにセンサが付いていて、そのセンサのデータが、あるパーツが今後２週間で故障するかもしれないというデータを持つようになる。実際に故障しそうだとなったら、ロジスティックシステムの方でそのパーツを入れて直すことになる。これは２番目のレベルでの話です。

これに対してスマートエコシステムでは、今までの経緯といった歴史的なデータが蓄積されているので、それによって予測ができるようになります。要するに、故障する前に取り替えることができる。故障しそうだとなったら、一番早くその部品を調達できるロジスティックに連絡をして、それを持ってくることになります。

具体的にスマートエコシステムでどういうことが起きるかというと、トラックがどういったルートのプランで走っているかというデータまで、いち早く得ることができるので、そのトラックがどこにいるときにそのパーツの故障が起こるのか、そこまで予測することによって、例えばトラックがドイツ国外だった場合、そのパーツを送って、ドライバーが休んでいる合間に部品の交換をすませてしまい、一刻の無駄もなくなるということです。機会を無駄にすることなくビジネスの効率を上げることができます。

第２のイノベーションでは、ビジネスモデルは変わらず、

その効率が良くなっただけですが、スマートエコシステムになればビジネスモデルそのものが変わることになります。トラックのシステムから送られてくるデータがあることによって、トラックの製造だけでなくその後のサービスまで行うことができるようになるわけです。
この新しい世界になると、ドイツや日本で、新たな仕事がたくさん生まれます。スマートエコシステムでは、実際には例えばグーグルがこのサービスを提供する可能性をもっています。グーグルは今奇妙な車を走らせて実験をしていますが、彼らは新たなプロダクト、新たなモビリティの発展性を追求しているのです。一般の目には奇妙で滑稽に見える自動走行の実験にしても、彼らは非常に真剣に取り組んでいる。それはなぜかというと、グーグルはこの本質が分かっているからです。またソフトウェアの大切さ、ビッグデータ解析によって何ができるかを知っていてその上でやっている。この統合がうまくいけば、グーグルが、例えばダイムラーやトヨタといった自動車メーカを買収するということがあるのかもしれません。
IoT、IoS と言った時には基本的にはプラットフォームを指します。これをできれば標準化して提供したい。IoT、IoS は何かという本質はスマートエコシステムに集約されますが、ドイツや日本においてスマートエコシステムは経済発展のためのプラットフォームとなっていてエンジンです。それに対してビッグデータは燃料です。ドイツや日本では組込みシステムや情報システムからプラットフォームを持ってきたので、スマートエコシステムに進む準備ができていると思います。
組込みシステムや情報システムから始まって第２のインテグレーションが起きるところまでは、徐々によりよく改善していくという意味で、だんだんと変わってきました。つまりエボリューションと言えると思いますが、第３のインテグレーションにおいては色々なことが全く変わる、大きな変化が起こるという意味でレボリューションということができると思います。
立石：ありがとうございます。頭の中にあったたくさんのキーワードが、１枚のホワイトボードの中に整理されました。IoT やインダストリ 4.0 といった物作りの改革が、レボリューションといわれるような形で大きく進んでいる背景には、色々な舞台装置が整ってきたということがあるわけですね。
ロンバック：そうです。私がお伝えしたいことは、ビジョンを持たなければいけないということです。そこを目指してインダストリ 4.0 といった改革も起こり、メリットが生まれます。投資もしながらそのビジョンへの到達を考えていくということです。

■システムズエンジニアリングとは

立石：今から 35 年前の 1980 年頃に話題となった本に、アルビン・トフラーの『第三の波』があります。そこには、今起こっているような変化がたくさん出てくるので、今読み返しても大変おもしろく、私の好きな書物ですが、そこに、物同士がお互いに話をするようになる、そして人間と意思を通じるようになるというくだりがあります。それから「プロシューマー」という言葉が出てきます。メーカと消費者の明確な区別が崩れ、消費者であると同時に生産者にもなる。思ったものが手に入りやすくなる。究極のカスタマイゼーションが起こる。こういうこともトフラーは指摘しています。博士がおっしゃるように、IT が変革のドライバーになる、変革力につながっていくということを指していたのではないかと思っています。この本が指し示している出来事が、正に今日のモバイルコミュニケーションの発達であり、インターネットの広帯域化であり、デバイスの技術であった。こうしたすべての舞台装置が整ってきた段階で、IoT やインダストリ 4.0 といったことが実現し始めたということではないでしょうか。
さて、次に今日の対談の本題といえるかもしれませんが、ソフトウェア・エンジニアリングと、システムズエンジニアリングの関係について考えていきたいと思います。
私ども IPA/SEC は、ご存じの方も多いかもしれませんが IESE をお手本としてつくられた組織です。とくにシステムのコンポーネントとなっているソフトウェアを正しく正確に作るための手法をソフトウェア・エンジニアリングとして追求してきました。ところが近年では、システムズエンジニアリングという工学分野が着目されています。これは二者択一ではなく相互補完的に発展していく

べきものだと思っていますが、このソフトウェア・エンジニアリングとシステムズエンジニアリングについて、それらの関係や重要性、役割の変化について、ぜひお話を伺いたいと思います。

ロンバック：ご質問にお答えする前に、先ほど立石さんがまとめてくださったことについて少し触れたいと思います。アルビン・トフラーの『第三の波』について語っていらっしゃいましたが、それはインダストリ 4.0 ではなくて 3.0 の世界なのだと私は思っています。組込みシステムが複数集まってひとつの大きなシステムをなすという世界だと思うからです。これは 1970 年に始まったものです。今では生産の環境などでは標準となっています。色々な舞台装置である製品・通信が使えるようになってきたということで、工場のような閉鎖型の環境のみならず、幅広い領域の中で利用機会が広がってきたということだと考えています。

さて、ご質問についてですが、おっしゃるようにソフトウェア・エンジニアリングとシステムズエンジニアリングは、二者択一であるとか、あるいは両者が競合するというようなものではないと思います。もちろんソフトウェアが全体的な大きなシステムの一部であると考えれば、システムズエンジニアリングの一部にソフトウェア・エンジニアリングが入ると考えることもできますが、これはスマートエコシステムが出てくる前からも必要な考え方でした。

その例としてお話したいことがあります。ある企業の方に「システムアーキテクチャとして何を使っていますか」と質問したことがあるのです。それに対して示されたのはハードウェアのアーキテクチャでした。そこで「そうではなく、システムのアーキテクチャとして何を使っているか知りたいのです」とあらためて尋ねると、今度は、ソフトウェアのアーキテクチャについて返事がありました。そこで「それらを俯瞰する全体的なアーキテクチャを知りたいのです」というと、そういう物は存在していない、アーキテクチャを示すことはできないという回答でした。

以前なら、そういう形でよかったのかもしれません。システムの中で使われるソフトウェアがごく小さな一部であり、ハードウェアの割合が大きかったときには、ハードウェアのアーキテクチャがおおよそシステムアーキテクチャなのであるという考え方をとっていても問題はありませんでした。しかし今は、ソフトウェアの量がどんどん大きくなり、色々なことを実現しているのはソフトウェアであり、ソフトウェアこそがイネーブラー（何かをさせるもの）であるということが多々あるなかで、もともとあったハードウェアのアーキテクチャに、どんどんソフトウェアを上乗せし、後付けすることによって問題が引き起こされています。

システムズエンジニアリングと言う時、私はハードウェアエンジニアリングのことを指しているのではありません。ハードウェアエンジニアリングという世界は、存在はしていますが、システムズエンジニアリングというのはより高い階層で、より抽象化された部分でのエンジニアリングということです。ハードウェアとソフトウェアで実現していく機能的領域、これに関してのエンジニアリングを指してシステムズエンジニアリングと言うのだと考えています。

ソフトウェア・エンジニアリングに長けている人たちは、何を強みとしているかというと、モデリングが得意であり抽象化がうまくできるというところです。これが今、システムズエンジニアリングにも求められている能力なのです。

これまで協力関係を保ちながら IPA/SEC とは研究を進めてきましたが、過去 10 年の間に私たち IESE も変革を遂げてきました。しかしそれは、ソフトウェア・エンジニアリングからシステムズエンジニアリングへ、ということではありません。ソフトウェア・エンジニアリングから、ソフトウェア・エンジニアリングとシステムズエンジニアリングの両方をやるという形への変革です。要件を設定するに当たっても、例えばドイツの海軍の艦船に関するシステムのシステムズエンジニアリングをしていますが、その時の要件は単なるハードウェアの要件でもソフトウェアの要件でもありません。現在の IESE の組織にはソフトウェア・エンジニアリング部とシステムズエンジニアリング部がありますが、お互いに認め合って、もちろん標準化をどうするかといった問題もありますが、実践的な協力関係を維持しています。

IESE のコンピテンスをよく T 字で表現します。その縦棒

はソフトウェアの深い理解を示し、横棒はシステムへの広い理解ということを示しています。

私は大学教授と IESE 所長という二つの役割を果たしています。大学においては、ソフトウェアに何ができるかといった基本的な分野での研究をしていますが、フラウンホーファーでは企業と協力しながら、いかにしてイノベーションを起こしていくかについて話しをするので、抽象化された議論だけではすみません。コンテクストが必要になります。小さいところから理解をして、そこから始めるべきだと言っていますが、人々の傾向としては問題が大きくなり始めてからシステムズエンジニアリングの重要性に気づくということがよくあります。小さいところからしっかり考えてシステムズエンジニアリングをすることによって先に進んでいけばいいところを、10 億行ものコードから成る膨大なシステムになってからシステムズエンジニアリングがどうなっているかと考え始める。それでは遅過ぎるのです。その前に、全体像をしっかり捉えるシステムズエンジニアリングが必要になります。

立石：コンテクストというキーワードが出てきましたが IoT の時代になって様々なシステムが相互につながっている状態になり「つながる IT」と言われる時代になっています。次に何につながるのかということが見えていない世界でシステムを組んでいかなければならない。とりわけ上流工程の設計がますます難しくなってくると思います。更に、専門的な教育を受けたオペレーターとしてのユーザが使うシステムから、マニュアルすら見ない一般の消費者がユーザとしてどういう使い方をしてくるかが全く予想できないという中で、システムズエンジニアリングの役割はますます重要になってくると思います。この点について博士はどうお考えでしょうか？とくに、かつて経験していない課題について全知全能の人がビッグピクチャーを描くということは考えられないので、様々なサブシステムのデザイナーたちがビッグピクチャーを共有していかなければならないという技術的にも困難な問題が出てくると思います。そのために必要な技術として interoperability、standardization、model based design といったキーワードがあると思いますが、その点についてどうお考えでしょうか？

ロンバック：おっしゃる通り、コミュニケーション、伝え方は変わってきます。研究する方も変わっていかなくてはならないし、課題もトピックも変わってくると思います。そこでインテグレーションについて、IESE のリサーチ部門で注目し、大切にしている 6 つのことについて説明したいと思います。

まず複雑性（complexity）ということです。システムの複雑性は爆発的に増加していると言えます。ここで重要になってくるのがシステムアーキテクチャであり、モデルベースのデザインも重要になってきています。モデルなしでは進められない状況になっています。

２つめはコンテクストです。コンテクストに関するデータをセンサで集めることができるようになってきました。コンテクストを利用し、サービス品質を保障していくことが、まだ完全ではありませんが、その方向に向かっていくと思っています。

そして３つ目がセーフティとセキュリティです。先ほどもご紹介しましたが、これまでは、セーフティとセキュリティは別々に扱われてきました。しかし、両方が統合され別々に考えていくようなものではなくなってきている。インテグレーションということで、両方を扱わなければならなくなりました。

４つ目は多様性（diversity）です。システムも多様なものであり、一つの同じホモジニアスなシステム環境ではなくなってきています。ステークホルダも多く、言語も様々です。相互に接続する interoperability が重要になり、標準化も重要になってきます。標準化なくしては実現できないからです。

５つ目がスマートデータの活用です。このスマートデータの活用の背後にはビッグデータがあるのですが、私どもの呼び方としてはスマートデータです。この点についても IPA と過去 10 年間、共有してきたところがあります。データを活用するということだけでは不十分であり、きちんと目標を見据えて、目標に基づいて評価することが必要です。そしてそのアプローチを、爆発的に伸びたビッグデータに適用していくことになります。ビッグデータは将来的には、企業の資本になるものです。そのためにはデータセキュリティを考えていく上でも、コンテクストに基づいた、そしてスマートな方法でのデータ

セキュリティを考えていかなければなりません。データを持つ企業にとってメリットもありますが、同時に、プライバシー、個人情報にかかわることも考えていかなければなりません。

６つ目にユーザ体験ということです。立石さんもおっしゃったように、これまでのように訓練を受けた、しっかりとした能力を持ったオペレーター以外の人も使うようなものがどんどん増えてくるということです。資格を持った人たちだけでなく、色々な人たちが使う。その中ではユーザ体験を考えていかなければなりません。過去にはグーグル、あるいはマイクロソフトといったところだけがユーザ体験を考えていたかもしれませんが、今ではスマートエコシステムを構成するすべてのものに対してユーザ体験を重視していくことが必要です。

以上、私たちの研究がどういう領域を重視しているのかということについて６つの例をあげましたが、私たちの研究は、全体としていえば、統合によって生まれる課題に対応するための研究ということになります。

■人材育成にいかに取り組むか

立石：ありがとうございます。工学手法や技術の話になってくると、次に出てくるのはそれを実践する人材育成のことではないかと思います。今後、システムズエンジニアリングの重要性がますます増していくことは間違いないわけですが、そのための人材はかなり不足しているというのが日本の状況です。ドイツではどうか、またドイツでは人材育成にどう取り組んでいるか、力を入れている施策があれば教えていただきたいと思います。

ロンバック：その問題については、日本もドイツも同じ状況ではないかと思います。世界全体として共通の人口動態的な状況があり、それが影響を与えていることは事実だと思います。ドイツでは、この問題を克服するための３つの取り組みを行っていますので、それをご紹介します。まず１つ目は大学における教育です。

これまでのようにソフトウェア・エンジニアリングやハードウェアエンジニアリングの、一つひとつを深堀りしていく形ではなく、その全体にまたがるシステムズエンジニアリングというクラスを持つようになっていきています。ジョン・ディア社という農機具メーカがありますが、その産業者向けのもの、あるいはダイムラーなど商業的な車を想定したシステムについての修士クラスがあります。そこではシステムズエンジニアリング、あるいはカーエンジニアリングもありますが、まず先にシステムズエンジニアリングを学習していくというようになっています。なぜかといえば、現在はあまりにもシステムの複雑性が増しているので、それぞれの構成要素についてのエンジニアリングだけを個別に考えていくのでは十分ではないからです。

もちろん新たな学生を対象とした教育だけでは十分ではありません。それに並行して、既にエンジニアとして職業に就いている人に対して再訓練を施していくということを進めています。これが２つ目の取り組みです。

多くのエンジニアは今、システムを考えた上でソフトウェア開発をしなければなりません。しかし、これまでそれに関してのトレーニングを受けたことがないという状況です。そこで私は、大学において、また、フラウンホーファーの研究機構において、２年の修士コースを作っています。ボッシュやダイムラーなどの民間企業で既にエンジニアとして働いている人にこのコースに参加してもらい、ソフトウェア・エンジニアリング、システムズエンジニアリングを履修してもらっています。これは今までシステム全体を扱ってこなかった人が今後システム全体を考えていくために、これまで行われていなかったトレーニングを補うものです。

更に３つ目の取り組みは、やや微妙な問題でもありますが、技術を持つ外国人の移民を受け入れるということです。例えば中国の工学系の大学卒業生は無視できない存在になっています。

立石：ありがとうございます。本日は博士をお迎えして直接お話を伺うせっかくの機会ですので、会場からご質問があれば、ぜひお訊ねください。

（会場）：『第三の波』のお話の中で、それはインダストリ 3.0 であって 4.0 ではないとのご指摘がありました。物がしゃべるという点では、既に 4.0 の世界かと思っていたのですが、それは外にロジックがあるということなのでしょうか？両者の違いをあらためてご説明いただけますか？

ロンバック：違いはインテリジェンスのあるなし、ということです。3.0 の中でも、マシンとマシンのコミュニケーション、データ交換、または同期化ということは行われています。スマートエコシステムでは更に何を行うかというと、それ以外の情報にアクセスします。ビッグデータを活用して予測するというように、インテリジェンスにビッグデータを加えた形で見ていくことができる、それが 4.0 なのです。確かにマシン同士がやり取りをするということは同じですが、そこに先ほどお話しした“燃料”があるという点が異なっています。この“燃料”、追加的な外の情報を得てインテリジェンスをもって解釈をするということが大きな違いなのです。

先ほど挙げたトラックの例でお話しすれば、トラックにおけるインダストリ 3.0 は、トラックが生産からの情報を受けて、パーツが壊れるかどうかを認識できるというところまでです。4.0 ではそれが物流のシステムと統合され、パーツの倉庫の情報と統合される。つまり追加的な価値、インテリジェンスがあるということが 4.0 になります。

（会場）：人材育成について大学教育に関するお話がありましたが、大学より若い小中高レベルの取り組みはいかがでしょうか？これからの IoT 時代を生き抜く若者の育成ということについて、ドイツでの取り組み、あるいは博士の見解をお聞かせいただけますでしょうか。

ロンバック：おっしゃる通り、大学から始めたのでは遅いといえるでしょう。既に一定の考え方が固まっているからです。また、ドイツでも大きな問題になっていることがあります。それはエンジニアリングの世界に女性が非常に少ないということです。そこで若い世代に対して、コンピューターサイエンスという形で教育を進めています。それはいかに作るかではなく、どう使うかという教育です。使うことによってどんなベネフィットがあるのか、ということから始めています。こうしてコンピューターサイエンスについて知らせることで、そういった専門に進んだときにどんな仕事があるかということが分かれば考え方が変わってきます。例えば、医者になりたいと思う人は、医者の仕事に対するイメージを持つことで初めて医者になりたいと考えるわけです。今は、高校生に将来何になりたいか、あるいはコンピューターサイエンスを専攻したときに何になると思うかと質問したときに、描くイメージが間違っていると私は考えています。高校生たちが思い描くのは窓のない部屋でひたすらコンピューターに向かう、いわばプログラマのようなイメージです。それしかコンピューターサイエンスの世界から導き出せない。いわば 30 年前にあったような世界しか思い描けないところに問題があるのです。ところが今のエンジニアというのは、色々な産業で、例えば製造業、自動車産業やエネルギー産業のエンジニアとわたり合って、様々な活躍をしている、そういったイメージが描けないところに問題があり、イメージを変えることが必要だと考え、それに取り組んでいます。

また大学のレベルでも、コンピューターサイエンスの概念を変える必要があると考えています。今までは、コンピューターサイエンスのコンピューターの部分は、物理的な考えの部分とメカニカルなあるいはエレクトリカルなエンジニアリングの部分に分かれていて、どちらかを選ぶというようになっていたと思いますが、システムズエンジニアリングを考えたときは、それらは統合されて理解する必要があるので、二者択一ということではありません。コンピューターサイエンスは、それらのベーシックな統合的で包括的なものであると考えています。

立石：ありがとうございます。まだご質問があるかと思いますが、時間になりましたので質疑はここまでにさせていただきます。ロンバック博士、本日は長時間貴重なお話をありがとうございました。お礼申し上げます。以上で、所長対談を終わらせていただきます。

アーキテクチャ横断的要素に着目したトレーサビリティ確保によるアプリケーションライフサイクル高信頼性維持のためのアプローチ

羽部　高志[†]

クラウド時代のサービスアプリケーション開発においては，スモール・スタートによって迅速にサービスを開始し，頻繁に派生開発を継続して機能拡張していくことは重要なプラクティスとなっている．しかしながら，時間やコスト的な制約により，長期に渡るサービスライフサイクルにおいて，派生開発の高い信頼性を継続して維持することは非常に困難である．本論文は，従来からのソフトウェア機能に対するトレーサビリティに加え，アーキテクチャ実装時に生じる横断的要素に着目して，これら横断的関心事へのトレーサビリティをも確保することによって，インパクト分析の信頼性を高める手法について提案する．本手法が派生開発の生産性向上に貢献できれば幸いである．

An approach towards reliability through the application lifecycle by maintaining traceability of concerns that crosscut the architecture.

Takashi Habe[†]

In the development of cloud computing services, starting the service speedy from a small scale then expanding functionality of the service via derivational development is an important practice. It is very hard, however, to maintain derivational development with high reliability during the service lifecycle, under the constraint of a short period of time or a limited amount of cost.
This paper introduces the method to increase the reliability of impact analysis by tracing both traditional functional requirements and non-functional requirements that crosscut the architecture. This method should be improving productivity of the derivational development.

1. はじめに

近年のクラウドサービスの提供におけるビジネスモデルとして，出来る限り早くコアとなる機能のリリースを実施して，サービス利用者のニーズをリサーチしながら市場の求める機能拡張を実施していくスモール・スタートというアプローチが提唱されている．これは成功への確信をもつことが困難な新規サービスへの初期投資コストリスクと，コンペティター登場による機会損失を避け

【脚注】
† キヤノンソフトウェア株式会社 エンジニアリング事業本部 技術士（情報工学部門）
Canon Software INC. Professional Engineer (Information Engineering)

つつ素早く市場優位性を築くことを目的としており，特に大きな資本力を持たないものの，アイデアの新規性に富むベンチャー企業にとっては，親和性の高いビジネスモデルであると言えよう．

そして，当ビジネスモデルを成功させるための重要な成功要因である，短い周期での頻繁なサービス拡張を実現するためには，いわゆる派生開発の生産性向上に努めなければならない．しかし，初期のサービス立ち上げに引き続いて継続的に実施される派生開発の実行に当たっては，機能追加・修正に伴う様々なリスクが存在する．

[例 1]. 機能変更実施に当たり，プロジェクト開始時点では想定していなかったアーキテクチャに関わる変更が必要になることが実装フェーズに入ってから判明．大幅な修正と，影響を確認するための大規模なリグレッションテストが必要となってしまった．

⇒ 当初の見積を大幅に上回るコストと期間が必要になってしまう．

[例 2]. 機能追加を実施した後，検証フェーズになって初めて思わぬ性能劣化や，セキュリティ上の問題が発覚．アーキテクチャの修正のため，スケジュールの見直しが必要となる．

⇒ サービスリリース計画に致命的なダメージを与え，著しい期間損失を被ることになってしまう．

このような事態を避けるため，一般に派生開発の計画策定においては，予め追加・削除・修正される機能が，既存のシステムにおいてどのような影響を及ぼす可能性があるのかを検証するインパクト分析の実施が推奨されている．

そして，このインパクト分析を十分に機能させるためには，トレーサビリティの確保が必要不可欠であるとされている．特に障害の発生が直接人命に関わる可能性が高い医療機器用ソフトウェア開発および保守のプロセスに関する安全規格である JIS2304(IEC62304) や，自動車の電気・電子に関する機能安全規格である ISO26262 においては，要求からのトレーサビリティ確保を実現することが明確に開発プロセスに対して求められている．

表 1　要求 ID とユースケース ID 間の関係を表現するトレーサビリティマトリクスの例 .

	REQ0110	REQ0121	REQ0110	REQ0110	Etc.…
UC001		✓			
UC002			✓	✓	
UC101		✓			
UC102	✓				
UC103			✓		
UC200					
Etc.…					

具体的なトレーサビリティ確保の手法としては，表 1 に例示するトレーサビリティ・マトリクス（TM）がよく知られており，また市販のトレーサビリティツールを利用してトレーサビリティの確保を図ることも，前述の医療や自動車等の高信頼性を要求されるソフトウェア開発においては，普及が進んでいる．

しかし，開発時には十分検討された非機能要求の実装については，局所化され易い機能要求の実装とは異なり，システムアーキテクチャや，ソフトウェアアーキテクチャが複数箇所に散りばめられて実装されるケースが少なくない．このため規定に則り，正しくアーキテクチャを記載した基本設計書や，実装に繋がる詳細設計書等のドキュメントを残していたとしても，それだけでは派生開発計画時にアーキテクチャに関するリスクの検出や，問題があった場合の代替案の検討を迅速に進めることは容易ではない．

本論文においては，一般にアーキテクチャの検証手法として知られる ATAM(Architecture Tradeoff Analysis Method)[Kazman 2000] を用いて，そのアウトプットである，アーキテクチャ検証のために作成した品質特性シナリオと，実装する設計ドキュメント，プログラムソース等へのトレーサビリティを確保することにより，従来の方法だけでは可視化することが困難であった非機能要求を実装するシステム横断的な関心事に対してもトレーサビリティを保証し，精度の高いインパクト分析を迅速に可能にするための一つの手法を提示する．

2.　既存のアプローチと解決すべき課題

2.1. 要求からのトレーサビリティ（V 字モデル /W 字モデル開発におけるトレーサビリティの確保）

一般的なソフトウェア開発のプロセスを V 字あるいは W 字モデルによって表現することは，上流工程と下流工程の関係と，各フェーズ間のアウトプットと対になるその検証プロセスを明確化出来ることから，これまで広く支持されてきた（図 1：トレーサビリティツール

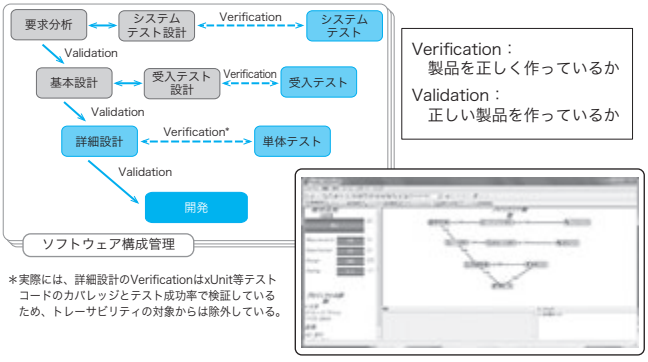


図 1　W 字開発モデルとトレーサビリティ

Reqtify でトレースした W 字モデルの例）．一般的に従来確保されてきたトレーサビリティは，設計した通りに正しく製品を作っているか (verification)，上位ドキュメントで定義した内容に基づき正しい製品を作っているか（validation）というＶ＆Ｖ検証をベースに設定することが多い．ある上流工程のアウトプットが，隣り合う下流工程で全て確実に実装されていることを検証できるようにするため，各ノード間のトレーサビリティを確保する．このノード間のトレーサビリティを積み重ねていくことで，要求からの一貫したトレーシングが可能になる．[IPA2013]

2.2. 従来のアプローチにおける課題

従来のトレーサビリティアプローチは，ウォーターフォール型の開発プロセスをベースに置き，Ｖ＆Ｖ検証を主たる目的に実施されてきた．また，従来型の自然言語をベースとした要求仕様と設計に対して，奥田等による UML(Unified Modeling Language) を用いたモデル駆動要求分析手法によって識別された要求仕様と設計仕様への適用（トレーサビリティ）が提案されている [Okuda2013].

一見，問題がないように見える従来からのトレーサビリティアプローチであるが，以下に述べる重要な課題が存在する．

すなわち，セキュリティや，スループットの確保等のいわゆる非機能要求として定義される分野の設計／実装に当たっては，インフラ基盤を含めたアーキテクチャ全体に渡る横断的な関心事として，その設計，実装がシステムアーキテクチャ，ソフトウェアアーキテクチャと階層を横断して散在するケースが多い．そしてこのようなケースにおいては，階層的に散在する実装が協調して動作することによって初めて，1 件の非機能要求を充足する仕組みになる．

このため，サービスの派生開発を実施する時点においては，機能要求における直列的なトレーサビリティを使用してのインパクト分析は実施可能であるものの，前述したようにアーキテクチャ上に散在する非機能要求へのインパクト分析を実施することが困難となる．

ゆえに，従来からのトレーサビリティアプローチだけでは，1 章の例で上げたような事象の発生を抑制することはできない．上記事象の発生を抑制するためには，

① トレースすべきアーキテクチャ横断的な関心事を洗い出し，トレーサビリティ情報を設計ドキュメントあるいは，検証結果ドキュメントにタグ情報として埋め込むと共に，トレーサビリティ情報を可視化する必要がある．

② 機能要求 / 非機能要求とその実装に変更・追加・削除が加えられる場合，トレーサビリティ情報に基づいて，変更のインパクトを検証する必要がある．

③ 変更に伴いトレーサビリティ情報の変更・追加・削除が発生した場合，トレーサビリティ情報の保守作業を継続的に行う必要がある．

本論文は，主に上記①に述べたような課題，すなわちアーキテクチャ実装における横断要素に対するトレーサビリティの可視化について述べるものであり，今後の派生開発の進展により，②③の課題についても継続して検証する予定である．

3. 課題解決に向けてのアプローチ

3.1. 直列型のトレーサビリティ＋分散型のトレーサビリティアプローチ

我々は，従来からの直列型の機能要求のトレーサビリティに加え，分散型の非機能要求の設計／実装についてもトレーサビリティを確保することで，インパクト分析の信頼性を向上させる必要がある．

この実現に当たって有効なアプローチとして着目したのがアーキテクチャ検証手法の応用である．

アーキテクチャ検証のプロセスが実施される場合，ステークホルダが関心を持つ非機能要求，即ち品質特性についてどのように解決を図っているのか，その妥当性やリスクを検証することになる．

このため，アーキテクチャ検証結果と，それら横断的な関心事を実装する設計書等とのトレーサビリティを確保することができれば，派生開発実施時にリストアップされた変更項目として，当該設計箇所が選択された場合においても，何らかのアーキテクチャに関して分散する関心事の一片がそこに実装されていることを認識することが可能になる．

更には，資産としてカタログ化されているアーキテクチャ検証結果に照らして，修正に伴うリスクやセンシティビティの検証も可能となると考える．

3.2. アーキテクチャ検証の手法（ATAM）

本研究においては，幾つか知られているアーキテクチャ検証手法の中から，カーネギーメロン大学の R.Kazman，M.Klein，P.Clements によって開発された ATAM(Architecture Tradeoff Analysis Method) [Kazman 2000] を採用して検証することとした．

3.3. ATAM プロセスとアウトプット概要

ATAM の実施プロセスの詳細については，[Kazman 2000]，[IPA2005] 等を参照されたい．ここでは簡単な

プロセスの概略と，直接トレーサビリティに関連するアウトプット，実際の適用に当たっての工夫した点について説明する．

【ATAM によるアーキテクチャ評価プロセス】

[ステップ 1]：ATAM の提示
評価リーダーによる ATAM の説明．

[ステップ 2]：ビジネスドライバーの提示
プロジェクトオーナーによるシステムの概要説明．

[ステップ 3]：アーキテクチャの提示
アーキテクトによるアーキテクチャ説明．

[ステップ 4]：アーキテクチャ技法を特定する
採用されているアーキテクチャを明確化した後，主なアーキテクチャをカタログにまとめる．

[ステップ 5]：品質特性のユーティリティツリーを作成する
品質特性の目標を明確化，詳細化，優先順位付けを行うためにユーティリティツリーを作成する．

[ステップ 6]：アーキテクチャ手法を分析する
ステップ 5 で重要と判断されたシナリオについて，それを実現するアーキテクチャ手法を詳しく調査する．

[ステップ 7]：ブレインストーミングとシナリオの優先順位付け
ステップ 6 で詳細調査した重要なシナリオについてブレインストーミングを実施した後，メンバーの投票によってシナリオを優先順位付けする．

[ステップ 8]：アーキテクチャ手法を分析する
ステップ 7 で高い優先順位付けとなったシナリオを実行する．アーキテクトは，そのシナリオを実現するために，関連するアーキテクチャ上の決定がどのような貢献をしているのか説明する．

[ステップ 9]：結果の提示
ATAM によって集められた情報を要約し，利害関係者に対してプレゼンテーションを行い，プロセスを終結する．

3.3.1. ユーティリティツリー

[ステップ 5] で作成されるユーティリティツリーは，ステークホルダが関心を持つ品質特性をツリー構造で表現したモデルである．

ここで，[ステップ 2] で提示されたシステム概要だけをインプットとした場合，ステークホルダの意識に上っていない品質特性を洗い出せない可能性がある．このようなリスクを低減するためには，網羅性の高いテンプレートを使用することが望ましい．2013 年から実施している適用プロジェクトにおいては ISO9126（表 2）お

表 2　ISO/IEC 9126 品質特性

品質特性	説明	品質副次特性
機能性 Functionality	機能とその特性に影響する	合目的性
		正確性
		相互運用性
		機密性
		標準適合性
信頼性 Reliability	定められた条件と期間においてソフトウェアがどの程度機能するかに影響する	成熟性
		障害許容性
		回復性
		標準適合性
使用性 Usability	ソフトウェアを利用するに当たり，使用者にとって必要とされる労力に影響する	理解性
		習得性
		運用性
		標準適合性
効率性 Efficiency	ソフトウェアの性能や，その実現に当たって必要とされるリソース量に影響する	時間効率性
		資源効率性
		標準適合性
保守性 Maintainability	何らかの変更を加えるに当たって必要となる労力に影響する	解析性
		変更性
		安定性
		試験性
		標準適合性
移植性 Portability	別の環境にソフトウェアを移植する可能性に影響する	環境適合性
		設置性
		共存性
		置換性
		標準適合性

表 3　ISO/IEC 9126-4　運用時品質特性

運用時における品質特性
有効性（effectiveness）
生産性（productivity）
安全性（safety）
満足性（satisfaction）

※なお，ISO/IEC9126 は，既に ISO/IEC25000:2005 に置換されているため，今後のプロジェクトにおいては，利用するフレームワークは変換する予定．

よび ISO9126-4（表 3）に定義された品質特性をフレームワークとして，適用プロジェクト非機能要求の洗い出しを行った．

完成したユーティリティツリーにおいて分解された最下層のノードは，当該システムにおける非機能要求と同義と捉え，後述する品質特性シナリオとそこに記載されたアーキテクチャの実装箇所とのトレーサビリティを確保していくこととする．

3.3.2. 品質特性シナリオ

品質特性シナリオは，[ステップ 6] で作成されるアーキテクチャ手法を分析した結果得られた，個々の品質特

SC01		異なるクラウドサービスにおいて SSO 可能			
品質特性		1. 機能性（fuctionality）	品質副次特性	1.3. 相互運用性（interoperability）	
環境					
刺激					
応答時間					
アーキテクチャ上の決定	ArchitectureID	Sensitivity	Trade-off	Risk	Non-risk
OpenAM による認証	SC0101		T01	R01	N01
	SC0102	S01	T02		
	SC0103	S02			
論理的根拠					
アーキテクチャ図					

図 2　適用プロジェクトにおける品質特性シナリオ

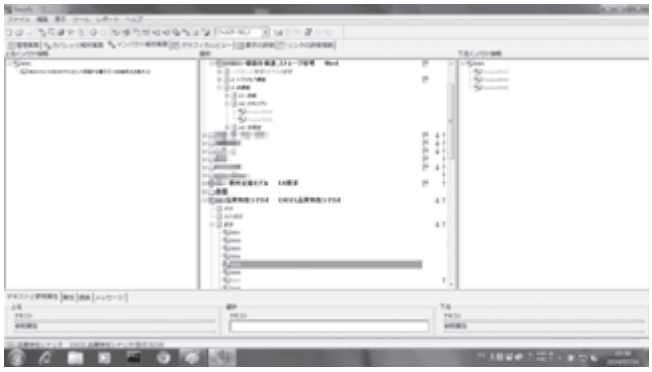


図 3　Reqtify によるインパクト分析

性実現に当たってのアーキテクチャのシナリオである（図 2）.

当該シナリオにおいて,「アーキテクチャ上の決定」としてシステム横断的な複数のアーキテクチャ上の決定が記述される．品質特性を実装するに当たって予めテンプレートを作成して使用するのだが，この時同一ファイル内の別シートに，当該シナリオ実施に当たって分析，識別したリスク，センシティビティ，トレードオフ等の情報を記載するようにしておく．1 つの品質特性シナリオ当たり，少なくとも 1 つ以上の「アーキテクチャ上の決定」があることから，決定毎にこれらリスク，センシティビティ，トレードオフが詳細化される．

即ち，この品質特性シナリオこそが，システム階層横断的な関心事を取りまとめて説明する重要な成果物となる．

3.4. 品質特性シナリオとのトレーサビリティ

品質特性シナリオによって明らかになった，システム横断的な関心事に対して，当該アーキテクチャを実装するプログラム設計ドキュメント，システム基盤設計書等とのトレーサビリティを確保する．但し，品質特性シナリオを作成するのは，全ての非機能要求に対してではなく，ATAM のプロセスの中で重要とされたシナリオに対してのみであることに注意されたい．コスト，期間とのバランスを取り，重要な品質特性についてのみ詳細化の対象とする．

具体的なトレーサビリティ確保の方法としては，機能 ID 等と同様に個々の品質特性シナリオに対して品質シナリオ ID を振り出し，更に品質シナリオを実現するための個々のアーキテクチャに対してアーキテクチャ ID を採番する．横断的関心事を実装するプログラム設計ドキュメント，システム基盤設計書等に該当する品質シナリオのアーキテクチャ ID をタグ情報として埋め込むものとする．

3.5. トレーサビリティツール

表 1 に例示した TM による可視化では，主に関連する 2 つのノード間の関係が可視化されるに過ぎない．市販のトレーサビリティツールを使用した場合，そのツールの機能および仕様にロックインされてしまうリスクが存在するものの, 表ベースで管理される TM を用いてトレーサビリティを管理するよりは一般に高い生産性とツールの機能によりインパクト分析の精度向上を期待することができる．今回の適用プロジェクトにおいては, ダッソーシステムズ（株）が提供するトレーサビリティツール「Reqtify」を採用し，各種ドキュメントに埋め込まれたタグ情報のトレーサビリティを可視化することとした．

3.6. トレーサビリティツールによる横断的関心事に関するインパクト分析

Reqtify のインパクト分析機能は，中央ペインの着目するノードを選択すると，左ペインに関連する上位のノード，右ペインに下位のノードを一目で表示するというものである（図 3）.

ここであるノードは，3.4 に述べた品質特性シナリオの実装箇所であり，予めシナリオのタグ情報が埋め込まれているものとする．Reqtify から当該ノードを選択すると，上位階層に関連する品質シナリオが左ペインに表示される．更に上位の当該品質シナリオを選択すると，選択したノードは中央ペインに，下位階層に横断的に関連する全ての実装箇所へのリンク情報は右ペインに表示されることになる．

図 4 における機能 11 を修正するケースを例に説明する．機能要求に関するトレースを考えた場合，クラス A, B の修正だけを考慮することになる. しかしここではアーキテクチャ ID11 に関してトレースが確保されているため，まずアーキテクチャ ID11 のセンシティビティ，トレードオフ，リスク等についての影響を検討することになる. 続いて, 上位階層の品質特性シナリオ 1 へのトレーサビリティを確認した後，当該シナリオの下位階層へのトレーサビリティに注目する．すると横断的な関心事としてアーキテクチャ ID12 とこれを実装するクラス D へのトレーサビリティが確認できる．機能 11 の修正に当

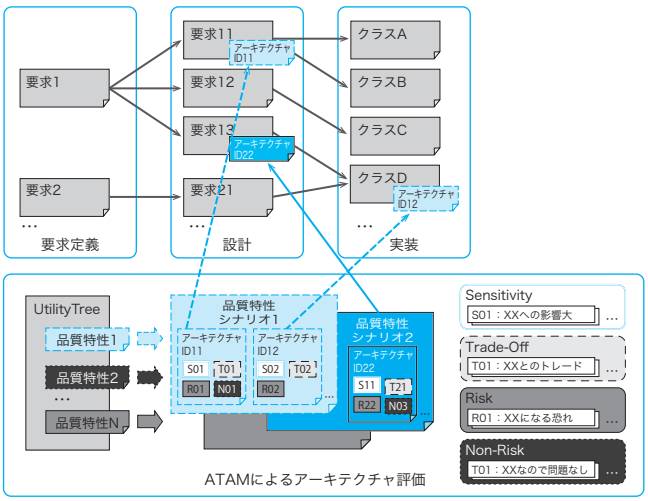


図 4 アーキテクチャへのインパクト分析

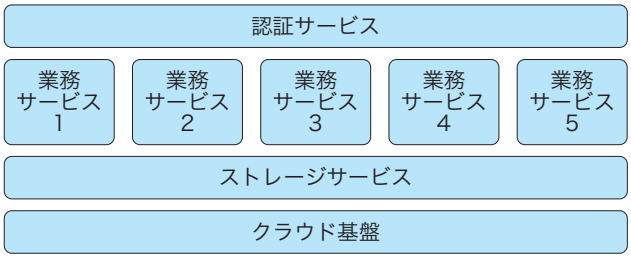


図 5　適用プロジェクトのサービス階層

たっては，これら横断的な関心事に関しても品質特性シナリオに沿った検証を実施することによって，インパクトについて高い精度で分析することが可能となる．

4.　適用と評価

4.1. 適用

これまで述べてきた ATAM 検証結果をシステム横断的な関心事へのトレーサビリティとして利用する手法について，以下のような実プロジェクトにおいて適用を実施して評価を行った．

要求定義フェーズにおいて，従来からの直列的なトレーサビリティ要素の起点となるソフトウェア機能要求として約 500 件の実装すべき機能を抽出し，合わせて ATAM によるアーキテクチャ評価の結果，ステークホルダ間によって特に重要と認識された品質特性が 24 件となったため，同じく 24 本の品質特性シナリオを作成し，アーキテクチャの検証を行った（表 4）．

また，本プロジェクトは異なる 7 つのサービスから構成され，内フロントサービスとしての認証サービス，バックエンドとしてのストレージサービスを 5 つの業務系サービスが共有して使用する構成となっている（図 5）．

ここで重要とされた品質特性の内，横断的な関心事の一例として，機能性の中の機密性の確保がある（表 5）．

表 5 の機密性確保の例においては，認証によるアク

表 4　適用プロジェクトの諸元値

適用プロジェクト	クラウドサービスの新規構築プロジェクト
開発期間	15 ヶ月
開発工数	約 170 人月
実装サービス数	7 件
ソフトウェア機能要求数	約 500 件
ATAM 実施により重要と認識された品質特性数	24 件（品質シナリオ数と同値）
トレーサビリティ設計工数	5 人日
トレーサビリティ進捗監視工数	1.5 人時 / 週

表 5　機密性の確保における例

品質シナリオ No.SC02

アーキテクチャ上の決定	Sensitivity	Trade-Off	Risk	Non-Risk
OSS XX による認証機能実装	S03		R02	
パスワードのハッシュ化保存		T03		N02
外部ストレージアクセスキーの暗号化	S04	T04	R03	N03
クライアント送信データの暗号化	S05	T05		
HDD の暗号化	S06	T06	R03	

Sensitivity

No.	説明	備考
S03	認証機能の修正に当たっては OSS XX の I/F，パラメータ設定に影響がおよぶ可能性がある	
S04	秘密鍵を変更する場合，暗号化済データの再暗号化処理が必要になる	
…	…	

Trade-Off

No.	説明	備考
T03	ハッシュは非可逆性のためシステム管理者もパスワードは不明となる	
T04	セキュリティは強固になるが鍵管理の仕組みが必要となる	
…	…	

Risk

No.	説明	備考
R02	OSS XX にセキュリティホールが検出された場合，ライセンス上，自社による修正が困難	
R03	秘密鍵が危殆化した場合，テーブル全体のデータの安全性が落ちてしまう	
…	…	

Non-Risk

No.	説明	備考
N02	ハッシュ値のみを保存するため，漏洩した場合でもオリジナルのパスワードを特定されることはない	
…	…	

セス権管理だけではなく，ある業務サービスにおいては，クライアントへサービスするデータの暗号化を実施して機密性の確保に貢献している．また，クラウド基盤の実装として，ハードディスクの暗号化を分担する等して，システム全体の機密性を担保している．このようにセキュリティについては，何処か一カ所にセキュリティホールが埋め込まれているだけでシステム全体が危殆化する恐れがある．これは，システム横断的な関心事として，リスク，センシティビティ，トレードオフについて十分な検証が必要になる一例である．

4.2. 評価

図 6 は Reqtify によってプロジェクト仕掛かり中のトレーサビリティ実装の進捗状況を表示した画面をスナップショットしたものである．

図 6 上は，Reqtify のフォルダ機能を使用して，最上位の「要求モデル」（図 6 上のノード名では「要求定義モデル」）と，「要求モデル」内に定義したユーティリティツリーから導出された「品質特性シナリオ」に対して，図 5 に提示した 7 つのサービスとクラウド基盤に関するノードをサービス毎に取りまとめた 8 つのフォルダとのトレーサビリティを可視化したものである．

ここでは，「要求モデル」と下位のフォルダ間の関連線によってソフトウェア機能要求に対するトレーサビリティを（図 6 上の黄色の関連線※），また「品質特性シナリオ」と各サービスのフォルダ間の関係において，横断的な関心事についてのトレーサビリティを追跡（図 6 上の緑色の関連線※）するようにしている．

一方，図 6 下は，図 6 上のあるフォルダの中身 (サービス毎のトレーサビリティ対象ノード) を別途表示させたものである．ここでは，最上位の「要求定義モデル」から導出した「ソフトウェア機能一覧」と「ユースケース記述書」が定義されている．「ソフトウェア機能一覧」からは機能 ID 毎に詳細機能を記述した「機能仕様書」と機能を実装した「ソースコード」が，また，「ユースケース記述書」からは，これに記載したシナリオに沿って機能の検証を行う「受入テスト設計書」に対してトレーサビリティを定義している．また，「テストケース」は，シナリオの実施手順毎に関連する機能 ID に対しても関連付けすることにより機能検証の網羅性も確保している．更に，「品質特性シナリオ」に記載されたアーキテクチャの一部について，「機能仕様書」に記載された該当箇所とのトレーサビリティを確保している．

ここで Reqtify におけるノード間の関連線の色について説明する．上位のノードに定義した ID に対して下位のノードでカバーされている割合によって色が変わる仕様になっており，カバー率が 70% 以下である場合には赤色，70 ～ 90% の場合には黄色，90% 以上になると緑色で表示される．この機能により，現在上位ノードに定義された要素が下位ノードで設計あるいは実装されたかというトレーサビリティの進捗が分かる仕組みとなっている※．

図 6 上のようにトレーサビリティ定義を行うことにより，プロジェクトマネージャは，従来からの直列系のトレーサビリティについても，今回の研究テーマとした非機能要求のような横断的な関心事についても，同時に現在のトレーサビリティの実装状況について視認性の高い形式で可視化を実現することが出来る．

一方，プロジェクトメンバは，自身が担当するサービスのトレーサビリティ確保に責任を持つ．従って，図 6 下の自身の担当するサービスについて，アーキテクチャ実装を含めたトレーサビリティの進捗についてのみ認識していればよい．

トレーサビリティを詳細に追おう

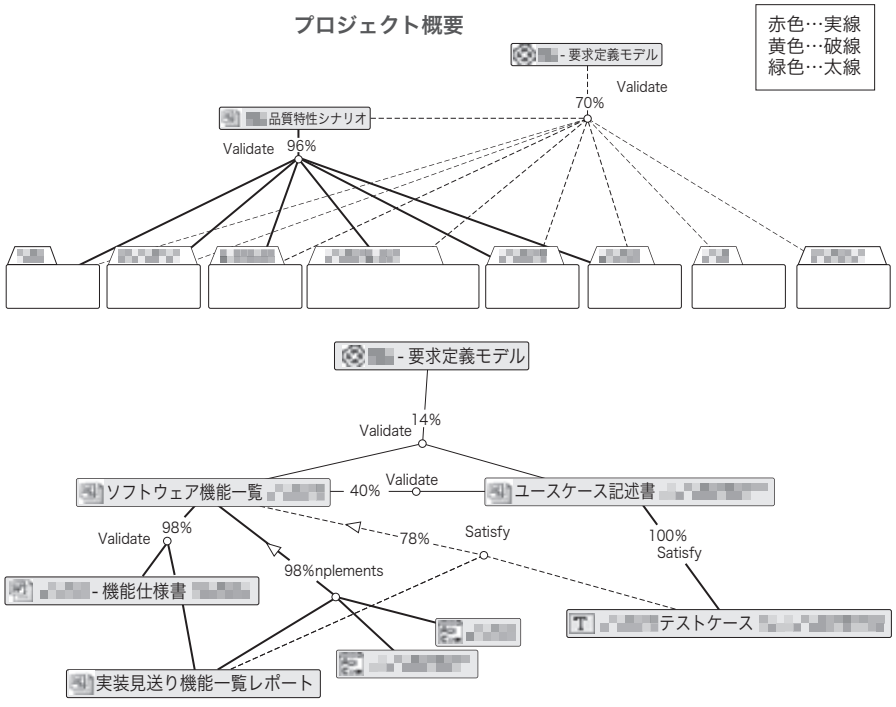


図 6 （上）：要求モデルから見た全体トレーサビリティの進捗と
（下）：個別サービスレベルのトレーサビリティ進捗

【脚注】

※ Reqtify 上では本来、関連線の色によって数値を示すが、本誌掲載にあたっては線種によりそれを示した

とすると，タグ情報が膨大となるため，PM/ プロジェクトメンバといった立ち位置に応じた粒度でトレーサビリティの進捗を可視化することが重要となると考える．

また，実装が既に完了した機能に対して変更要求が実施されることになったケースにおいて，3.6 に述べた手順の適用を試みたところ，インパクト分析に必要な関連ノードを抽出するコストは高々 1-2 人時であることも合わせて確認することができた．

5. 考察と今後の課題

5.1. 考察

トレーサビリティを開発当初から実装することにより，次期派生開発を待たなくとも，設計凍結後の変更管理の運用段階より，リスク評価の点で非常に大きな効果を発揮することが期待できると考える．

但し，開発当初から設計ドキュメントやプログラムコードに対してタグの埋め込みを実施する場合，設計の修正・変更や，プログラムの障害修正・リファクタリングの実施に伴い，トレーサビリティ情報の修正が必須となる．

今回採用した手法，ツールについては，目視によって修正に誤りがないかどうか担保するため，特に修正漏れによって古いトレーサビリティ情報が残存してしまうリスクについて十分考慮する必要がある．

また，今回提示の手法は，既存の開発資産に対する適用を実施する場合，特に初期プロジェクトにおいて，アーキテクチャ評価を実施していないプロジェクトへ後追いで適用する場合において特に注意を要する．すなわち，正しいアーキテクチャ評価が可能なインプット情報を収集する事が出来るかどうかに強く依存することに留意されたい．

このため，既存開発資産へのアーキテクチャ再評価に当たっては，ドキュメント類のインプットだけではなく，初期開発に携わったアーキテクトの参加が必須となると考える．

5.2. 今後の課題

今回のような新規適用プロジェクトについては，考察に述べた設計変更やプログラム修正時のトレーサビリティ保守が大きな課題と認識している．

今後は，差分認識に優れた構成管理ツールとの組み合わせによって，トレーサビリティのメンテナンス漏れを検出できる仕組みを検討していきたい．

また，既存の開発資産に対する適用を実施する場合，トレーサビリティの整備に必要なコストについて見積もる必要がある．

独立行政法人情報処理推進機構，トレーサビリティ確保におけるソフト開発データからの効果検証 [IPA2013] においてトレーサビリティ対策工数の試算とその検証が試みられている．但し，トレーサビリティ対策工数については，開発規模だけではなく費用対効果の観点や，プロジェクトに求められる品質レベルの観点等から，必要となるトレーサビリティの粒度は異なって然るべきである．

6. おわりに

6.1. まとめ

今回は新規クラウドサービス開発プロジェクトにおいて，アーキテクチャ横断的な関心事についてもトレーサビリティを確保することでソフトウェア信頼性の向上に対するアプローチを試みた結果について報告させていただいた．

まずは今後の派生開発に備えた初期開発の一環として，アーキテクチャ横断的な関心事に対してもトレーサビリティの対象にすることで，変更管理における効果を確認することができた．

引き続き，今後の派生開発においても当手法を継続適用することで，トレーサビリティ確保によるアーキテクチャ横断関心事へのインパクト分析が，アプリケーションライフサイクル全般の信頼性確保に対して貢献できることを検証していきたいと考える所存である．

6.2. 謝辞

今回の試みに協力していただいた適用プロジェクトの開発メンバーに対して，まずお礼を申し上げます．

また，本プロジェクトで採用した Reqtify の使用に当たって数々の助言をいただき，また検証にもご協力いただいた弊社 澤木 孝さん，舟田 学さん，論文主旨の英訳に協力していただいた弊社 和良品 文之丞さん，元上司である塚田 恒博さんに感謝の意を表します．

最後に，当論文執筆に当たり監修いただくだけでなく，様々な示唆に富むアドバイスをいただいた同じく弊社の上原 敏幸さんに対して特にお礼の言葉を捧げます．

【参考文献】

[Kazman2000] R.Kazman, M.Klein, P.Clements, ATAM: Method for Architecture Evaluation, 2000.

[IPA2013] 独立行政法人情報処理推進機構，トレーサビリティ確保におけるソフト開発データからの効果検証，pp.3-4, pp7-11, pp30-56, 2013

[IPA2005] 独立行政法人情報処理推進機構，参照アーキテクチャ調査報告（2005 年度），II-ii IT アーキテクチャ評価技法 pp.4-12, 2005

[Okuda2013] 奥田 博隆，小形 真平，松浦 佐江子，要求仕様と設計の機能要件のトレーサビリティを保持する為の Web アプリケーション設計手法の評価，情報処理学会第 75 回全国大会，pp1_441-1_442,2013

SEC2014 年度活動概要

SEC 副所長 杉浦　秀明
SEC 次長 日下　保裕
SEC 企画グループリーダー 石川　智
SEC 企画グループ主幹 江野村　亮輔

2014 年度は、IPA 第三期中期計画（2013 年度～ 2017 年度）の２年度目として、中計計画で掲げた事業目標の達成に向けた活動を加速すると共に、IoT 時代の動きを見据えた活動を進めた。本稿では、2014 年度の主な成果概要を紹介し、本稿以降で詳しい事業内容を紹介する。

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

（1）重要インフラの３産業分野で障害情報の共有体制が始動

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・一般性のある教訓事例を 28 件導き出し、対応策として類型化し、産業分野を越えて活用可能な「情報処理システム高信頼化教訓集 2014 年度版」として公開した。

さらに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、情報共有体制の拡充を目指し、業界団体などへの成果の普及展開活動を実施した。2014 年度は行政分野、電力分野、情報通信分野の３つの産業分野で情報共有体制を構築し、情報共有活動が始動した。

（2）「ソフトウェア開発データ白書」の発行及び「組込みソフトウェア開発データ白書」の発行決定

ソフトウェア開発データのベンチマーキングへの活用により情報システムの品質・信頼性向上に資することを目指し、「プロジェクト体制とソフトウェアの関係情報」などの新規分析項目を追加して、「ソフトウェア開発データ白書 2014-2015」を発行した。また、組込み分野まで取組みの範囲を拡張し、「組込みソフトウェア開発データ白書」の発行に向けて、10 社約 200 プロジェクトのデータを収集し、分析を開始すると共に、2015 年度の発行を決定した。

2 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）セーフティ設計とセキュリティ設計の見える化を推進

利用者が様々な製品やサービスを組み合わせて使用する「つながる世界」において、利用者が安全・安心につながる製品やサービスを利用するためには、サプライチェーンを構成する事業者が取り組むべき事項として、セーフティとセキュリティ設計が確実に実施されることが重要である。そこで実際に先進的な取組みを行っている企業におけるセーフティ設計、セキュリティ設計の実施状況について明らかにするために調査を実施した。調査を通じて、先進的な企業で使われている分析手法、対策手法なども確認すると共に、システムの設計品質の見える化を行う手法を解説し、セーフティ設計とセキュリティ設計の見える化を推進するためのガイドブックを取りまとめた。

（2）「つながる」システムに向けたソフトウェア品質向上のためのガイドブックを作成

IT システムに対する利用者の期待が、機能の提供だけでなく使用時の高い満足感を求めるよう変化する中、「つながる」システムに関わる多くのステークホルダが持つ様々な期待を、品質要求としていかに漏れなく洗い出し整理するかという課題がある。そこで SEC では、製品・サービスを提供する事業者がより広範囲な視点で品質要求を整理し評価するための対策を「つながる世界のソフトウェア品質ガイド」として取りまとめ、そのダイジェスト版を公開した。

（3）先進的な設計手法・信頼性検証手法・技術などの取組み事例を収集し、適用事例集として取りまとめ

ソフトウェアの高信頼性を確保するためには、上流工程（初期段階）での要件定義や設計が極めて重要であり、ソフトウェア開発の検証・妥当性確認技術や設計・開発手法などの導入が求められる。そこで、SEC では、先進的な取組みを実施している企業・団体・大学から、手法や技術などの導入上の工夫や実際の導入効果などのある実事例を 24 件収集・分析し、「先進的な設計・検証技術の適用事例報告書 2014 年度版」として取りまとめた。さらに、開発現場への成果の導入を促進するために、高信頼化技術適用セミナーを開催して、事例の紹介を積極的に実施した。

3 SEC 成果の国際的情報発信、国際連携

(1) IT プロジェクトベンチマーキング・プロセス評価の SEC 成果に基づく国際規格が発行

IT プロジェクトベンチマーキング・プロセス評価などの SEC 成果の国際標準化活動を推進し、1 件が国際規格の発行手続に入り、もう 1 件は国際規格として発行された。日本企業にとって馴染みの深い手法が国際標準になることで、中小企業などの海外進出や日本と同等品質の海外オフショア開発実現などの一助として、我が国産業の国際競争力向上が期待される。

(2) 海外有力機関との更なる関係強化

これまで連携をしている海外代表的機関の米国 NIST※1、米国 SEI※2、米国 MIT※3、独国 IESE※4、英国 MISRA※5、蘭国 TNO-ESI※6 との関係を更に強化すると共に、英国 RSSB※7 との関係を構築した。

NIST とは 2014 年 12 月に第 5 回定期協議をワシントンで開催し、今回は SEC 成果である組込みソフトウェア開発向けコーディング作法ガイド（ESCR※8）のセキュリティ対応状況について紹介し、NIST が進める CWE※9 の観点での意見交換を実施した。

SEI とは両機関の連携の一環として、「SEC 特別セミナー」（2014 年 7 月開催）に SEI 所長などを講演者として招聘すると共に、2014 年 12 月に SEI を訪問し、意見交換を行い、日米の違いを明らかにすべく両者の保有するプロジェクトデータの比較を行うことで合意した。

MIT には 2014 年 12 月に訪問し、Nancy Leveson 教授が提唱する STAMP（システム理論に基づく事故モデル）の適用方法について意見交換を実施し、2015 年度に開催する「SEC 特別セミナー」の講演者として招聘することを決定した。

IESE とは、IoT（Internet of things）をテーマとした「SEC 特別セミナー」（2015 年 2 月開催）に IESE 所長を講演者として招聘すると共に、IoT や独国が進める戦略的プロジェクトである Industrie4.0 に関する意見交換を実施した。

MISRA には 2015 年 3 月に訪問し、SEC の ESCR［C++ 言語版］と MISRA の C++ 言語版のコーディング規約（プログラミング・ルール）の改訂や普及に関する連携について進め方を協議し、2015 年度に MISRA 関係者を招聘して、「SEC 特別セミナー」を開催することについて調整を実施した。

TNO-ESI には 2015 年 3 月に訪問し、SEC の 2014 年度の取組みの成果であるモデルベースアプローチによる事後 V&V※10 フレームワークなどについて説明し、先方からはモデルベース検査手法や故障個所の診断技法に関する説明を受けるなど、両者の取組みの状況について意見交換を実施した。

RSSB とは「IPA グローバルシンポジウム 2014」（2014 年 10 月開催）にヒューマンファクターの専門家である Huw Gibson 氏を招聘すると共に、ヒューマンファクターを重視した事故／障害の分析手法について意見交換を実施した。

4 SEC 成果の普及展開

(1) セミナーやイベント出展を通じた SEC 成果の普及展開を積極的に推進

業界団体などと連携し、SEC セミナーを計 68 回開催した（東京 52 回、地方 16 回、参加者数 3,236 名）。さらに、上記のセミナーのほか、地域・団体などからの要請に応じた講師派遣についても、計 18 回実施（参加者数 1,506 名）するなど、きめ細かい支援を実施した。

また、ソフトウェア開発技術関連の技術展示会（組込み総合技術展 関西 2014（ETWest2014※11）、組込み総合技術展 2014（ET2014※12）など）に出展し、SEC 成果や取組みの紹介を行うなど、積極的に普及活動を実施した。

さらに、JAXA※13 と共催で、第 12 回クリティカルソフトウェアワークショップ（12thWOCS[2]※14）を開催した（2015 年 1 月）。2014 年度は「Sociotechnical Science and Systems Engineering」のテーマと共に、テーマを実現する重要な技術領域である「信頼性と検証・妥当性確認（Reliability and V&V）」「安全性とセキュリティ（Safety and Security）」「プロセスと計測指標（Process and Metrics）」というサブテーマを掲げ、いかにして信頼性・安全性を確保したソフトウェアシステムを作り上げるかについて様々な講演を実施した。

次頁からは、これらの内容について詳しく紹介する。

【脚注】

※ 1 NIST（National Institute of Standards and Technology）：米国商務省国立標準技術研究所。
※ 2 SEI（Software Engineering Institute）：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。
※ 3 MIT（Massachusetts Institute of Technology）：マサチューセッツ工科大学。
※ 4 IESE（Institute for Experimental Software Engineering）：フラウンホーファー研究機構実験的ソフトウェア工学研究所。
※ 5 MISRA（The Motor Industry Software Reliability Association）：自動車メーカ、部品メーカ、研究者から成る欧州の自動車業界団体。
※ 6 TNO-ESI（Netherlands Organization for Applied Scientific Research-Embedded Systems Innovation）：応用科学研究機構 組込みシステムイノベーション。
※ 7 RSSB（Rail Safety and Standards Board）：英国鉄道安全標準化機構。
※ 8 ESCR（Embedded System development Coding Reference）
※ 9 CWE（Common Weakness Enumeration）：ソフトウェアにおけるセキュリティ上の脆弱性の種類を識別するための共通の基準。
※ 10 V&V（Verification and Validation）：検証と妥当性確認。
※ 11 ETWest2014（Embedded Technology West 2014）組込み総合技術展　関西。
※ 12 ET2014（Embedded Technology 2014）組込み総合技術展。
※ 13 JAXA（Japan Aerospace eXploration Agency）：国立研究開発法人宇宙航空研究開発機構。
※ 14 WOCS[2]（Workshop on Critical Software Systems）：クリティカルソフトウェアワークショップ。

システムグループ

情報処理システムの信頼性向上に向けて

SEC システムグループリーダー
山下　博之

1 ソフトウェア障害事例の収集・分析及び再発防止策

国民生活や社会・経済基盤を支える重要インフラ分野などにおける情報処理システムの信頼性向上のため、システムの障害事例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・分野を越えて幅広く共有し、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取り組みを前年度に引き続き推進した。具体的には、一定の機密保持ルールのもとに重要インフラ分野をはじめとする企業などからの情報提供や有識者からのヒアリングなどにより、28 件（製品･制御システム分野 10 件、IT サービス分野 18 件）の障害事例を収集した。並行して、重要インフラ分野などの有識者・専門家の委員を中心とする委員会 2 種を設置し、これまでの産学官の連携のもとに蓄積されたソフトウェア・エンジニアリングに関する幅広い知見を基礎として、収集した障害事例の分析と対策の検討を行い、それらを産業分野横断で活用可能な普遍的な教訓として一般化・抽象化した。これらの教訓を原因などの観点で分類整理し、「情報処理システム高信頼化教訓集 2014 年度版」として取りまとめ、公開した（図 1）。

また、このような仕組みを幅広く展開するため、障害事例の分析に基づく情報処理システムの高信頼化活動について整理すると共に、上記取り組み内容と成果を各産業分野の団体などに説明した。その結果、業界団体などとの連携により、情報通信、行政、電力の 3 分野で障害事例共有の仕組みが構築された（図 2）。

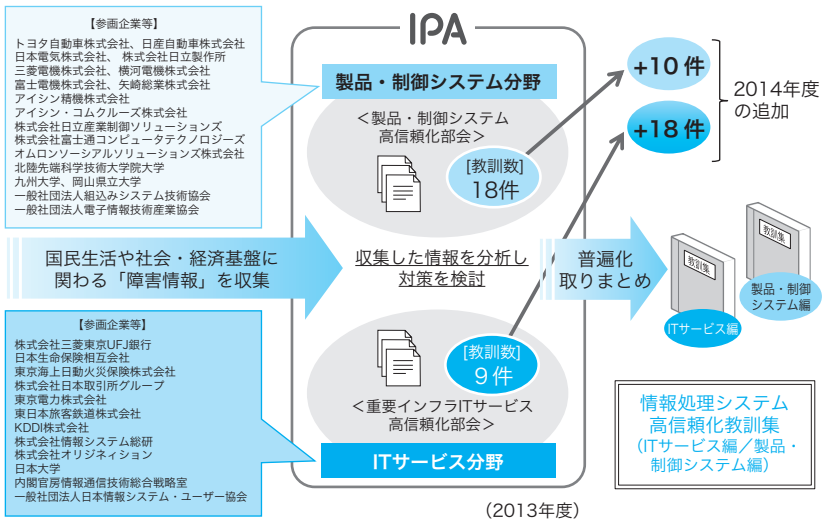


図 1　ソフトウェア障害事例の収集・分析と教訓化の取り組み

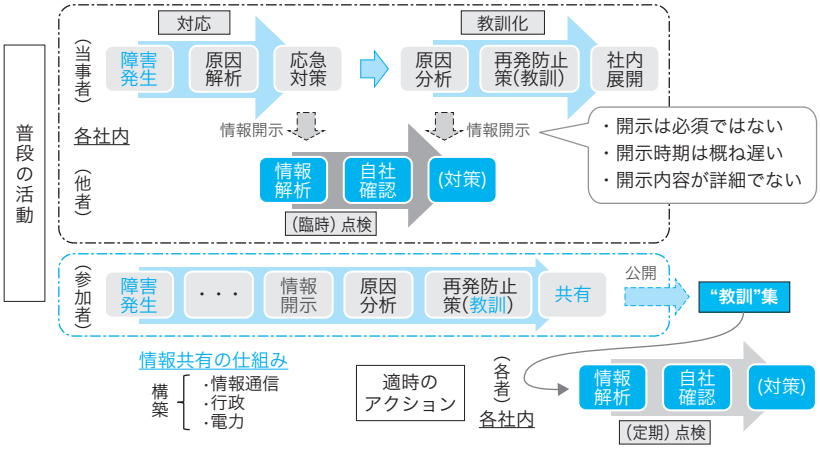


図 2　障害事例の分析に基づく情報処理システムの高信頼化活動と情報共有

2 定量データに基づくシステムの構築及び運用の管理

10 年間にわたるデータ収集・蓄積に基づくソフトウェア開発データ白書の定期的な発行を継続すると共に、組込みソフトウェア開発データ白書の発行に向けた取り組みが進展した。また、システムの運用管理方法に関する現状調査を行い、課題などを明らかにした。

3 システム高信頼化手法等の普及展開

上記取り組みの成果及びこれまでのソフトウェア・エンジニアリング関連の成果について、社会状況に整合させるための改訂作業やイベント出展、セミナーなどによる普及展開活動を実施した。とくに、コーディング作法ガイド（ESCR）の改訂や、前年度に引き続き、SPEAK-IPA 準アセッサ育成コースの開講などを行った。

システムグループ

重要インフラ等システム障害対策（製品・制御システム）

SEC 調査役
三原　幸博

SEC 研究員
松田　充弘

SEC 調査役
石田　茂

SEC 調査役
十山　圭介

SEC 調査役
石井　正悟

交通機関や電気・水道システムなどの機器の制御を行う製品・制御システム（組込みシステム）の障害事例の収集・分析と対策の検討を行い、その結果を普遍化した「教訓」として取りまとめた「情報処理システム高信頼化教訓集（製品・制御システム編）」2014 年度版※1 を公開した。併せて企業における品質文化を醸成するための「製品・制御システム高信頼化のための行動指針」※2、障害発生時の真因分析のための「障害分析手法事例解説書」※3、モデルベースアプローチ、システムズエンジニアリング手法に基づく障害診断のための「大規模・複雑化した組込みシステムのための障害診断手法」※4 を公開した。

1 製品・制御システム分野における高信頼化

近年、機器や製品（以下、製品・制御システム）の機能の大半がコンピュータを利用してソフトウェアで実現されるようになってきている。それらには社会インフラとして重要な役割を担うものも多く、高い信頼性が求められる場合が少なくない。しかし、製品・制御システムは、実現する機能規模が肥大化すると共に異なるシステム同士が複合化する傾向にあり、システム全体として信頼性を確保するための技術面での工夫や運用管理での工夫が求められている。

一方、企業間競争の激化により、差分開発といった短納期の製品開発が主流となり、システム高信頼化のための技術やノウハウが企業内でうまく伝承されていないといった問題も顕在化している。

そこで IPA/SEC は、製品・制御システムのシステム信頼性に関する現状を鑑み、産業界におけるシステム高信頼に関する知見を集積し、将来に向けたシステム信頼性向上に関する技術的な布石を打ち、その結果としてシステム信頼性に関する社会的な認識レベルを上げていくことを目的に、「製品・制御システム高信頼化部会」とその傘下の下記３つの WG を設置し、産学の有識者を交えた議論を進めた（図１）。

(1) 未然防止知識 WG：製品・制御システムの障害を未然に防止するためのノウハウや知見を収集分析し、産業界で共通利用できるよう教訓化する。

(2) 障害事例検証 WG：製品・制御システムにおけるシステム障害に関する事例研究を通して、システム障害発生時の対処法や障害要因分析の手法を整理する。

(3) 障害原因診断 WG：製品・制御システムに障害事象が発生したときに、ソフトウェア面の原因を、モデルベースアプローチ、システムズエンジニアリングに基づき迅速かつ的確に、更に透明性・客観性を確保しつつ指摘できるようにする。事後 V&V（Verification and Validation）として体系化し、人材育成につなげる。

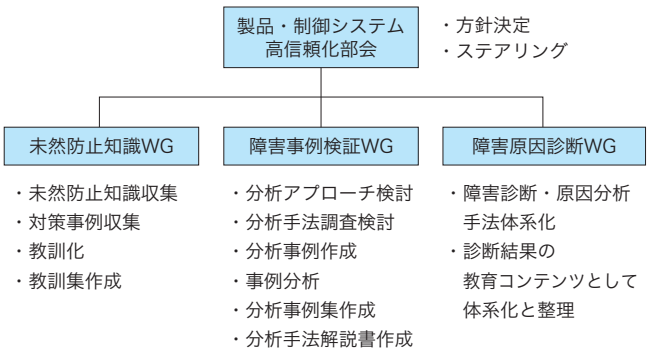


図１　製品・制御システムに関する部会・WG 構成

【脚注】
※１　http://www.ipa.go.jp/sec/reports/20150327_2.html
※２　http://www.ipa.go.jp/sec/reports/20150330.html
※３　http://www.ipa.go.jp/sec/reports/20150327_2.html
※４　http://www.ipa.go.jp/sec/reports/20150331_2.html

2 障害対策事例の収集と教訓集・対策事例集作成

2.1 背景と狙い

2013 年度に引き続き、産業界で実際に活用されているシステムの品質上の問題を未然に防ぐための知識をもとに、製品・制御システムの障害から得られた知見やノウハウを抽象化、一般化することによって、組込みシステム開発企業において幅広く利活用できるための『智恵』として教訓を作成し、前年度に公開した教訓集を改訂した。併せて教訓を実践するための対策の事例集を改訂した。

2.2 教訓の分類

(1) 観点による分類

発生した障害から得られる知見を他製品や産業領域に適用、展開するなどといった、障害を未然防止化するための取り組みはその重要性が認識されてはいても、開発形態やプロセス、技術の違いにより容易でないという現実もある。一方、各事例の障害事象を引き起こした原因には共通する要素も見受けられる。そこで事例を抽象化することにより未然防止の教訓として自社・自部門製品で適用する際のトリガーとなるよう、28 教訓事例の直接原因と真因を観点マップとして抽出・整理した（図２、図３）。また開発現場での活用方法についても参考例として記載した。

図２　直接原因観点マップ

図３　未然防止観点マップ

(2) 教訓と開発工程による分類

開発工程においてどのような対策を施せば、教訓が解決しようとしている問題を未然防止できるのかという観点から解決策を整理した。プロセスモデルは「【改訂版】組込みソフトウェア向け開発プロセスガイド（ESPR Ver.2.0）」のモデルを採用した。また、直接開発にかかわるシステム・エンジニアリング・プロセス（SYP）やソフトウェア・エンジニアリング・プロセス（SWP）だけでなく、サポート・プロセス（SUP）も対象とし、ESPR では定義されていない教育に関する工程も工程別未然防止知識の一部として採用した。また、組込みシステム開発において多く見られる差分開発特有の未然防止知識についても、切り出して記載した。

教訓と対策が必要な工程との対応例を表 1 に示す。

3 障害分析手法事例解説書

開発中や出荷後に発生した障害をきちんと分析して根本的な原因を特定し、同種の障害が再発しないように開発の進め方を改善していくことが信頼性を高めるために重要となる。障害が発生した後に行われる作業の種類や内容について、表にまとめると共に、作業の流れをフローで示した。また、分析に必要となる情報の種類についても述べている。開発現場で役立つ知識の提供を目指して経験豊富な技術者が普段どのように障害を分析しているのか、事例に沿った形で解説した。具体性を持たせることで分析作業を行う際の思考の過程や分析手法の使われ方を読み取れるように努めた。なお、より深い分析を行うため、障害事例として本書内で公開した情報では、不明情報を創作して補って具体的に障害分析を行った。分析に際して HOW だけでなく WHY を記載することにより理解を容易にしている。

また、分析手法の解説とは別に、分析結果を再発防止につなげる取り組みの事例も紹介している。

4 製品・制御システム高信頼化のための行動指針

重要インフラを支える製品・制御システムにおいて求められる信頼性を発揮するためにシステムのライフサイクル（企画・設計・開発・保守・運用）全体を通して経営層及び開発責任者、品質責任者が遵守すべき事項を行動指針として取りまとめた。

主な特徴は以下の通り。

・ 重要インフラなどの製品・制御システムの開発企業の

表1　教訓一覧と対策が必要な工程との対応例

教訓番号	教訓タイトル	システム要求定義	システムアーキテクチャ設計	ソフトウェアアーキテクチャ設計	ソフトウェアアーキテクチャ設計（変更設計）	実装（コーディング）	レビュー	システムテスト	教育	プロジェクトマネジメント	運用
1	複雑な条件式のロジック変更を行う場合は、デシジョンテーブルなどによる検証が有効である			○	○						
2	条件が整理されていない状態で、トータルの条件数が 100 を超えるような機能、または 10 個以上の条件を有する機能を修正する場合、関連する条件を全て洗い出して整理し不整合がないことを確認する			○	○						
3	複数機能モジュールを統合する場合、統合前の条件数の総和と統合後の条件数を比較し差がある場合は、条件の抜けがないか確認する				○			○			
4	変数値域が広く、組み合わせバリエーションが非常に多くなる場合には、値域を適切な大きさに分割した上で境界値テストを実施する				○						
5	内蔵電池を使用する場合には、深放電時の起動シーケンスを考慮すること		○	○			○	○	○		
6	フラッシュメモリを使用する場合には、書き込み寿命回数を考慮すること	○	○							○	○
7	消費電力の多い機能を追加する場合には、一時的な電圧降下による影響（リセット、フリーズなど）や電源の種類、電池の場合は残量を考慮すること		○								
8	想定可能な例外を形式的に漏れなく分析する	○	○								
9	システムを二重化する場合は、同期すべきデータ領域を適切に設定する			○							
10	制御対象のハードウェアが同一でも、運用条件が変わるときは、ハードウェア仕様を再確認する		○		○		○		○		
11	プロセス間、スレッド間でデータを共有（引き渡し）する場合は、排他・同期処理が正しく行われているか、あるいはデッドロックが発生していないかどうか注意する			○		○			○		
12	歩留りのある製品の良品／不良品を検査する装置では、全てが良品あるいは、不良品との検査結果は異常と判断すべきである	○									○
13	既存ソフトウェアの性能改善を実施する際には、アイドリングタイムの発生、処理の同期ずれの発生等と影響を確認する			○	○			○	○	○	
14	・大量のデータを通信経由で扱う場合、一連の処理の流れの中にボトルネックを作りこまないように注意する ・時間帯による負荷変動について考慮する	○	○	○			○				
15	納入したあと、お客様が運用するような業務システムでは、業務シーケンス中のあらゆる異常操作（リセット、電源断、放置も含め）、への対応を考える				○			○			
16	障害解析時の保守メンテ用ログ処理であっても、仕様書を作成し、影響評価を実施すること			○							
17	判断処理は、必要条件だけでなく、制限すべき条件も漏れなく抽出する				○						
18	ログファイルの断片化に注意する			○							
19	人による変更作業ではミスが起きることを前提に、ツール活用などで不具合の作り込みや流出の防止に心がける	○			○			○			
20	信頼性向上施策を採る場合は、故障発生確率と影響の定量評価を行い、対策は確実に実装する		○	○			○			○	
21	高い信頼性対策が求められるシステムでは重大な影響を及ぼす事象の想定と復旧手順を十分に検討する		○								○
22	処理時間がクリティカルなシステムではツールを活用し、変数やその取りうる状態数とそれぞれの状況における動作処理に最大バラツキを意識し余裕を把握し設計する			○	○		○	○	○		
23	開発を伴わない保守案件でも，システム構成変更が発生する場合は，手順など作業内容の妥当性を確認できるようなプロセスを経る						○	○		○	○
24	物理量（時間、重量など）を扱う場合は単位、桁数を確認する		○			○		○			
25	顧客が要求していることの目的と背景に遡って、その意図を確認することが、要求仕様のあいまいさ排除に役立つ	○					○				
26	遠隔地など物理的に離れた装置をネットワーク接続して稼働させるシステムでは、故障などの状態検知やメンテナンスも容易ではないため、システム的視点での状態把握を行う	○	○					○			
27	マルチベンダーシステムでは仕様に外れた想定外事象が発生することを前提とした自己防衛策を採る	○		○				○			
28	データベースなど COTS 製品のバージョン、動作仕様の相違などの情報が関係者にタイムリーに参照できるようにする							○	○	○	○

有識者・専門家と組込み系ソフトウェア工学の学識者による「製品・制御システム高信頼化部会」にて検討・整理し、実際に産業界で活動している方の声を反映。

- IT サービスと異なる製品・制御システムに特有の状況に鑑みた技術的及び事業的特性に配慮。
- 経営層及び開発責任者、品質責任者が取り組むべきエンジニアリング・コンピテンシーを記述。

本指針が製品・制御システムの関係者に積極的に活用され、信頼性にかかわるコンピテンシーの維持向上が図られ、社会経済活動全体の信頼性向上、ひいては安心・安全な社会生活の実現に資することを期待している。

5 障害原因診断手法

5.1 背景と狙い

複数のシステムが連携することで、システム全体が大規模・複雑になり、システムを管理・操作する人とシステムの関係も複雑になる傾向がある。そのため、大規模・複雑なシステムで、人とシステムまたは複数システム間に起因する複合的な要因による障害が発生すると、原因を見つけ出すことは困難であり、その影響も広範囲かつ深刻になる。

また、システムの制御を担う要素が、ハードウェアから、制御ソフトなどソフトウェアを主体としたものに替わって

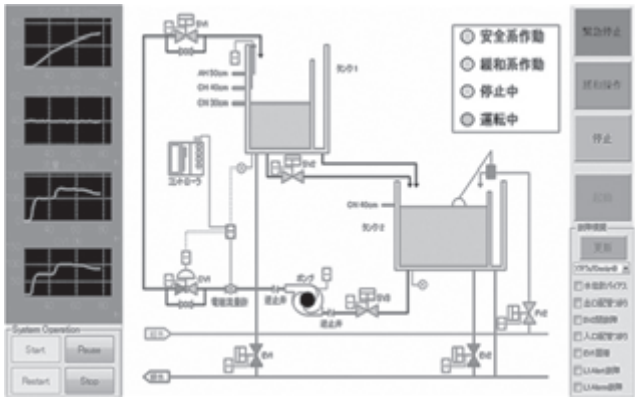


図４　化学プラントシミュレータの表示画面

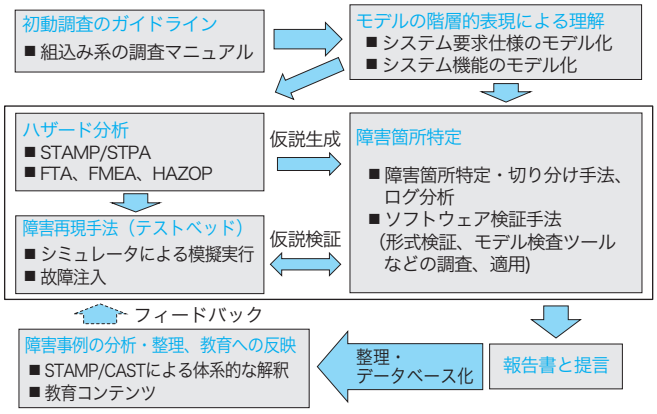


図５　事後 V&V 体系

きており、障害発生時のシステムの診断や原因の分析と対策も、従来の物理的要因中心の視点から、ソフトウェアやシステム中心のシステムズエンジニアリングに基づいた視点に変える必要がある。

そこで、従来の方法よりも容易に障害診断を可能にするため①障害発生後の初動調査と情報収集の手法、②人為的な要因も含め、システムとソフトウェアのどこに原因があるかを見極める障害診断手法、③障害原因となるソフトウェア制御ロジックの不具合を見極める形式検証手法やテスト手法を組み合わせて事後 V&V（Verification and Validation）として体系化し（図 5）、「大規模・複雑化した組込みシステムのための障害診断手法～モデルベースアプローチによる事後 V&V の提案～」として公開した。

5.2 事後 V&V の特徴

モデルベース手法をベースに、対象システムを故障原因仮説に沿って動作をシミュレートすることにより障害を再現させ、診断を行う。

これにより、以下の点が期待できる。

- システム全体の振る舞いを確認しながら分析できるため、原因個所の特定がしやすく、分析に要する作業時間の短縮が期待できる。また、類似の原因による障害の再発防止にもつながる。
- 実際のシステムを動作させることなくシミュレータ上で仮想的に動作検証ができるため、実際のシステムを一切毀損することなく障害を再現でき、通常は試験が困難な障害まで確認できる。
- システム実装前の動作検証にも使用できるため、障害の未然防止にもつながる。
- 人（操作員）の動作とシステム機能間の不整合などを含む複合要因に対する障害診断が可能。
- 実際のシステムをシミュレーションできるため、操作員向けのトレーニングや、非常事態を想定した訓練などへの活用が可能。

5.3 事後 V&V の有効性の確認

ソフトウェアの欠陥が関連している過去の障害事例を調査し、事後に検証を行うためのテストベッドの基本設計と、その実現例とすべくモデルベース開発ツールを用いて、化学プラントシミュレータを開発して検証を行った（図 4）。

その結果、人間の操作と制御ロジックの間に矛盾があるようなシステム障害の模擬や、ソフトウェアロジックの形式検証が可能であることを確認し、障害診断手法の有用性を確認した。

5.4 今後の取り組み

米国 MIT※5 の Nancy Leveson 教授が提唱し、欧米を中心に活用されている STAMP※6 のような新しい安全解析手法を取り入れるなど、実用性の向上を進める。更に、

- 障害が発生した際に、第三者として診断活動を行う役割の確立
- 障害の再発防止
- 透明性・客観性の確保
- 診断作業の迅速化を図るため、障害原因の診断結果を抽象化
- 教育コンテンツとして体系的に整理・蓄積

を進め新たなフレームワークとして普及展開を図っていく。

6 今後の課題

「教訓集」を始め「製品・制御システム高信頼化のための行動指針」、「障害分析手法事例解説書」、「大規模・複雑化した組込みシステムのための障害診断手法」について、現場での適用を促進するため、セミナーなどの開催と、活用を意識した質・量両面からのブラッシュアップを進めていく。また、現場からの評価の収集にも努め、今後の活動にフィードバックしていく。

今後は、行動指針に示された考え方にのっとり、各企業が自ら高信頼なものづくりを継続的に取り組んでいくための教材作成や教育の普及に向けた取り組みを推進していく。

【脚注】

※5　MIT（Massachusetts Institute of Technology）：マサチューセッツ工科大学

※6　STAMP（Systems-Theoretic Accident Model and Process）：システム理論に基づく事故モデル

システムグループ

重要インフラ等システム障害対策（IT サービス）

SEC 研究員
加藤　均

SEC 研究員
目黒　達生

SEC 主任
八嶋　俊介

SEC 調査役
三縄　俊信

SEC システムグループリーダー
山下　博之

前年度に引き続き、一定の機密保持ルールのもとに重要インフラ分野などの企業からの情報提供や有識者からのヒアリングなどにより障害事例を収集し分析と対策の検討を行った。その中から産業分野横断で活用可能な普遍的な教訓を 18 件導出し、2013 年度分と合わせた 27 件の教訓を分類整理した上で、「情報処理システム高信頼化教訓集（IT サービス編）」2014 年度版※1 として公開した。また、情報処理システムの障害事例を社会で共有する仕組みの構築に向けた普及活動を行い、３つの産業分野で情報共有の仕組みを構築し運用を開始した。

1 システム障害事例の収集・分析及び対策の検討

情報処理システムは、銀行や証券などの金融サービス、住民情報サービス、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に深く浸透し、ひとたび障害が発生するとその影響は非常に大きくなり、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、IT サービスの一層の信頼性向上が求められる。

IPA/SEC の調査結果では、報道された IT サービス障害の発生件数は、図 1 に示すように、2009 年から 2014 年にかけて増加傾向にあった。とくに 2014 年度は、消費税率 8 % 引き上げに伴う障害が多く発生した。また、クラウドサービス型のシステムにおいても障害が発生した。

従来、情報処理システムの障害に対する原因分析と再発防止対策の実施は、多くの場合、当事者においてのみ行われ、その情報は公開されて来なかった。そのため、別のシステムにおいて、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。

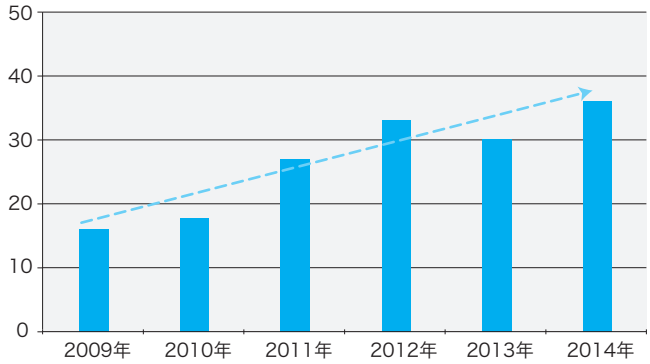


図 1　報道された IT サービス障害の発生件数の推移

情報処理システムの構築・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、一人や一企業がカバーできる範囲には限界がある。そして、その複雑性・多様性は今後ますます拡大していくことは明らかである。従って、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向上面での課題を解決するために、より多くの人たち・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いる。

そこで、システムの障害事例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・分野を越えて幅広く共有し、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。

2014 年度も継続して重要インフラ IT サービス高信頼化部会※2 の活動を通じ、障害事例を収集し、障害原因の分析

表 1　2014 年度に導出した教訓の分野別件数

産業等分野	教訓数
情報通信分野	3 件
金融分野	6 件
交通分野	3 件
行政分野	5 件
その他	1 件
計	18 件

【脚注】

※ 1　http://www.ipa.go.jp/sec/reports/20150327_1.html

※ 2　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 CIO クラスを中心とする有識者・専門家で構成する委員会

表２　教訓一覧（IT サービス編）
※太枠は 2014 年度版追加分

	教訓番号	教訓タイトル
ガバナンス / マネジメント領域		
1	G1	システム開発を情シス部門だけの仕事にせず、各事業部門が自分のこととして捉える「態勢」をつくることが大切
2	G2	発注者は要件定義に責任を持ってシステム構築にかかわるべし
3	G3	運用部門は上流工程（企画・要件定義）から開発部門と連携して進めるべし
4	G4	運用者は、少しでも気になった事象は放置せず共有し、とことん追求すべし
5	G5	サービスの拡大期には業務の処理量について特に入念な予測を実施すべし
6	G6	作業ミスとルール逸脱は、個人の問題でなく、組織の問題！
7	G7	クラウド事業者と利用者が連携した統制がとれたトラブル対応体制を整備すべし
8	G8	共同利用システムでは、非常時対応を含めて利用者間の情報共有を図ること
9	G9	システム利用不可時の手作業による代替業務マニュアルを作成し定期的な訓練を行うべし
技術領域		
10	T1	サービスの継続を優先するシステムにおいては、疑わしき構成要素を積極的にシステムから切り離せ（“フェールソフト”の考え方）
11	T2	蟻の目だけでなく、システム全体を俯瞰する鳥の目で総合的な対策を行うべし
12	T3	現場をよく知り、現場の知識を集約し、現場の動きをシミュレートできるようにすべし
13	T4	システム全体に影響する変化点を明確にし、その管理ルールを策定せよ
14	T5	サービスの視点で、「変更管理」の仕組み作りと「品質管理責任」の明確化を！
15	T6	テスト環境と本番環境の差異を体系的に整理し、障害のリスク対策を練る
16	T7	バックアップ切替えが失敗する場合を考慮すべし
17	T8	仮想サーバになってもリソース管理、性能監視は運用要件の要である
18	T9	検証は万全？それでもシステム障害は起こる。回避策を準備しておくこと
19	T10	メッシュ構成の範囲は、可用性の確保と、障害の波及リスクのバランスを勘案して決定する
20	T11	サイレント障害を検知するには、適切なサービス監視が重要
21	T12	新製品は、旧製品と同一仕様と言われても、必ず差異を確認！
22	T13	利用者の観点に立った、業務シナリオに則したレビュー、テストが重要
23	T14	Web ページ更新時には、応答速度の変化など、性能面のチェックも忘れずに
24	T15	緊急時こそ、データの一貫性を確保するよう注意すべし
25	T16	システム構成機器の修正パッチ情報の収集は頻繁に行い、緊急性に応じて計画的に対応すべし
26	T17	長時間連続運転による不安定動作発生の回避には定期的な再起動も有効！
27	T18	新たなサブシステムと老朽化した既存システムとを連携する場合は両者の仕様整合性を十分確認すべし

を行い普遍化した上で 18 件の教訓を導出し（表１）、2013 年度に取りまとめた教訓９件に追加して、計 27 件の教訓を収録した「情報処理システム高信頼化教訓集（IT サービス編）」2014 年度版を公開した（ガバナンス / マネジメント領域の教訓 “Gn”、技術領域の教訓 “Tn”）（表２）。

障害事例は幾つかの複合要因を包含しており、一つの障害事例について部会委員による専門的見地による分析を実施した結果、５件の教訓を導出した例もある（図２）。

導出した教訓について、ITIL[※3] をベースとした国際規格 ISO20000[※4] によるサービスマネジメント分類（図３）との対応付けを実施した（表３）。

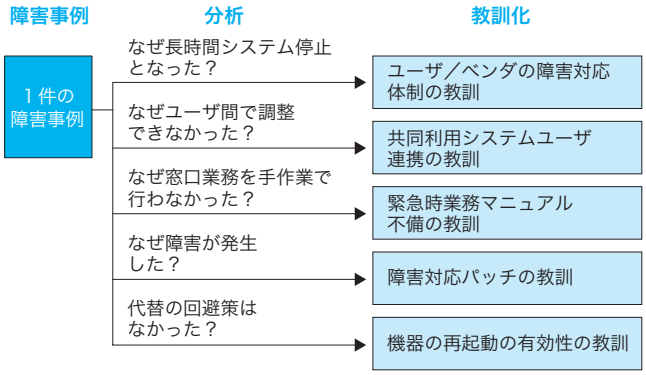
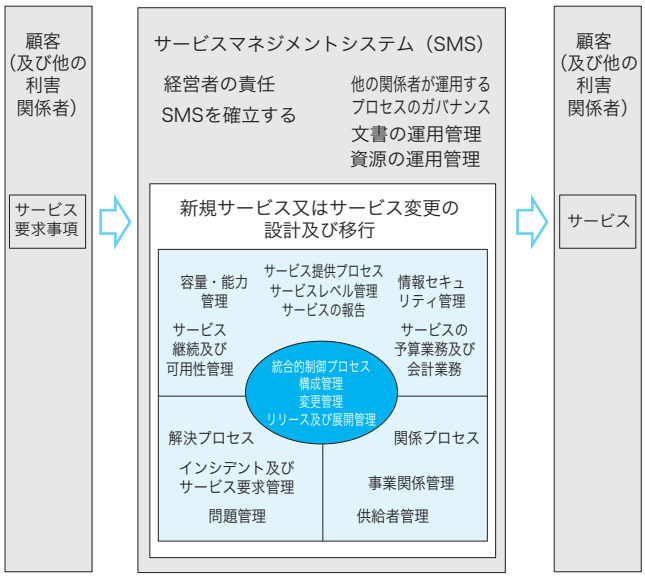


図２　一障害事例から複数の教訓導出の例



（JIS Q 20000-1:2012 より引用）

図３　ISO20000（JIS Q 20000-1）サービス管理の全体像

【脚注】

※３　ITIL：Information Technology Infrastructure Library、IT サービスマネジメントのベストプラクティス集で、IT サービスを提供するためのガイドライン。

※４　ISO 20000：IT サービスを提供している組織が、サービスの内容やリスクを明確にすることで、IT サービスの継続的な管理、高い効率性、継続的改善を実現するための国際規格。

表３から、統合的制御プロセスの構成管理、変更管理、サービス継続・可用性管理のプロセスに問題が多いことが分かる。

障害分析手法は、2013 年度版に掲載していた STAMP[※5] の記載内容を最新の内容に改訂した。また、対策手法は新たな教訓の追加に伴い、2013 年度の９件に 12 件を追加（表４）した。

2 システム障害教訓の普及活動

SEC セミナー「事例から学ぶ IT サービスの高信頼化へのアプローチ」を 2014 年度は２回開催した。セミナーではシステム障害事例に基づいた障害分析、対策作成から教訓の導出までの一連の演習を参加者全員でのワークショップ形式で実施した。参加者のアンケートでは「今回のようなケースに基づくグループワークはとても有意義であった（50 代、人材育成コンサルタント）」などの声があった（図４）。

また、各業界団体（21 団体）に「情報処理システム高信頼化教訓集（IT サービス編）」を紹介し、活用について説明と意見交換、必要に応じて講演会を実施した。

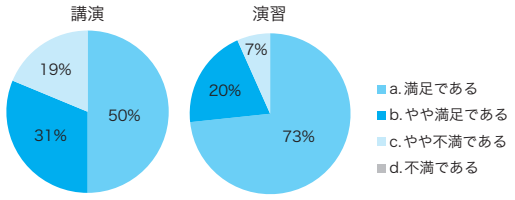
教訓集について「IPA が障害事例やケーススタディを出していくのは重要と考える」、「ベンダのユーザ会などで情報共有はあるが、ベンダをまたがって事例共有することはより有意義である」、「情報処理システム障害の教訓をまとめることは良い取り組みと考えており IT サービスの障害事例の提供にも協力したい」、「教訓の横展開は障害を減らすために重要であり、自社内でも必要と考えるが汎用化の方法や共有の方法など整理できていない。IPA から教えて欲しい」などの意見が出た。更に「クラウド化が増えているが、これに関する事例や留意点が欲しい」、「パッケージ製品を利用して発生した障害事例があると良い」など、今後の拡張に期待する要望もあった。

表３　各教訓と IT サービスマネジメントの対応

教訓 ID	JIS　Q 20000-1：2012 より（●主な問題個所、△関連する問題個所）												
	5. 新規またはサービス変更の設計及び移行	6. サービス提供プロセス					7. 関係プロセス		8. 解決プロセス		9. 統合的制御プロセス		
		サービスレベル管理	サービス継続・可用性管理	サービス報告	容量・能力管理	情報セキュリティ管理	事業関係管理	供給者管理	インシデント管理	問題管理	構成管理	変更管理	リリース管理
G1	△						●						
G2	●							△					
G3	●	△											
G4				△				△	●	△			
G5					●		△						
G6											△	●	△
G7				△			△	△	●				
G8							●				△		
G9			△				●						
T1			●								△		
T2					△				△		●		
T3	●	△								△		△	
T4			●							△	△	△	
T5	△											●	
T6											●	△	
T7			●		△								
T8					●				△		△		
T9			●							△	△		
T10			△								●		
T11		●			△								
T12								△			△	●	△
T13	●	△										△	
T14					△							●	△
T15											△	●	
T16						△		△		●	△		
T17			△								●		
T18	●										△		

表４　新たに追加した対策手法

追加した対策手法
共同センター利用におけるユーザ企業の連携、合意形成
クラウドセンターと利用企業の連携、合意形成
障害管理の取り組み
プロセス改善
ヒューマンファクターズ
製品に関するトレーサビリティ　ISO9001
テスト網羅性の高度化技法
仮想化技術
レビュー手法
サイレント障害対策
パッチ管理技法
高回復力システム基盤導入



※ 2015 年３月 20 日実施分のセミナーアンケート集計結果

図４　SEC セミナーの満足度アンケート結果の例

【脚注】

※５　STAMP（Systems-Theoretic Accident Model and Process）：システム理論に基づく事故モデル

なお、各業界団体への IPA の取り組みの説明会の後にアンケートを実施した（電気事業連合会の会員企業（回答９企業）、地方共同法人 地方公共団体情報システム機構（J-LIS）から推薦された地方公共団体（回答８団体）、神奈川県市町村自治体（回答 30 件））。

結果によれば、障害事例に基づく教訓共有の取り組みについて、「関心がある」、「成果が適用できる」との回答が高い割合を示した（図５）。

3 システム障害事例共有の仕組み構築

各業界団体にシステム障害事例の共有の仕組み構築を働きかけ、３つの業界団体において仕組みを構築し運営を開始した。

（情報通信分野）

ITA（Information Technology Alliance：情報サービス団体（加盟 18 社））内の９社による「障害再発防止策研究会」が 2014 年に発足し、システム障害事例の共有の仕組みを同団体内に構築し活動を開始した。IPA 職員が同研究会に参加すると共に、IPA の重要インフラ IT サービス高信頼化部会においてその活動状況を紹介していただき、活動の成果は ITA の Web サイト上で「ITA 情報処理システム障害事例集」として公開され IPA のウェブサイトから URL リンクによる連携を実施した。

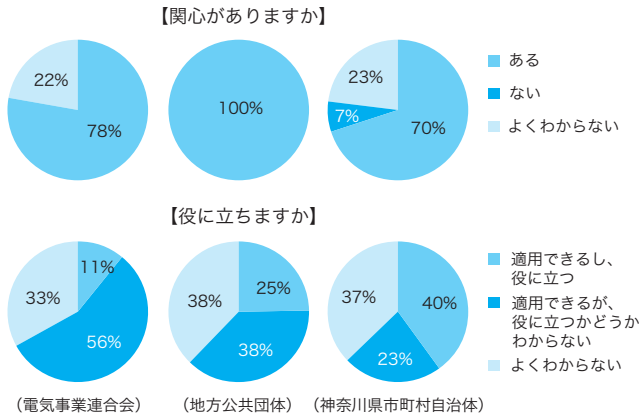


図５　IPA の取り組みへの関心度に関するアンケート結果例

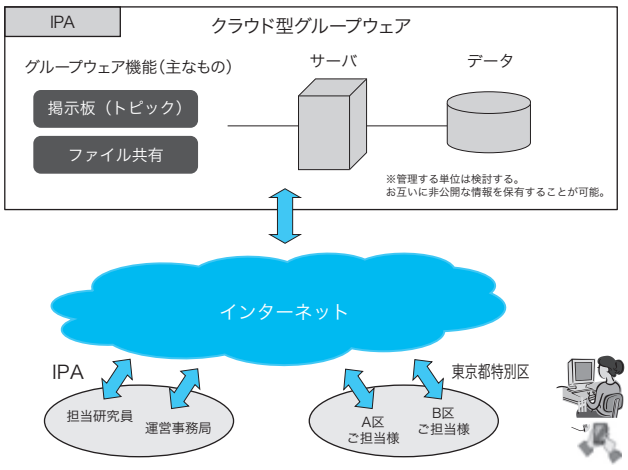


図６　仮想的情報共有グループのイメージ

（行政分野）

東京都特別区電子計算主管課長会にて障害事例共有の意義に賛同いただき、情報共有を行うことが決議された。電子掲示板を使った仮想的な情報共有グループ（図６）を IPA 内に設置し、各区及び IPA 間での情報共有などを行う試行運用を開始した。

（電力分野）

電気事業連合会の協力のもと、情報共有の取り組みに賛同する電力関連９団体・企業を中心にメーリングリストを使用した情報共有を行うこととし、その運用を開始した。IPA もこの情報共有に参加する。

4 今後の予定

システム障害事例を収集してその普遍化を行い教訓として整理する活動は継続し、教訓集の内容の更なる充実を図っていきたい。

また、社会インフラ情報処理システムの一層の信頼性向上を目指し、2014 年度にシステム障害事例の共有の仕組みを構築した３つの産業分野の効果的な運営を支援すると共に、新たな産業分野にも仕組みの構築を働きかけ、自律的な活動を促しつつ、システム障害事例共有の裾野を拡大していきたい。

このために、新たに「障害事例教訓集の活用ガイド（仮称）」や「障害事例から学ぶ情報処理システムの信頼性向上ガイド（仮称）」などを作成する予定である（図７）。

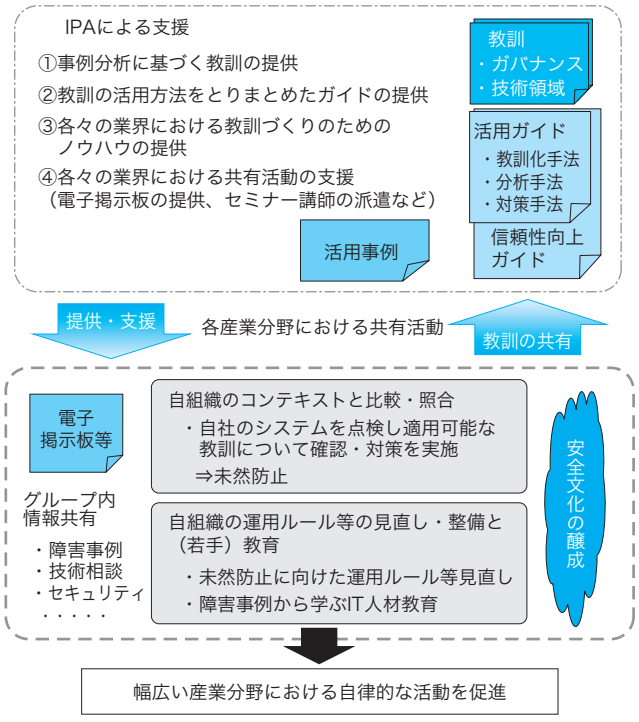


図７　情報共有活動の仕組み

システムグループ

定量的管理による信頼性・生産性向上

SEC 研究員
佐伯　正夫

SEC 研究員
塚元　郁児

SEC 調査役
三縄　俊信

SEC 専門委員
森下　哲成

SEC 研究員
松田　充弘

SEC 調査役
三原　幸博

SEC システムグループリーダー
山下　博之

ソフトウェア開発データの分析に基づくソフトウェアの信頼性・生産性向上を目的に、「ソフトウェア開発データ白書」を発行すると共に、メトリクス分析に関する研究などへの蓄積データの活用拡大などを目指した取り組みを実施した。また、組込みソフトウェア開発データ白書の作成に向けたデータの収集及び試行分析を行った。

1 ソフトウェア開発データ白書の定期的発行

IPA/SEC が蓄積するソフトウェア開発データに、2013、2014 年度にデータ提供企業 23 社から収集した 452 プロジェクト分のデータを追加した計 3,541 プロジェクトのデータを対象に、信頼性や生産性等に関する各種分析を行った「ソフトウェア開発データ白書 2014-2015」を 2014 年 10 月 1 日に発行した※1（図 1）。

図 1　ソフトウェア開発データ白書 2014-2015

（1）主な分析結果

「ソフトウェア開発データ白書 2014-2015」から見られた主な傾向は次の通りである：

1．顧客の要求レベルが高い方が、信頼性は高いが、生産性は低い傾向が見られる（図 2）。
2．プロジェクト体制が以下の各条件を満たす場合、それぞれにおけるソフトウェアの信頼性は高くなる傾向にある。
   - ソフトウェアの品質保証に関する専門部署・専門スタッフを設置していること
   - ソフトウェアのテスト体制として、要員数やスキルが十分であること
   - 定量的な出荷基準が設けられたプロジェクトであること

（2）グラフデータの公開

「ソフトウェア開発データ白書 2014-2015」に掲載されているグラフを利用者側で Excel などを活用し、自由に加工することにより、経営層やユーザへの訴求力のあるプレゼンテーション資料の作成を可能とするため、グラフデータのダウンロードサービスを開始した（2014 年 12 月）※2。

（3）新たな分析の試行結果

蓄積データを対象に、新たな観点での追加分析を試みた。

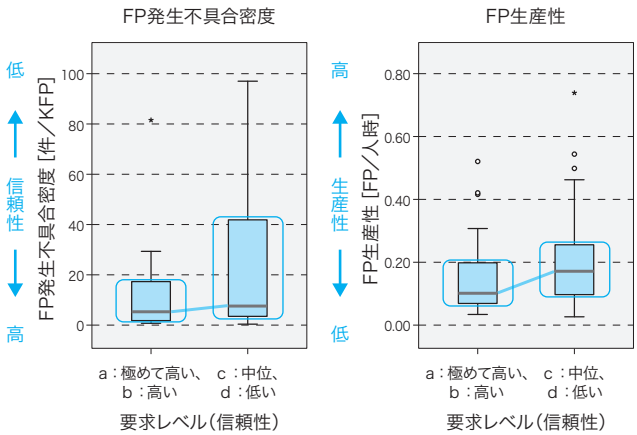


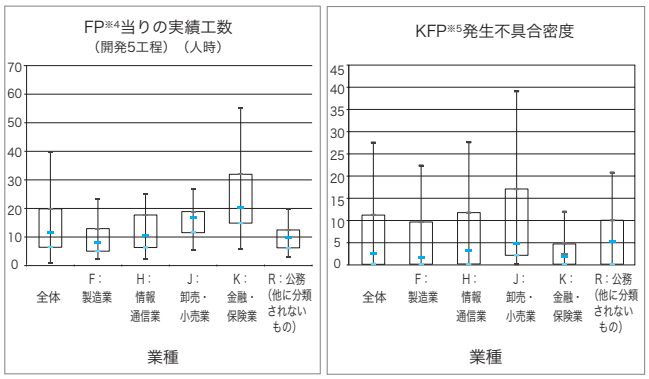
図 2　顧客要求レベルと生産性・信頼性との関係

【脚注】
※ 1　http://www.ipa.go.jp/about/press/20141008.html
※ 2　http://www.ipa.go.jp/sec/reports/20141226.html

追加分析から見えた下記の傾向を SEC セミナーで紹介した。

- 金融保険業は、他業種に比べ、信頼性は高いが、生産性は低い傾向。その要因としては、顧客の信頼性要求レベルの高さや各工程での設計書密度、設計書レビュー、テストの密度が高いことなど（注 SLOC 規模※3、改良開発では、同様の傾向は見られない）（図３）。
- 経年変化では、生産性はあまり変化が見られないが、FP 規模で改良開発のケースでは、信頼性は向上している傾向（図４）。

業種ごとの生産性・信頼性の比較結果から読み取れる傾向：



業種ごとの生産性比較例
（FP規模・新規開発における
FPあたりの実績工数）

業種ごとの信頼性比較例
（FP規模・新規開発における
発生不具合密度）

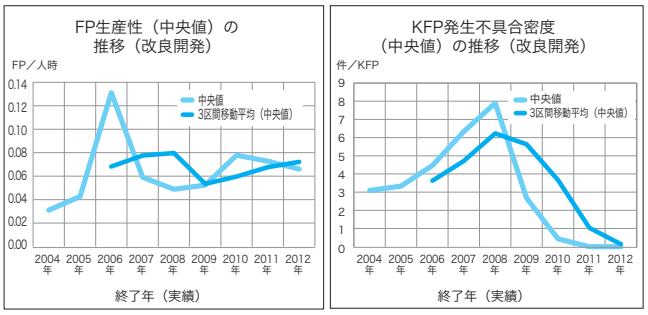
（注）他の条件（SLOC規模、改良開発）では顕著な傾向が見られない結果もある。

図３　生産性・信頼性の業種間比較

生産性・信頼性の経年変化分析結果から読み取れる傾向：

・生産性は、ここ6,7年間あまり変化がない

・信頼性は、ここ5年間向上している



生産性の経年変化例
（FP規模・改良開発における
生産性の推移）

信頼性の経年変化例
（FP規模・改良開発における
発生不具合密度の推移）

（注）他の条件（SLOC規模、新規開発）では異なる傾向がみられる
／顕著な傾向が見られない結果もある。

図４　生産性・信頼性の経年変化

2 定量的管理の推進

信頼性向上のための定量データ分析（メトリクス分析）に関する方策の検討を目的とした高信頼性定量化部会（委員 16 名）と具体的検討作業を目的とした信頼性メトリクス WG（委員 10 名）及び IT サービス定量データ分析 WG（委員 13 名）の活動を行った（図５）。両 WG の活動を通じて、新しい有用なメトリクス分析手法や事例を提供すると共に、メトリクス分析によって得られた信頼性向上のための新たな知見の発信を行った。

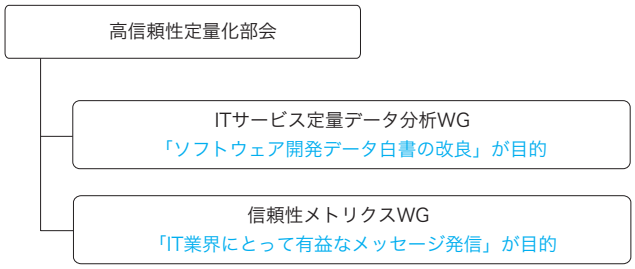


図５　高信頼性定量化部会の構成

2.1 ソフトウェア開発データ白書の改良案の検討

ソフトウェア開発データ白書の改良を目的とし、IT サービス定量データ分析 WG にて、次の内容を検討し、「ソフトウェア開発データ白書 2016-2017」の目次案を作成した。

◇　分かりやすさ・読みやすさの向上
◇　データ及び白書の信頼性向上（収集停止項目の検討）
◇　分析項目の追加／改良

2015 年度は、この案を参考にして「ソフトウェア開発データ白書 2016-2017」の原稿作成に取り組む予定である。

2.2 新しいメトリクス分析手法の検討

新しいメトリクス分析手法及び事例を提供すると共に、信頼性向上のための知見を発信することを目的とし、信頼性メトリクス WG にて、従来経験的に知られていた次の二つの命題を、「横断的分析アプローチ」という新たな分析手法を用いてデータで実証するための分析を行い、その結果を報告書として公開した※6。

命題１：信頼性に影響する開発プロセス要因についての分析
「開発の早い段階から品質をコントロールすれば信頼性が良くなる」

命題２：システムリスクと開発工数の関係分析
「高度な IT システム（システムリスクの高いソフトウェア）の開発には、それ相応の品質保証（レビュー、テストなど）の工数が必要である」

【脚注】

※３　SLOC 規模は、新規にまたは改造してコーディングした行数（Source Lines of Code）。派生や改良開発の場合で母体を含む全体のプログラム行数を SLOC 規模（母体含む）としている。
※４　FP（ファンクションポイント）とは、機能の数を基にソフトウェアの規模を算出したもの。
※５　KFP とは、1,000FP を表す。
※６　http://www.ipa.go.jp/sec/reports/20150416_1.html

本分析により、これらの命題を支持する次のような結果を得た。

- 命題１に対しては、稼働時の信頼性が良いという結果をもたらすプロジェクトの方がおおむね、上流での不具合摘出比率（設計レビューでの不具合摘出数÷（設計レビューでの不具合摘出数＋テストでの不具合摘出数））が高い傾向と、下流での（テスト時の）不具合検出密度（不具合検出数÷開発規模）が低い傾向を確認できた（図６）。
- 命題２に対しては、システムリスクが高いプロジェクトの方が、規模当りのテスト工数（及び部分的に規模当りの総開発工数）が大きくなる傾向を確認できた。

品質に関する分析結果サマリ（改良開発）

★良群、否群の歪度の絶対値がどちらも２未満の場合には実数の検定結果を採用し、それ以外の場合には対数の検定結果を採用する。

◇良群の方が指標値が大きい　↑P<0.1　↗P<0.2　→P<0.5　ー P>=0.5
◇良群の方が指標値が小さい　↓P<0.1　↘P<0.2　→P<0.5　ー P>=0.5

P<0.1:	Welch	t検定のP値が10%以下	10%有意
P<0.2:	Welch	t検定のP値が10%より大きくて20%以下	20%有意
P<0.5:	Welch	t検定のP値が20%より大きくて50%以下	有意ではない
ー　:	Welch	t検定のP値が50%以上	有意ではない

ドメイン等		開発規模	①（上流）レビュー指摘数／規模	②（上流）レビュー工数／規模	③上流工程での欠陥摘出比率	④（下流）不具合検出数／規模	⑤（下流）テスト項目数／規模	⑥（下流）不具合検出数／テスト項目数	⑦（下流）テスト工数／規模	⑧総欠陥数／規模
A社 改良開発	傾向	↓	ー	→	↑	↓	ー	↓	ー	ー
	良群件数	19	19	19	19	19	19	19	19	19
	否群件数	22	22	22	22	22	22	22	22	22
B社 改良開発（外れ値除外）	傾向	↓	↗	ー	↑	↓	↑	↓	↑	ー
	良群件数	221	224	198	233	229	222	225	217	221
	否群件数	67	66	42	71	73	66	68	66	68
B社 改良開発（異常値除外）	傾向	↘	→	ー	↑	ー	↑	↘	↑	→
	良群件数	241	240	219	233	240	239	241	238	240
	否群件数	69	73	45	71	73	72	73	72	73
C社 改良開発	傾向	↓	ー	ー	↗	↓	↓	↓	→	↓
	良群件数	320	320	40	320	320	320	309	69	320
	否群件数	58	58	3	58	58	58	57	4	58

図６　横断的分析結果例

また、これにより「横断的分析アプローチ」が命題の実証に有効であることも確認できた。

2.3 定量的管理の普及促進

データ白書や定量的管理に関する下記の普及促進活動を実施した。

- ISBSG 主催の IT Confidence 2014 Conference にて、IPA が実施している定量的データ活用内容を発表。
- ET2014 にて、「ソフトウェア開発データ白書 2014-2015」の普及活動の一環として、デモ展示、セミナーを実施。
- JFPUG と IPA の共催による「ファンクションポイント法及びソフトウェア開発定量データの基礎と実践的活用」セミナーを 2014 年度内に 2 回開催。
- 「ソフトウェア開発データ白書 2014-2015」の概要及び追加分析について紹介するために、SEC セミナー「ソフトウェア開発データ白書 2014-2015」のデータ分析結果解説～組織における定量的管理のすすめ～」を開催。
- オープンソースとして公開中の定量的プロジェクト管理ツール（EPM-X）に関するセミナーについて、PPMA との共催セミナーを 2014 年度内に東京で計 6 回実施。

3 蓄積ソフトウェア開発データの活用促進

3.1 メトリクス分析に関する研究への活用

蓄積されているソフトウェア開発データをより一層活用し、ソフトウェアの信頼性・生産性向上につながる新たな分析手法の発見などを目指し、所定の守秘義務の下で蓄積データを東海大学、法政大学及び同志社大学に貸与し、各大学の研究に貢献した。

3.2 データ提供企業間での独自分析目的のデータ活用

ソフトウェア開発データ白書のデータ提供企業間に対してデータ活用ニーズの調査を行ったところ、ニーズがあるという結果が得られた。この結果を受け、データ活用の実現に向け運用の規約及び手順案を作成し、調整を開始した。

4 組込みソフトウェア開発データ白書

2013 年に開始した「組込みソフトウェア開発データ白書」の発行に向けた活動は、２年目にして総計 174 件の組込みソフトウェア開発プロジェクトデータを収集することができた。これは、文字通り組込み分野を対象にしたものである。

2013 年度は、６社から 65 件のプロジェクトデータを提供していただいたが、もっと多くのデータを集めるためには、SEC の取り組みが本気であることを広く産業界に知って貰い協力していただける企業を増やしていく必要があった。そのための広報活動として 2014 年７月、組込み総合技術展関西 2014（ETWest2014）にてプレス向けの発表を行った。その結果、プレス効果もあってか新たに組込み関連企業４社がデータ提供企業に加わり、総計 174 件の貴重なプロジェクトデータを集めることができた。

その後、これらのデータについての分析を試行した。その分析結果はデータ提供企業で構成される製品・制御システム定量データ収集・分析 WG で共有しているが、生産性

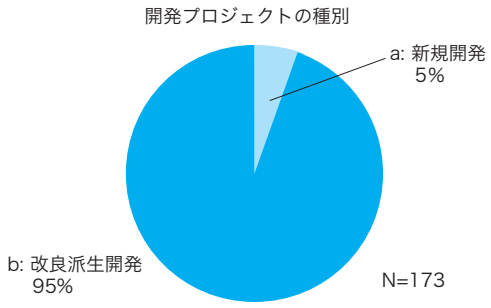
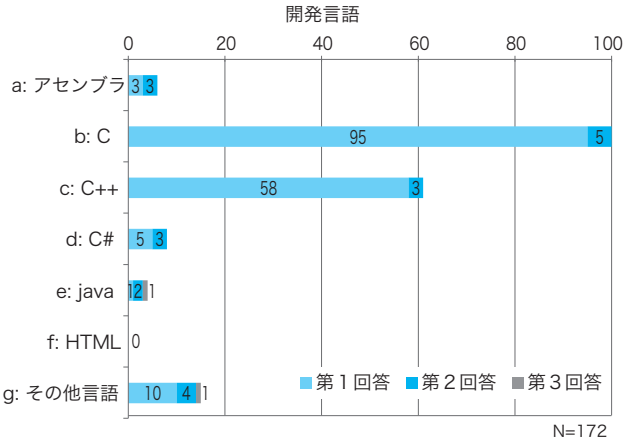


図７　開発プロジェクトの種別



注）第２回答、第３回答は複数言語を使っている場合の回答

図８　開発言語の種別

や信頼性の定量的な指標となり得るものや、定性的な傾向が見える結果も出ている。2015 年秋の「組込みソフトウェア開発データ白書」の公開に向けては、その中から価値のある分析結果を選定する予定である。以降では、現時点での分析結果を幾つか紹介する。

（１）新規開発案件

まず、新規開発がどれだけ行われているかの結果を見ると、収集データの範囲では、図７に示す通り全開発案件の５％に過ぎないことが分かった。

この結果から、「組込みソフトウェア開発データ白書」の活用対象は、新規開発プロジェクトではなく、既存ソフトウェアの改良、または既存ソフトウェアを資産とする類似ソフトウェアの開発にかかわるプロジェクトマネージャや品質保証関連部門をメインターゲットとした。

したがって、白書では、改良開発や派生開発プロジェクトのデータを分析した、価値ある結果を公開したい。

（２）開発言語

組込みソフトウェア開発のための言語は、圧倒的に C 言語が使われているとの想定の下、収集したデータを調べてみた（図８）。

その結果、172 件のプロジェクトのうち 100 件が C 言語を使っており、C++ 言語の使用も 60 件あった。C 言語、C++ 言語のそれぞれの用途については、古い装置やシステムの改良開発では、もともと C 言語で作られているものが多く、比較的最近の装置やシステムは C++ 言語で作られているようである。今回の収集データからは、C++ 言語の多くは C 言語の代用として使っており、必ずしもオブジェクト指向開発を実現する目的ではないようである。

（３）SLOC 規模

データ白書の利用者にとっては、自組織のプロジェクトと同じような規模、特性のプロジェクトデータが集まっていることが望ましい。ここでは、収集データの SLOC 規模の P25 値、中央値、P75 値を紹介する。

表１　SLOC 規模　［単位：KSLOC］

	N	P25	中央値	P75
SLOC 規模	173	2.4	6.1	24
SLOC 規模（母体含む）	173	61	249	511

2015 年秋公開予定の「組込みソフトウェア開発データ白書」に収められるプロジェクトデータの規模、言語、新規 / 改良派生開発のプロフィールを一部紹介した。読者がかかわる組込みシステム開発プロジェクトがこのプロフィールに近いものであれば、少しはお役に立てるものが提供できることを願って今後の活動を進めていく予定である。

5　ベンチマーキング標準化

ISO/IEC JTC1/SC7 にて進められている IT プロジェクトベンチマーキングの国際標準化に IPA 成果に基づく規格案を提案してきており、2014 年度には以下の進展があった：

- ISO/IEC 29155-3（ベンチマーキング―報告様式）が発行手続に移管。
- ISO/IEC 29155-4（ベンチマーキング―データの収集と管理）については、第一回 CD 投票※7 時のコメントを日本の主張を踏まえつつ反映し、第二回 CD 投票に付議。

【脚注】

※７　CD（Committee Draft）：委員会原案

システムグループ

組込みソフトウェア開発向けコーディング作法ガイド（ESCR）の改訂について

SEC 調査役
十山　圭介

SEC 調査役
三原　幸博

1 組込みソフトウェア開発向けコーディング作法ガイド（ESCR）の改訂状況

IPA/SEC では組込みソフトウェアのソースコード品質をより良いものとすることを目的に、C 言語と C++ 言語において、コーディングの際に注意すべき事柄やノウハウを「コーディング作法」という形で整備し、それらを取りまとめたガイドラインとして「組込みソフトウェア開発向けコーディング作法ガイド（ESCR）」（以下 ESCR）を公開している。ESCR は、コーディングの際の基本的な考え方（作法）と作法を具体化した守るべき個々の事項（ルール）をソフトウェア品質特性の観点で整理しており、組織内でコーディングルールを決める際や実際のコーディング時の参考、またプログラミング学習のため、書籍や PDF 版などこれまで３万部を超えて多くの方々に利用いただいている。

新たな機能を導入するなど言語の標準規格は定常的に改訂されており、ESCR についても多くのユーザが使用する規格に準拠して内容の更新の有無を検討し、必要な改版を実施しなければならない。ESCR［C 言語版］は 2006 年に Ver. 1.0、2007 年に一部修正を行った Ver. 1.1 を発行している。2014 年３月には JIS 最新の規格である C99 に準拠し、更に近年 C99 に対応して大幅に改訂された欧州組込み業界標準規格である MISRA C:2012 との整合性も確保した。

また、近年広く使用されるようになってきている C++ 言語向けにも 2003 年版の言語規格に準拠した ESCR［C++ 言語版］Ver. 1.0 を 2010 年に発行している。こちらについても、C++ 言語の新規格 C++11 及び C++14 に準拠し、また先に改訂した ESCR［C 言語版］との整合性を確保するべく、2014 年度より ESCR［C++ 言語版］の改訂作業を開始している。

2 ESCR のセキュアコーディングへの対応

ソフトウェアは常に攻撃の脅威に曝されており、ソフトウェアの欠陥による「脆弱性」が攻撃されることでセキュリティ被害がもたらされる。この脆弱性を作り込まない / 軽減させるよう、正しく動作するプログラムを書くことがセキュアコーディングである。

これまで、JIS によるソフトウェア品質規格では「セキュリティ」は「機能性」（設計段階の特性）に含まれる副特性であり、実装段階を支援する ESCR では積極的には取り上

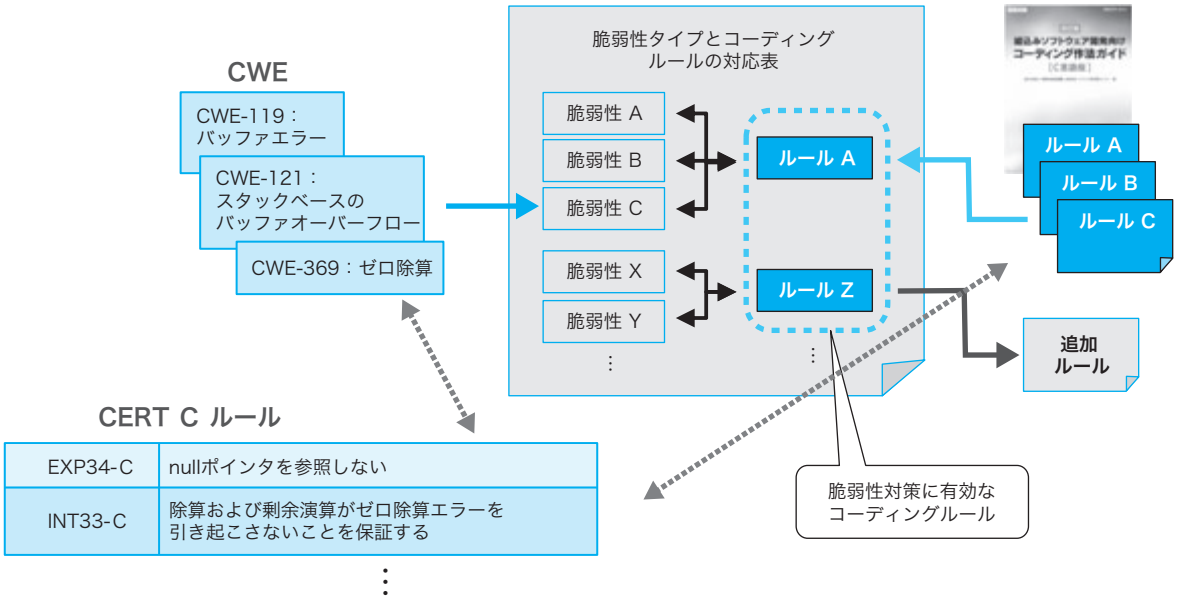


図１　ESCR ルールと CERT C ルール、CWE との関連

表1　ESCR ルールと CERT C ルール、CWE の対応表（抜粋）

作法詳細	ルール		MISRA ルールとの関係 C:2004	MISRA ルールとの関係 C:2012	CERT C	CWE
［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 領域は、初期化してから使用する。	R1.1.1	自動変数は宣言時に初期化する。または値を使用する直前に初期値を代入する。	9.1	R9.1	EXP33-C	CWE-456
	R1.1.2	const 型変数は、宣言時に初期化する。			EXP40-C	CWE-456
R1.2 初期化は過不足無いことが分かるように記述する。	R1.2.1	要素数を指定した配列の初期化では、初期値の数は、指定した要素数と一致させる。			ARR02-C STR11-C	
	R1.2.2	列挙型（enum 型）のメンバの初期化は、定数を全く指定しない、すべて指定する、または最初のメンバだけを指定する、のいずれかとする。	9.3	R9.4	INT09-C	CWE-665
R1.3 ポインタの指す範囲に気を付ける。	R1.3.1	（1）ポインタへの整数の加減算（++、-- も含む）は使用せず、確保した領域への参照・代入には [] を用いる配列形式で行う。 （2）ポインタへの整数の加減算（++、-- も含む）は、ポインタが配列を指している場合だけとし、結果は配列の範囲内を指すようにする。	17.1 17.4	R17.1 R17.4	ARR30-C ARR37-C	CWE-468 CWE-788 CWE-823
	R1.3.2	ポインタ同士の減算は、同じ配列の要素を指すポインタにだけ使用する。	17.2	R17.2	ARR36-C	CWE-469
	R1.3.3	ポインタ同士の比較は、同じ配列の要素、または同じ構造体のメンバを指すポインタにだけ使用する。	17.3	R17.3	ARR36-C	CWE-188
	R1.3.4	restrict 型修飾子は使用しない。【MISRA C:2012 R8.14】		R8.14	EXP43-C	
［信頼性 2］R2 データは、範囲、大きさ、内部表現に気を付けて使用する。						
R2.1 内部表現に依存しない比較を行う。	R2.1.1	浮動小数点式は、等価または非等価の比較をしない。	13.3	D1.1	FLP00-C	
	R2.1.2	浮動小数点型変数はループカウンタとして使用しない。	13.4	R13.1	FLP00-C FLP30-C	
	R2.1.3	構造体や共用体の比較に memcmp を使用しない。			EXP42-C	CWE-188
R2.2 論理値などが区間として定義されている場合、その中の一点（代表的な実装値）と等しいかどうかで判定を行ってはならない。	R2.2.1	真偽を求める式の中で、真として定義した値と比較しない。				
R2.3 データ型をそろえた演算や比較を行う。	R2.3.1	符号なし整数定数式は、結果の型で表現できる範囲内で記述する。	12.11	R12.10	INT30-C	CWE-190
	R2.3.2	条件演算子（?: 演算子）では、論理式は括弧で囲み、戻り値は 2 つとも同じ型にする。			INT02-C	
	R2.3.3	ループカウンタとループ継続条件の比較に使用する変数は、同じ型にする			INT02-C	

げていなかった。後継規格である JIS X 25010 で「セキュリティ」が特性として位置付けられたが、ESCR［C 言語版］Ver.2.0 ではセキュリティの観点での整理は行わず、バッファオーバーフローを避けるなどセキュリティに影響するコーディングもあるとして「CERT C コーディングスタンダード※1」を参照する旨を記載している。

一方、ESCR の個々のコーディングルールの中には、セキュリティの観点から見て重要なものが含まれている。それらと脆弱性との対応関係を示してセキュリティ面からも体系化することで、ソフトウェア品質の一層の向上に寄与できると見込める。

以上の背景から、SEC では IPA セキュリティセンターと連携して以下のように ESCR のルールとセキュアコーディングとの関連付けを実施した。

- 国際的に活用される脆弱性タイプ CWE（共通脆弱性タイプ一覧）※2 の中から、重要な 14 種を選定し、それを軽減しうる ESCR ルールとの対応を明らかにする
- ESCR ルールに対して、それと同等の意味を持つ CERT C コーディングスタンダードのルールを選定し、対応付けを行う

なお、この対応表については、ESCR の PDF 版の付録として公開した※3。

図 1 はこれらの対応関係の概要を図示したもので、表 1 は ESCR ルールと CERT C ルール、CWE の対応関係の表（抜粋）である。

【脚注】

※ 1　脆弱性につながる恐れのある危険なコーディングや未定義の動作を削減することを目的に定められたコーディング規約。CERT C Coding Standard の日本語版。

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

※ 3　https://www.ipa.go.jp/sec/publish/tn13-001.html

システムグループ

システム運用時の定量的管理方法

SEC 調査役　三縄　俊信
SEC システムグループリーダー　山下　博之

1 システム運用時の定量的管理の現状

システム構築時については以前より定量的管理の手法などがまとめられ、それにより情報処理システムの信頼性向上に効果を上げている事例も多くある。しかし、運用時についてはこれまで十分整理されていなかった（図１）。

	開発管理（システム構築）	運用管理（システム運用）
定量的管理	定量的開発管理あり	定量的運用管理？
目的 （究極には）	目的：開発プロセスの改善 効果：リスクの早期発見 ・プロジェクトの失敗 ・信頼性の低下 見積り／計画 ↓ 信頼性向上／生産性向上	目的：運用プロセスの改善 効果：リスクの早期発見 ・システム障害 ・キャパシティ超過 見積り／計画？ ↓ 信頼性向上／効率向上
対象	開発プロセス（組織、マネジメント含む） システム（構築中プロダクト）	運用プロセス システム（運用中プロダクト） 構築プロセス、アーキテクチュアへフィードバック
方法	・メトリクス（判断基準と対策含む） ・分析技術 ・（インプロセス）モニタリング ・ベンチマーキング	・メトリクス（判断基準と対策含む） ・分析技術 ・モニタリング ・ベンチマーキング

図１　定量的管理に関するシステム構築時と運用時のアナロジー

そのため、主として定量的なアプローチによる運用時の信頼性向上に対する取り組み（使用指標、指標測定データに基づく対策手法、予兆などの観測項目、観測データの分析手法など）の現状を明らかにし、その課題を見出すことを目的として調査を実施した。

2 調査の内容

運用にかかわる指標、プロセス、ツールなど定量的管理の視点から、文献や公開情報の収集及び各産業分野の企業など（１大学・８企業）へのヒアリングを通じて、次の項目について調査を行った。

- 現在の情報処理システムを取り巻く環境やシステムの特徴
- 運用プロセスとして標準的に参照されている国際規格
- 運用時の定量的指標として提案されたり実際に使われたりしている指標の事例と考察
- プロセス管理や障害予知に関するツール及び研究事例
- 運用に関する技術とツールの現状及び動向
- 運用のプロセスや指標に関する実態及び運用時の信頼性にかかわる知見や問題点
- 運用時の信頼性に関する現状分析及び今後の動向と課題
- 公的機関などにおける今後の取組み課題の提案
- 情報処理システムのライフサイクルにおける運用の位置付けと運用時の信頼性向上に関する方向性

3 調査の結果

多くの組織での下記の現状や課題が明らかとなった。

- 情報処理システムではなくサービスの運用の視点を持つ
- ITIL※1 を参考にして独自に運用プロセスを作成している
- KPI（運用指標）を設定して運用管理を行っているが、SLA（サービスレベル指標）と KPI の関連付けは課題
- 統合監視ツール活用も、障害予兆検知は今後の課題
- 運用の人材育成やスキル評価の問題意識を持つ

とくに、顧客あるいはステークホルダとの間の契約としての SLA などエンドユーザーに対するサービス品質を直接的に表す指標を運用の最終目標として定めている場合には、SLA を実現するために必要な要素に分解した指標として、あるいは SLA を補うための目標となる指標として KPI などの内部指標を定めて運用の目標値としている。図２にその一例を図示する。

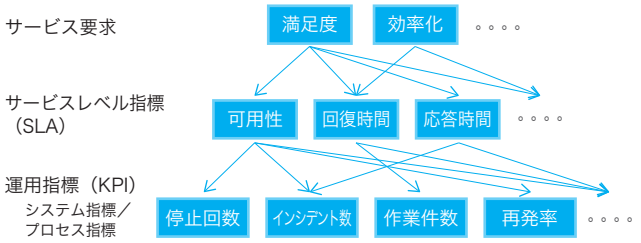


図２　運用指標の階層構造例

本調査の結果を「情報システム運用時の定量的信頼性向上方法に関する調査報告書」として取りまとめ公開した※2（2015 年４月）。本調査により明らかとなったこれらの現状や課題を受け、情報処理システムに対する認識の浸透と、システム運用時における取り組みの見直し・改善につながることを期待すると共に、今後も情報処理システムにおける課題の解決に向けて取り組んでいく予定である。

【脚注】
※１　Information Technology Infrastructure Library
※２　http://www.ipa.go.jp/sec/reports/20150416_2.html

ソフトウェアグループ

つながる世界に向けた基盤作り
～ 2014 年度の取り組み結果～

SEC ソフトウェアグループリーダー
中尾　昌善

1 はじめに

今日、ソフトウェアが組み込まれた製品・システムは日常生活に無くてはならない社会基盤となってきている。さらに、そのような製品・システムがつながって、新しいサービスを生み出す世の中へと変遷しつつある。例えば、家庭内の機器をつなげるスマートハウスやスマート家電、複数のヘルスケア製品をつないだ健康サービスなどが考えられる。このように、あらゆる製品・システムがつながって新しいサービスや価値を創造する世の中を、我々は「つながる世界」と呼んでいる。そこでは、色々な品質の製品が氾濫し、それらを利用者が自由に選択してつなぐことが可能になる。一方、意図しないつなぎ方により、安全上やセキュリティ上での問題を引き起こす危険性もでてくる。

この利用者リスクの増大を防止する取り組みが必要と考え、2013 年度からの IPA 第三期中期計画では「ソフトウェアの利用者視点での信頼性の見える化」という取り組みを開始した。その取り組みの中で、2014 年度は、以下の活動を行ってきた。

2 コンシューマデバイスの開発方法論の標準化

コンシューマデバイスとは、自動車、家庭用のサービスロボット、スマートハウス（スマート家電）など、一般の利用者が使用する機器のことである。このコンシューマデバイスの開発方法論の国際標準案を、産業技術総合研究所、トヨタ自動車株式会社、富士通株式会社、電気通信大学との共同で、OMG（Object Management Group）に提案してきた。その結果、2014 年度末に OMG の国際標準規格として成立した。

今後のつながる世界では、コンシューマデバイス間の連携は欠かせぬものになると想定され、その開発方法に関して我が国からの規格提案が認められたことの意義は大きく、各産業界での適用が期待される。

3 ソフトウェア・サプライチェーンの課題解決

ソフトウェア・サプライチェーンの課題調査を行い、その結果を「ソフトウェア開発の取引構造（サプライチェーン）の実態にかかわる課題の調査報告書」にて公開した。そこから見えてきた数多く存在する課題の中から、まずは、セーフティ＆セキュリティ設計とその見える化に関する取り組みを行った。

4 つながる世界に向けたソフトウェア品質ガイド

これまでのソフトウェア開発では、品質と言えばバグ含有量を指すというのが一般的な理解であった。しかし、異分野の製品がつながる世界では、品質に関する共通的な捉え方が必要であり、SQuaRE のような標準規格に従う動きが加速している。そこでは、品質はバグ含有量だけを指すものではなく、性能効率性や使用性などの幅広い観点での捉え方になっている。その理解の手助けとするために、「つながる世界のソフトウェア品質ガイド」と「SQuaRE 品質モデル活用リファレンス」を作成した。

5 先進的な設計・検証技術の適用事例

ソフトウェア開発方法によるコスト、品質、手間などの改善は、ソフトウェア産業界では永遠のテーマであり、多くの企業がチャレンジしている。そこには、単純な手法適用だけでなく、現場に則した工夫や改善適用が存在している。過去 2 年間に収集したこれらの事例は、48 事例に及び、既に公開済の事例については、読者からの「具体的な取り組みが伝わり有意義だ。」などの感想とともに、自社の事例をぜひとも紹介したいという申し出もあり、高い関心が寄せられている。

6 おわりに

つながる世界でのソフトウェアの信頼性の確保に向けた課題は多岐にわたり、今後もそれらの課題に焦点を当てた取り組みを実施していく。

ソフトウェアグループ

コンシューマデバイス機能安全規格が正式に OMG 標準規格へ

SEC 研究員 春山　浩行
SEC 調査役 室　修治
SEC 専門委員 内田　功志

1 はじめに

一般消費者向けのシステムを対象とした製品※1をコンシューマデバイス（消費者機械）と呼ぶ。コンシューマデバイスには高いディペンダビリティ※2が求められ、既に自動車に関しては ISO 26262 でその機能安全規格が定められている。しかし、これは自動車に特化したものであり、あらゆるコンシューマデバイスに適用できるものではなかった。そこで、2013 年 11 月、あらゆるコンシューマデバイスに横断的に適用できる体系化された枠組みとして、Dependability Assurance Framework for Safety Sensitive Consumer Devices（DAF for SSCD）を OMG※3に初期提案した。そして、2015 年 3 月に正式に OMG の標準規格となった。

DAF for SSCD の主な特徴は以下の 3 つである。

① 対象がコンシューマデバイス全般であること
② ディペンダビリティ向上のための仕組みであること
③ 個別の特性を保証する仕組みを開発プロセスと開発の仕組みで体系化していること

2 DAF for SSCD の概要

ディペンダビリティを保証する体系的な枠組みの標準規格 DAF for SSCD は、以下の 3 つで構成されている。

(1) Dependability Conceptual Model（DCM）

【課題】機能安全などの国際規格は、自然言語で記載されているため、全体の構造が分かりにくかった。

【対応策】提案する規格をメタモデル（概念モデル）として構造を見える化し、複雑な要素が絡み合った規格をやさしく見通すことができるようにする。

DCM では、コンシューマデバイスのディペンダビリティを保証するための規格の概念構成を UML のクラス図を使って提示した。

(2) Dependability Process Model（DPM）

【課題】従来の国際規格では、安全性などを実現する開発プロセスに関してはほとんど言及されていなかった。

【対応策】コンシューマデバイスのディペンダビリティを保証するための開発プロセスをモデルベースのグラフィカルな標準記法で規定できるようにする。

DPM では、ディペンダビリティを保証するためのコンシューマデバイスの開発・運用プロセスを BPMN※4で規定した。

(3) Dependability Assurance Case（DAC）のテンプレート

【課題】コンシューマデバイスのディペンダビリティを保証するためには様々な保証ケースを整備する必要があった。

【対応策】コンシューマデバイスのディペンダビリティを確保するためのより多くの保証ケースを作成し、各業界で流用可能なテンプレートを用意できるようにする。

DAC のテンプレートにより、コンシューマデバイスのディペンダビリティを保証する保証ケース※5の雛形をできるだけ多く用意し、作業効率の向上を図る。本規格では、検討が先行する自動車のエンジンストールの事例を一部提示した。

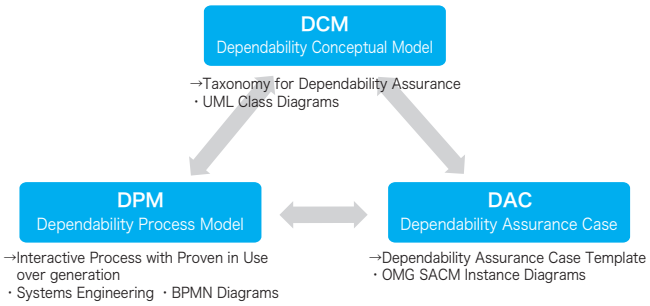


図 1　3 つの定義で構成する DAF

3 おわりに

IoT 時代では、コンシューマデバイスがインターネットにつながる世の中になる。そのような時代に DAF for SSCD が世界標準の一つとして認められたことには重要な意味がある。

2015 年度は、9 月に DAF for SSCD を正式に OMG 標準規格として公開し、普及活動を始動する予定である。

【脚注】

※ 1　自動車、サービスロボット、スマート家電、スマートハウスのようなものが該当する
※ 2　ディペンダビリティ：信頼性性能、保全性性能及び保全支援能力を記述するために用いられる包括的な用語とされている
※ 3　Object Management Group：国際的な標準化団体
※ 4　Business Process Modeling Notation：ビジネスプロセスモデリング表記法
※ 5　ディペンダブルにするための観点、実現手段、実現された証拠等からなる文書など

ソフトウェアグループ

セーフティ＆セキュリティ設計とその見える化の推進

SEC 研究員
鈴木　基史

SEC 研究員
西尾　桂子

SEC 研究員
宮原　真次

1 背景と課題

複数の健康器具を組み合わせたヘルスケアサービスや、スマートフォンで家電を制御するサービスなど、異なる分野の製品やサービスを組み合わせた新たなサービスが始まっており、今後は、更に様々な製品などによる高度なつながるサービスが出現すると見込まれる（図１）。

こうした認識の下、2013 年度に実施した「ソフトウェア開発の取引構造（サプライチェーン）の実態にかかわる課題の調査」において抽出し整理した課題の中から、今後とくに対応が求められる「ユーザ組み合わせ型への変化」の課題とその対策案に沿った取り組みを 2014 年度より開始した。

【課題】

- ユーザ自ら製品・サービスを選択し、組み合わせて利用する形態が増え、利用時の品質を出荷時に想定して検査することが難しくなり、品質確定のタイミングが開発段階から利用段階にシフトした。
- 製品・サービスを提供する複数の企業の責任の所在があいまいになった。
- 組み合わせ利用によるセキュリティのリスクが増大した。
- 利用者が連携時のリスクを十分に理解できていない。

【対策案】

- 組み合わせ利用における動作保証範囲、提供者の責任範囲を明確にし、利用者に対して注意喚起も含む、分かりやすい説明を行う。
- 必要に応じて製品間の制御可否を行う仕組みを導入する。

2 課題への取り組み

つながる世界において、利用者がつながる製品やサービスを安全・安心に利用するために、サプライチェーンを構成する事業者が取り組むべき事項として、組み合わせ利用における動作保証範囲、提供者の責任範囲を明確にし、利用者に対して分かりやすく説明する仕組みが今後必要である。この仕組みの実現のためには、接続先システムの品質（とくに重要なのがセーフティとセキュリティの品質）の見える化[※1]が必要であり、このためにはセーフティ設計[※2]とセキュリティ設計[※3]が確実に実施されることが重要である。

実際に IPA/SEC で実施したアンケートでは、セーフティ設計とセキュリティ設計について、「具体的に何を

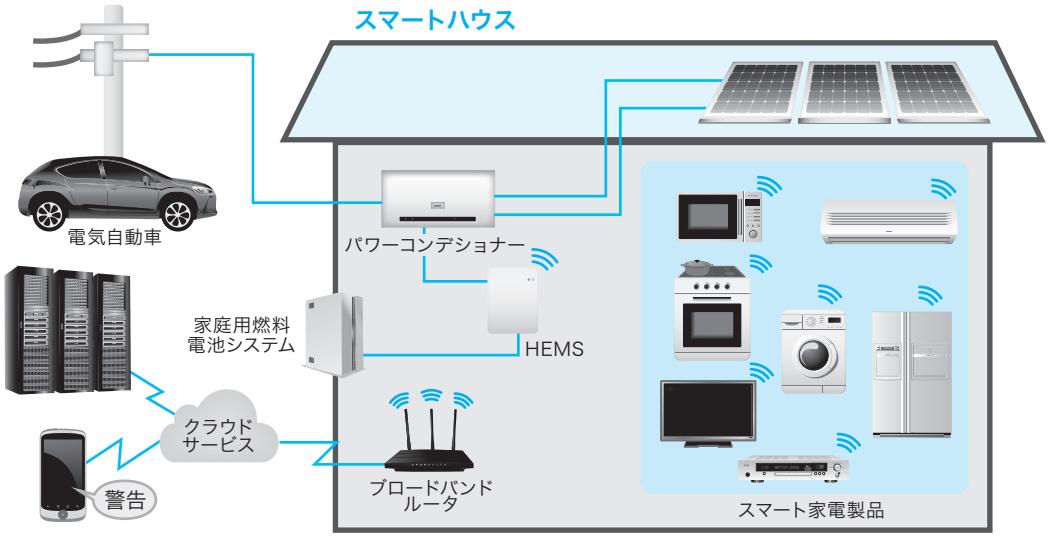


図１　製品やサービスを組み合わせて使用する「つながる世界」例

すれば良いかがわからない」「具体的なやり方が分からない」「何をどこまでやれば良いのか分からない」などの課題や、セーフティ設計とセキュリティ設計の見える化については、「チェックすべきポイントが理解されていないので何を見えるようにすべきか決めかねている」や「表記法がまだ固まっていない」などの多くの課題がある現状が見えてきたため、これらの課題に取り組む必要がある。

2.1 セーフティ・セキュリティ設計の見える化推進のための調査結果

そこで、IPA/SEC では、4 分野（自動車、スマートフォン、ヘルスケア、スマート家電）の先進的な取り組みを行っている企業におけるセーフティ設計とセキュリティ設計の実施状況について明らかにするために、「セーフティ・セキュリティ設計の見える化推進のための調査」を実施した。調査においては各企業で使われているセーフティとセキュリティの分析手法・対策手法と共に、セーフティ設計とセキュリティ設計の見える化の手法・技術についても調査を行った。

調査結果からは、ほとんどの人がセーフティ設計とセキュリティ設計の両方が必要だと考えていると分かった一方で、担当部門では「製品のセーフティ設計」または「製品のセキュリティ設計」に関する明文化された基本方針を持っていないとの回答が半数以上を占めており、セーフティ設計とセキュリティ設計について十分に普及していないことも明らかとなった。

2.2 ワーキンググループ（WG）より得られた知見

前述の課題解決に向け、セーフティ設計とセキュリティ設計及びその設計品質の見える化の普及のために、これらの分野の有識者から成る「サプライチェーンにおける品質の見える化 WG」を設置した。

WG 活動を進めるに従って、セーフティ設計に関しては、過去からの確立した設計手法があるものの、セキュリティ設計に関しては、まだ新しい分野のため公開情報が少なく、標準的な設計手法（分析・対策）の確立には至っておらず、独自の手法も多く使われていることが分かってきた。また、同じ業界の中でも、セーフティ分野とセキュリティ分野とでは使用する用語の意味や使い方が違うということも明らかになった。さらに、セーフティ設計とセキュリティ設計を統合したプロセスはまだ確立されているとは言えない状況であることも分かった。

このような状況の中、WG 委員からの意見などを踏まえ、セーフティとセキュリティの分析・対策手法を中心に初心者にも分かりやすい内容のガイドブックとして、「つながる世界のセーフティ & セキュリティ設計入門」の作成を行った。

2.3 ガイドブックの特徴

今回作成したガイドブックは以下の特徴を持つ。

① セーフティ設計、セキュリティ設計、その設計品質の見える化の 3 つを 1 冊のガイドブックに整理
② セーフティ設計とセキュリティ設計の分析・対策などの手法を初心者に分かりやすく解説
③ アシュアランスケースの表記法を設計品質の見える化手法として紹介

前半の 1 ～ 3 章は、セーフティ設計とセキュリティ設計の必要性をマネージャ層にも理解してもらえるような内容になっている。例えば、2 章ではセーフティ設計にかかわる事故事例やセキュリティ設計にかかわるインシデント事例を掲載している。また、後半の 4 ～ 6 章では、前述のセーフティ・セキュリティ設計の見える化推進のための調査結果から、実際の開発現場で使われている分析、対策、見える化の手法を中心に解説しているので、ソフトウェア技術者の参考になる内容となっている。

「つながる世界のセーフティ＆セキュリティ設計入門」
（2015 年発行予定）

・セーフティ＆セキュリティ設計、その設計品質の見える化を 1 冊のガイドブックに整理
・セーフティ設計とセキュリティ設計の分析・対策手法を初心者に分かり易く解説
・アシュアランスケース表記法を見える化手法として紹介

1 章 つながるシステムのセーフティとセキュリティ
2 章 事故及びインシデント事例
3 章 セーフティとセキュリティのための開発プロセス
4 章 ソフトウェア技術者のためのセーフティ設計
5 章 ソフトウェア技術者のためのセキュリティ設計
6 章 ロジカルな設計品質の説明

図 2　ガイドブックの特徴と目次

3 今後の取り組み

2014 年度はセーフティ設計とセキュリティ設計の重要性を紹介したプレセミナーを開催した（2015 年 3 月）。2015 年度は、本ガイドブックに基づいたセミナーを行い、セーフティ設計とセキュリティ設計及びその見える化の普及を図っていく予定である。

【脚注】

※ 1　対象システム（製品）の設計品質（セーフティやセキュリティなど）が設計において確保されていることを、エビデンスを使って論理的に第三者に分かるように説明

※ 2　設計の段階での安全の作りこみ。そのためのリスク分析とリスク低減

※ 3　設計の段階で脆弱性の低減や脅威への対策を考慮。そのためのリスク分析とリスク低減

ソフトウェアグループ

つながる世界に向けたソフトウェア品質ガイド

SEC 研究員
宮崎　義昭

SEC 研究員
細目　紀子

SEC 研究員
宮原　真次

1 はじめに

様々な製品やサービスが複雑に連携した「つながる」システムの増加に伴い、新たな品質上のリスクが高まっている。IPA/SEC では、製品・サービスを提供する事業者が理解しておくべき品質に関する基本的な知識と、国際規格 SQuaRE※1 の活用について解説した「つながる世界のソフトウェア品質ガイド　～あたらしい価値提供のための品質モデル活用のすすめ～」を作成、発行した※2。

2 背景

IT を活用したビジネスの領域は、様々な製品やサービスが複雑に連携しながら「つながる」システムを構成し、利用者に新たな価値を提供する世界に確実にシフトしている。このような価値を利用者に届けるために、また利用者に対する品質に関する説明責任を果たすために、事業者にはこれまで以上に品質に対する広範囲な理解と包括的な取り組みが求められる。

IPA/SEC では、品質説明力を強化するための環境整備の一環として、事業者の品質説明が適切であることを、第三者の専門家が客観的に評価し、利用者に向けて分かりやすく公開する仕組みを構築するための指針をまとめた「製品・システムにおけるソフトウェアの信頼性・安全性等に関する品質説明力強化のための制度構築ガイドライン」（以降、制度ガイドライン）を作成し、2013 年 6 月に公開した。その後、業界団体などが具体的な制度構築を行う際に、この制度ガイドラインの活用を支援するなど普及を行い、実際に制度ガイドラインに準拠した制度が構築された。しかし、具体的な制度設計にあたっては、客観的な品質評価の基準を定義し、かつ公正に評価するのが難しいという課題があった。そこで、2014 年度はこの対策として、品質基準の定義と評価の指針を検討し整理するための活動を行った。

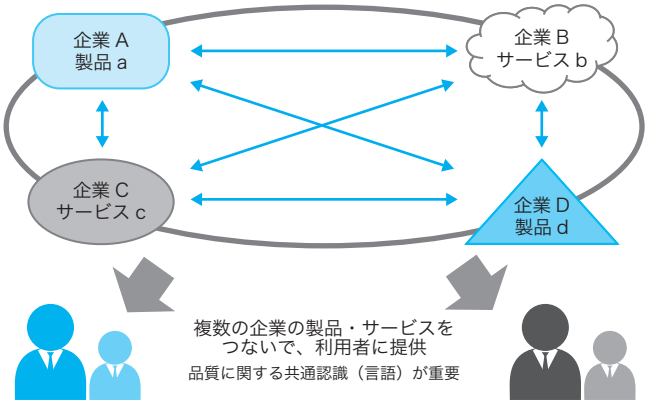


図 1　「つながる」システム

3 つながるシステムの品質を整理する難しさ

第三者が客観的かつ公正な品質評価を行うためには、多様化する品質や様々なステークホルダの存在を意識した、広範囲な視点からの品質の分類と整理が必要となる。しかし、IT を活用した製品の種類や社会における役割が増えるにつれて、利用者の期待は、提供される機能だけにとどまらず、安全性・セキュリティはもとより、快適さや楽しさ、またビジネスへの高度の貢献といった内容まで多様化し、かつ高い満足度を得られることが期待されている。また、ひとつの製品・サービスに関与するステークホルダ（様々なタイプの利用者、販売者、システム運用者、など）は多岐にわたるため、これらのステークホルダが求める品質を漏れなく洗い出すことは容易ではない。

さらに、スマートフォンを利用したクラウドサービスや、街全体の生活基盤となるスマートコミュニティのような「つながる」システムは、その活用範囲が広がっており、その品質の良し悪しが国民生活へ大きな影響を与えるリスクをはらんでいる。「つながる」システムの構築には、これまで接点がなかった複数の事業者が直接的あるいは間接的に協力することが必要だが、事業者間で品質に対する定義や考え方が異なると、システム全体の品質について正しい共通認識を持つことが難しい。

4 国際規格 SQuaRE による課題解決

このような課題に対して、システムやソフトウェアの多様な品質を整理する上での枠組みを規定した国際規格として SQuaRE がある。SQuaRE は、品質モデル（製品

品質、利用時の品質、データ品質）について、品質特性及び品質副特性を定義しており、品質を幅広く捉えることができるようになっている。製品・サービスを提供する事業者が、広い視点で各ステークホルダの品質要求を洗い出し、かつ定量的に評価する取り組みを進めるためには、SQuaRE の活用が有効である。また、SQuaRE は、「つながる」システムにおいて、複数の事業者が品質に対する共通認識を持つうえでも有用である。

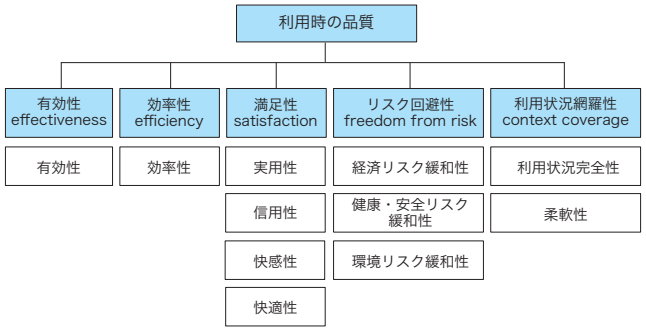


図２　利用時の品質モデル※3

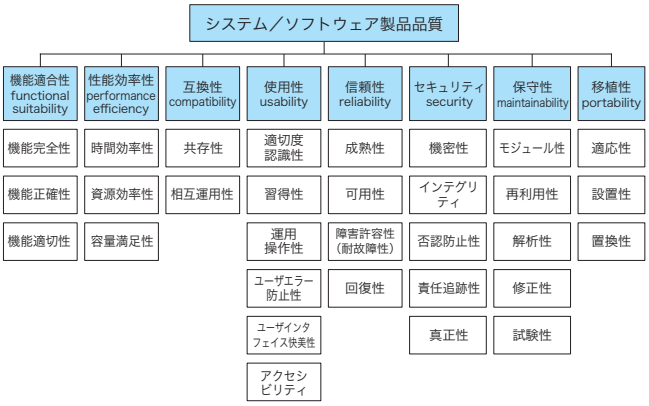


図３　製品の品質モデル※3

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

固有の視点からのデータ品質特性

固有の視点及びシステム依存の視点からのデータ品質特性

システム依存の視点からのデータ品質特性

図４　データ品質モデル※4

5　ガイドブックの概要

ガイドブックでは、上記で説明した背景、品質の考え方、各事業分野における品質確保の取り組み、及び SQuaRE の活用に関する基本的な知識を分かりやすく取りまとめている。現場の技術者やリーダー、さらには会社経営者にも読んでもらい、製品の品質確保に役立ててもらうことを期待している。

本ガイドブックの構成と内容：

ソフトウェア品質ガイド編（約 100 頁）
品質をあらためて考える背景、国際規格 SQuaRE の概要、ユーザビリティ / セーフティ / セキュリティなど重要な品質に関する解説、品質向上に向けた改善ポイント、品質説明と第三者評価、などについて分かりやすく説明。
SQuaRE 品質モデル活用リファレンス編（約 110 頁）
国際規格 SQuaRE で規定された品質モデル（製品品質、利用時の品質、データ品質）について、品質特性 / 品質副特性の各項目について、説明 / ニーズ例 / 測定量の例などを記載。 実際に品質要件の洗い出しや評価計画を作成する場面で、作業効率を期待。

6　普及に向けた活動

2014 年度は、ここで紹介した事業の普及活動として、展示会などでの紹介のほかに、SEC セミナーを開催した（2015 年 3 月）。2015 年度は、本ガイドブックをより多くの事業者で活用してもらうため、具体的な活用方法の提案や適用事例の紹介など、様々な普及活動を推進する予定である。

【脚注】

※ 1　Systems and software Quality Requirements and Evaluation：システム及びソフトウェア製品の品質要求及び評価に関する国際規格 ISO/IEC 25000 シリーズ、国内規格 JIS X 25000 シリーズの総称。スクウェアと読む。SEC journal 39 号 特集：品質について考える に関連記事掲載
http://www.ipa.go.jp/files/000043960.pdf

※ 2　2015 年 3 月ダイジェスト版を公開
http://www.ipa.go.jp/sec/reports/20150331_1.html
2015 年 5 月書籍発行
http://www.ipa.go.jp/sec/publish/20150529.html

※ 3　JIS X25010:2013（ISO/IEC 25010:2011）に基づく

※ 4　JIS X25012:2013（ISO/IEC 25012:2008）に基づく

ソフトウェアグループ

先進的な設計・検証技術の適用事例

SEC 調査役 室　修治　SEC 研究員 春山　浩行　SEC 研究員 藤原　由起子　SEC 研究員 佐々木　方規

1 はじめに

2014 年 5 月に公開した「先進的な設計・検証技術の適用事例報告書 2013 年度版」では設計事例 13 件、検証事例 11 件の計 24 件の事例を紹介した※1。

2014 年度は上記事例を念頭にサービス・製品分野、該当開発工程、適用技術・手法の網羅性を向上させるべく設計事例 12 件、検証事例 12 件の計 24 件を収集した。

さらに、この 2 年間の収集事例 48 件を後述する様々な観点で分析・整理することにより新技術の現場適用のための有益な参考情報としている。これら分析とさらに追加で収集している事例をまとめた書籍を刊行予定である。

今後は、「つながる世界」に向けた取り組みやそれらを支えるであろうと考えられている技術に重点を置き、更にその適用事例の収集件数を増やしていく。また、2014 年度に実施した分析を進め、有用性や適用領域を整理したガイドブックを作成する予定である。

2 今回収集した事例の特徴

① 前年度収集件数が比較的少なかったエンタープライズ系の事例を多く収集した。
② より上流側の取り組み事例を多く収集した。
③ 適用技術については、アジャイル開発、モデルベース開発、BPM※2、テストの自動化などを多く収集している。

3 収集した事例の一覧

今回収集した事例を、設計と検証に大別して、表 1 に示す。

4 普及に向けた活動

有用な技術や手法の普及のために、以下の 2 つの観点でセミナーを開催している。

① 導入への関心を高めるための事例紹介セミナー
② 具体的な技術や手法に関する技術セミナー

昨年度は、ソフトウェアの高信頼化に関連した技術（MBSE※3、形式手法など）に関するセミナーを開催した。

今後も事例紹介セミナーとその事例に適用されている技術に関するセミナーを企画・開催していく予定である。

5 事例からの分析

2 年間の収集事例数は 48 件となった。書籍刊行時は約 60 件まで収集できる予定である。分析は事例から得られる有益な情報を効果的に参考とできるよう様々な観点で実施した。下記に一部を紹介する。

- 実施目的
- 適用技術
- 工程と技術の関連
- 実施時期と適用技術の傾向
- 適用技術と同時に実施した取り組み
- 想定効果と結果の関連
- 効果の傾向

（1）適用工程と適用した技術・手法

すべての事例について適用工程を明確にした。参照プロセス標準は「共通フレーム 2013※4」であり開発フェーズを中心として、企画フェーズ / 要件定義フェーズ / 移行・運用準備フェーズまでを範囲としている。

事例が適用した技術・手法は事例中に現れた技術・手法を整理・分類し事例で収集できなかった重要と認められる技術・手法を追加し、各事例とマトリクス状にマッピングした（表 2）。

ある技術・手法を導入したい、ある工程に課題があるという場合、該当箇所にマークされた事例が参考となる。

（2）課題の分類と解決状況

各事例が何を解決すべき課題とし（1）適用工程と適用した技術・手法で示した技術・手法を実践した結果としてどのような効果が得られたかを分析・整理した。

【脚注】
※ 1　http://www.ipa.go.jp/sec/reports/20140530.html
※ 2　Business Process Management
※ 3　Model-Based Systems Engineering
※ 4　http://www.ipa.go.jp/sec/publish/tn12-006.html

（3）その他分析・整理

技術・手法を導入した効果・成果、技術・手法を導入するためにとった施策・工夫、技法・手法を導入した上での新たな課題、などについても事例からできるだけ抽出している。新たに技術・手法を導入する際の参考として使っていただきたい。

課題は品質、コスト、納期、生産性など事例から抽出し、適当と考えられる形で項目化した（表3）。

解決したい課題について、該当箇所にマークされた事例が参考となる。

6 おわりに

本年度についても事例収集を継続し、分析・整理含め拡充していく。収集分野はとくに要望の多いシステムズエンジニアリングやモデルベース開発、セキュリティ対応開発、派生開発について充実していく。成果は書籍として刊行する予定である。また例年通り、事例を提供していただいた企業から講師を招き、現場の生の声が聴ける事例紹介セミナーなどを企画している。

表1　先進的な設計・検証技術の適用事例一覧（2014 年度収集）

No.		標題	事例提供元
設計	1	BPM をベースにした会社統合での新業務プロセス設計の適用事例	三菱商事 RtM ジャパン 株式会社
	2	人間系プロセスを含む業務を BPM ソフトウェア活用により改善した事例	日本電気 株式会社
	3	D-Case 導入によるシミュレーションS／Wの期待結果明確化と合意形成	三菱電機 株式会社
	4	ソニーの電子お薬手帳システム「harmo」に適用したセキュリティ設計分析手法	ソニーデジタルネットワークアプリケーションズ 株式会社
	5	XDDP におけるデグレード防止効果を高めるための手法 －『気づきナビ』の考案－	アズビル 株式会社、株式会社 インテック、キヤノン IT ソリューションズ 株式会社
	6	組込システムのモデルベース開発適用における DI コンテナの活用	ヤマハ 株式会社
	7	大規模システムへのモデルベース開発手法の適用	株式会社 IHI エアロスペース
	8	自動車のパワーバックドアシステム開発のためのモデルベースシステムズエンジニアリングの適用	慶應義塾大学、日産自動車 株式会社
	9	D-Case を用いたゴール共有による開発プロセスの適用 ～ ET ロボコンでの試行と成果～	富士ゼロックス 株式会社
	10	システム記述言語（AADL）による複合システム設計 （航空機前方車輪の回転数から速度を計測・記録・表示するシステムへの適用）	Institut Supérieur de l'Aéronautique et de l'Espace（ISAE）
	11	ロケットエンジンにおけるモデルベース信頼性評価技術の構築と試行	国立研究開発法人 宇宙航空研究開発機構
	12	デジタル制御電源製品開発に対するモデルベース開発の適用	株式会社 富士通研究所
検証	13	SysML と CML によるシステムオブシステムズの検証	Newcastle University、 FP7 COMPASS プロジェクト
	14	通信制御ソフトウェア開発における状態遷移設計の品質向上への取組み ～状態遷移表へのモデル検査の適用～	富士通 株式会社
	15	Friendly による内部 API を使ったシステムテスト自動化	株式会社 Codeer
	16	アジャイルプロセスにおける実践的な品質向上施策の適用事例	株式会社 日立ソリューションズ
	17	メトリクス分析手法を用いた試験品質向上の取組み	株式会社 東芝
	18	ユーザーエクスペリエンスを業務に定着化させるための取り組み事例の紹介	株式会社 ベリサーブ
	19	パッケージ開発プロセス改善による品質向上と生産性向上	株式会社 富士通マーケティング
	20	Web システムにおける単体テストの品質向上の取組み	住友電工情報システム 株式会社
	21	安心なサービスの品質改善を実現する為の継続的システムテスト	楽天 株式会社
	22	セキュア開発手法の考察と診断ツールの活用事例の紹介 ～お客様に「安心してご利用ください」と言えるための脆弱性対策～	ビッグローブ 株式会社
	23	モデル検査とテストによる車載オペレーティングシステムの検証	北陸先端科学技術大学院大学
	24	モデルベース開発とコード解析を用いた組込みソフトウェアの開発	アルプス電気 株式会社

表２　適用工程と適用した技術・手法

No.	標題	適用領域①		適用領域②			適用工程 共通フレーム 2013									
		設計	検証	エンタープライズ系	Web・フロント系	エンベデット・制御系	企画フェーズ	要件定義フェーズ	開発フェーズ						移行・運用準備フェーズ	運用・保守フェーズ
									システム・ソフトウェア要件定義	基本設計	詳細設計	製作・ユニット・テスト	ソフトウェア結合・総合テスト	システム結合・総合テスト		
1	BPM をベースにした会社統合での新業務プロセス設計の適用事例	■		■				■	■							
2	人間系プロセスを含む業務を BPM ソフトウェア活用により改善した事例	■		■				■	■	■	■	■				
3	D-Case 導入によるシミュレーションS／Wの期待結果明確化と合意形成	■		■				■	■	■	■	■	■	■		
4	ソニーの電子お薬手帳システム「harmo」に適用したセキュリティ設計分析手法	■		■	■			■	■			■				■
5	XDDP におけるデグレード防止効果を高めるための手法－『気づきナビ』の考案－	■		■			■	■	■							
6	組込システムのモデルベース開発適用における DI コンテナの活用	■		■					■	■	■	■	■	■	■	■
19	パッケージ開発プロセス改善による品質向上と生産性向上	■				■			■	■	■	■	■	■		
20	Web システムにおける単体テストの品質向上の取組み	■				■			■	■	■					
21	安心なサービスの品質改善を実現する為の継続的システムテスト	■				■		■	■							
22	セキュア開発手法の考察と診断ツールの活用事例の紹介～お客様に「安心してご利用ください」と言えるための脆弱性対策～	■				■				■	■	■				
23	モデル検査とテストによる車載オペレーティングシステムの検証	■				■		■	■	■					■	■
24	モデルベース開発とコード解析を用いた組込みソフトウェアの開発	■				■					■	■				

No.	技術・手法																																
	非機能要求						機能要求・ソフトウェア開発方法																										独自開発・検証ツール
		全般	運用	アシュアランスケース（D－Case）	セキュリティ	リスクアセスメント		アジャイルソフトウェア開発モデル	スパイラル開発モデル	ソフトウェアプロトタイピング開発モデル	イテレーション開発モデル【反復型開発モデル】	W（開発）モデル	BPM【ビジネスプロセスマネジメント】	MBSE【モデルベースシステムズエンジニアリング】	形式仕様記述	HCD【人間中心設計】	UX【ユーザー・エクスペリエンス】	MBD【モデルベース開発】	MDD【モデル駆動開発】	保守・更改・派生開発（XDDP）	MILS／HILS／SILS	RAD【高速開発】	CASE	IDE【統合開発環境】	SPLE【ソフトウェアプロダクトライン開発】	ソフトウェアフレームワーク	テスト自動化	バグ分析（ODC分析）	形式検証	モデル検査	IV&V	ソフトウェア構成管理	
1													■																				
2								■					■	■								■					■						
3													■																				
4													■																				
5										■	■		■																				■
6										■	■			■	■				■	■		■	■	■									
19														■				■			■												
20														■				■			■												
21														■				■															
22				■										■				■															
23		■												■	■			■															
24				■		■		■										■		■													

［適用工程］

どの工程での取り組みかを示す。傾向として複数工程にわたる取り組みが多く、個別工程から見た場合、設計系の適用事例ではより上流（超上流）から課題を認識し、検証系の適用事例ではより前工程から課題を認識していることが伺える。

［技術・手法］

全事例から抽出した技術・手法を一覧として示す。それぞれの事例では、複数の技術・手法を組み合わせて課題解決に当たっている例が多い。

表３　課題の分類と解決状況

事例から、その解決すべき課題を品質 (品質特性)・コスト・納期など、該当する項目に分類した。

No.	標題	品質モデル（ISO/IEC25000：SQuaRE）※5													アシュアランス（保証）	障害原因の分析	コスト	納期	生産性（対応時間短縮）	人材育成 意識改革	プロジェクトマネジメント	見積支援	普及促進	体制（強化・再構築）	グローバル展開
		システム / ソフトウェア製品品質								利用時の品質																
		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	有効性	効率性	満足性	リスク回避性	利用状況網羅性												
		機能完全性　機能正確性　機能適切性	時間効率性　資源効率性　容量満足性	共存性　相互運用性	適切度認識性　習得性　運用操作性　ユーザエラー防止性　ユーザインタフェース美性　アクセシビリティ	成熟性　可用性　障害許容性　回復性	機密性　インテグリティ　責任追跡性　真正性　否認防止性	モジュール性　再利用性　修正性　試験性　解析性	適応性　設置性　置換性	有効性	効率性	実用性　信用性　快感性　快適性	経済リスク緩和性　健康・安全リスク緩和性　環境リスク緩和性	利用状況完全性　柔軟性												
1	BPM をベースにした会社統合での新業務プロセス設計の適用事例			□↓■				□↓■									□↓■								□↓■
2	人間系プロセスを含む業務を BPM ソフトウェア活用により改善した事例		□↓■	□↓■	□↓■		□↓■	□↓■	□↓■		□↓■	□↓■					□↓■		□↓■							
3	D-Case導入によるシミュレーションS／Wの期待結果明確化と合意形成	□↓■				□↓■		□↓■									□↓■		□↓■	□↓■	■					
4	ソニーの電子お薬手帳システム「harmo」に適用したセキュリティ設計分析手法						□↓■						□↓■		□↓■				□↓■							
5	XDDP におけるデグレード防止効果を高めるための手法ー『気づきナビ』の考案ー							□↓■							□↓■				↓■							
6	組込システムのモデルベース開発適用における DI コンテナの活用		□↓■			□↓■		□↓■	□↓■								□↓■	□↓■	□↓■				□↓■	□↓■		
7	大規模システムへのモデルベース開発手法の適用		↓■					□↓■									□↓■		□↓■							
15	Friendly による内部 API を使ったシステムテスト自動化	□↓■															↓■	□↓■								
16	アジャイルプロセスにおける実践的な品質向上施策の適用事例	□↓■															□↓■	□↓■								
17	メトリクス分析手法を用いた試験品質向上の取組み	□↓■														□↓■										
18	ユーザーエクスペリエンスを業務に定着化させるための取り組み事例の紹介									□↓■		□↓■								↓■			□↓■	□↓■		
19	パッケージ開発プロセス改善による品質向上と生産性向上	□↓■				↓■		□↓											□↓■							
20	Web システムにおける単体テストの品質向上の取組み	↓■						□↓■									□↓■		↓■							
21	安心なサービスの品質改善を実現する為の継続的システムテスト	□↓■						□↓■										□↓■								
22	セキュア開発手法の考察と診断ツールの活用事例の紹介～お客様に「安心してご利用ください」と言えるための脆弱性対策～						□↓■										□↓■	□↓■	□↓■	□↓■						
23	モデル検査とテストによる車載オペレーティングシステムの検証	□↓■				□↓■									□↓■											
24	モデルベース開発とコード解析を用いた組込みソフトウェアの開発	□↓■															□↓■		↓■							

凡例：
□：課題、計画時の目的として取り上げた項目
■：取り組みの結果、効果が得られた項目
□→■：課題解決に効果があった項目
□→：課題解決に効果がでなかった項目
→■：計画時の課題以外で効果があった項目

【脚注】
※ 5　JIS X25010:2013（ISO/IEC 25010:2011）に基づく

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目的で、「ソフトウェア工学分野の先導的研究支援事業」を 2012 年度より実施している。2015 年度は大学・公的研究機関から研究提案を広く公募した結果、19 件の応募があり、このうち 6 件を採択した。また、2014 年度に完了した研究 4 件については IPA/SEC の Web ページでその成果を公開した。この中には研究成果であるシステム開発支援ツールを大学でフリーウェアとして公開しているものもある。本稿では 2014 年度に完了した研究成果と、2015 年度事業の公募における採択状況について報告する。本事業は 2017 年度までの継続を予定しており、次年度以降も公募を実施する予定である。

1. 本事業の概要

ソフトウェアは、あらゆる産業や市民生活を支える基盤として不可欠な存在となってきており、複雑化・大規模化するソフトウェアの高信頼化や開発プロセスの高度化、その運用や保守についても様々な課題が存在している。また、システム同士を組み合わせて新しいシステムやサービスを開発し提供する場面が増えてきているが、ここでも開発のためのアプローチやシステムの信頼性確保のための課題が存在している。

これらの課題に対して工学的なアプローチで解決策を提供しようとするソフトウェア工学や複雑な統合システム（System of Systems）へのシステム工学の適用にかかわる研究や、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促進をねらいとして本事業を実施している。

本事業では、研究内容の新規性・独自性だけでなく、研究成果の産業界への展開も重視している。IPA/SEC では産業界の有識者から成る「ソフトウェア工学研究推進委員会」を設置し、同委員会により公募内容を決定し、研究提案の選考と研究に対する助言も行いながら実施している。

表 1　2014 年度に完了した研究

期間	区分*	研究テーマ名	提案者名
2年	A	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	国立大学法人 奈良先端科学技術大学院大学
2年	A	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 ―次世代の定量的プロジェクト管理ツールとリポジトリマイニング研究基盤―	国立大学法人 和歌山大学
2年	B	形式仕様とテスト生成の部分的・段階的な活用 ～探索を通したコード中心インクリメンタル型開発の支援	大学共同利用機関法人 情報・システム研究機構
1年	B	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	学校法人 芝浦工業大学

* 公募した研究分野で A 区分は「ソフトウェア工学分野の先導的な研究」、B 区分は「ソフトウェア開発現場へのソフトウェア工学の適用に関する研究」

2. 2014 年度に完了した研究の成果

2014 年度に完了した研究は、2013 年度に採択した 2 年度にまたがる研究期間の 3 件と、2014 年度に採択した単年度の研究期間の 1 件である（表 1 参照）。

これらの 4 件の成果報告書を IPA/SEC が公開すると共に、委託研究先である和歌山大学はソフトウェア開発現場で利用可能な支援ツールをフリーウェアとして大学の Web ページで公開した。成果報告書のダウンロード及びフリーウェアの紹介ページへのリンクは以下の URL を参照いただきたい。

http://www.ipa.go.jp/sec/reports/20150417.html

それぞれの研究成果の概要を以下に示す。

◎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（国立大学法人奈良先端科学技術大学院大学）

本研究は、2012 年度に本事業に採択された内容を発展させたものである。2012 年度の委託研究では、ソフトウェア品質の第三者評価の技術基盤の確立を目指し、「ソフトウェアプロジェクトトモグラフィ」と呼ばれる新しい概念・手法を提案した。2013 年度の研究では、研究成果をソフトウェアの品質評価方法、プロジェクトデータの解析やその可視化方法を高度化し、その妥当性・有用性をプロトタイプシステムの実装と実証実験を通じて示した。本手法では、プロジェクトを様々な観点で可視化し、ソフトウェアやその品質が実現される過程（プロセス）を表すことが可能となる。これによりソフトウェア品質の第三者評価で必要となる「ソフトウェアプロジェクトデータの提供」及び「提供されたデータに基づくプロジェクト理解」を容易に行うことができる。

◎ 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 ―次世代の定量的プロジェクト管理ツールとリポジトリマイニング研究基盤―（国立大学法人和歌山大学）

定量的なプロジェクト管理の普及のため、IPA/SEC の成果物である「定量的プロジェクト管理ツール（EPM-X）」の機能を拡張し、「リポジトリマイニング（品質予測・工数予測など）」及び「プロアクティブマイニング（異常・予兆の検出など）」を支援するツール（プラグイン）を作成した。これにより、これまで勘や経験に頼りがちであったソフトウェア開発を、客観的・定量的な形で行うと共に、プロジェクト内の異変や問題発生の予兆をリアルタイムに検出することで、プロジェクトや管理の見直しのための定量的な材料にすることができる。研究成果である機能拡張プラグインは和歌山大学よりフリーウェアで公開している。

http://oss.sys.wakayama-u.ac.jp/msr/

◎形式仕様とテスト生成の部分的・段階的な活用 ～探索を通したコード中心インクリメンタル型開発の支援（大学共同利用機関法人情報・システム研究機構）

アジャイル開発、形式手法、品質保証テストの３つの技術分野において、それぞれの特徴に起因する課題や、これらの相互補完や総合的な施策に対応するため、コードスケルトン（変数定義やメソッドシグネチャ定義）上に書き加えた断片的な仕様やテスト設計を基に、テストケースを探索、提示するツールを構築する。これにより、形式手法や品質保証のテスト生成ツールで求められる厳密・十分な記述を段階的かつフィードバック付きで行えるようになり、また、仕様やテストケース間の関係、基礎技術について、３つの技術分野にまたがった総合的な考えを行うきっかけになることが期待できる。

◎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（学校法人芝浦工業大学）

システム開発時に、そのシステムが当初の「仕様」を満たしているかどうかの確認作業を高効率・高品質に実施することを目的に、モデル検査を用いたソースコード検証を、形式手法のスペシャリストでなくても実現できるような検査方法の構築に取り組んだ。これにより、仕様を満たしていることを低コストで確認でき、手戻りの少ない開発に貢献することが期待できる。

3. 2015 年度公募の状況と採択結果

2015 年度の公募に際しては、産学連携の一層の推進のため、関係業界団体からのニーズを踏まえ、新たに 4 つの研究区分を設定し、合計８区分とした（表２参照）。

2015 年度の公募は前年並みの 19 件の応募があった。これらの提案については、ソフトウェア工学研究推進委員会において厳正な審査を行い、６件の研究提案の採択を決定した（表３参照）。このうち３件が業界ニーズを踏まえて新たに設定した研究テーマによる提案となった。

表２　公募した研究区分

区分	区分名	対象となる研究分野の概要
A	ソフトウェア工学分野の先導的な研究	要求工学、プロセス改善、高信頼性、アジャイル開発、形式手法、モデルベース開発などのソフトウェア工学分野の先導的な研究
B	ソフトウェア工学・システム工学の実践的な適用に関する研究	ソフトウェア開発現場への適用を目的としたソフトウェア工学の成果・手法を詳細化・具体化・実用化する研究またはスマートコミュニティ、ヘルスケア、ロボット、次世代自動車と交通システム等の複雑な統合システム（System of Systems）の研究開発において、ソフトウェア工学・システム工学の成果・手法を適用する研究
C	ソフトウェアが経済社会にもたらす革新的効果に関する実証研究	ソフトウェアが社会や組織経営にもたらす経済価値、生産性向上、競争力強化、イノベーション等の経済効果についての実証研究
D-1	ソフトウェア開発データの分析	IPA/SEC が過去 10 年間にわたり収集・蓄積してきたソフトウェア開発データを新たな視点や手法により分析・研究することにより、ソフトウェア開発における課題や方向性を提唱する研究
D-2*	ソフトウェア・エンジニアリングの実践事例研究	技術シンポジウム SPES の 2011 年より 2014 年にわたる講演資料から、産業界に資する技術や課題を選定し、特定の技術や課題に関する深掘り、幅広い技術や複数の課題に関する研究
D-3*	マイグレーションの課題に関する研究	既存資産のオープン化・クラウド化やリプレース、外部の異種サービス連携における人材面・技術面の課題解決、アジャイル開発の採用による効率化などを実現する方法論など、マイグレーションを進める上での様々な課題の解決を目指す研究
D-4*	モデルベースによるリスク評価を活用したシステムの安全性や品質の向上に関する研究	複雑化するシステムにおいて、ソフトウェア中心のシステム視点からの障害リスク検証を進めるため、システム全体の振る舞いを確認しながら、かつ仮想的に動作検証可能なモデルベースアプローチを利用することで、システムの安全性や品質を向上させることを目指す研究
D-5*	ソフトウェアの総合的品質指標の設定とその実証的評価	ソフトウェアの品質について、プロセス、不具合量、保守性、拡張性など、様々な要素をもとに総合的に評価する指標を設定するとともに、それを実データに基づいて評価する研究。また、その指標および指標を構成する個別要素と、顧客満足度との相関関係を調べ、更にその関係の時代変遷などについても考証する研究

*2015 年度に新たに設定された研究区分

表３　2015 年度採択研究提案一覧

期間	区分	研究テーマ名	提案者名
１年	B	保証ケース作成支援方式の研究	国立大学法人 名古屋大学
１年	C	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	国立大学法人 福井大学
２年	D-2	要求定義の高品質化のための要求仕様の検証知識の形式知化と一貫性検証支援ツールの開発	学校法人 工学院大学
２年	D-4	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	国立大学法人 広島大学
２年	B	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	国立大学法人 電気通信大学
２年	D-5	ソフトウェア製品群の測定評価と分析による製品品質の実態定量化及び総合的品質評価枠組みの確立	学校法人 早稲田大学

SWEBOK V3.0 日本語訳版[※1]の連続紹介 −3の2

新谷 IT コンサルティング
新谷　勝利

1．はじめに

前回の SWEBOK V3.0 連続紹介の 3 の 1 では以下について紹介した。

1）1968 年の NATO 会議の Software Engineering というタイトルの報告書[※2]
2）2001 年 5 月発行の SWEBOK Trial Version[※3]
3）2005 年 6 月発行のソフトウェアエンジニアリング基礎知識体系—SWEBOK 2004 −日本語訳版[※4]

今回は、SWEBOK V3.0 日本語訳版を紹介してゆく。この紹介にあたり、原文の発行元の IEEE 及び日本語訳の発行元のオーム社から引用をこころよく承諾していただいたことに感謝する。

2．V3.0 の構造

V3.0 は、以下の章から構成される。章名の右端の＊印は SWEBOK 2004 に存在していたことを示す。ただし、第 1 章から第 8 章及び第 10 章は、章名が同じであるからといって内容まで全く同じではない。違いは後で示すが、副知識、及びトピックスを見るとはっきりする。第 9 章は、前版では、「ソフトウェアエンジニアリングのためのツールおよび方法」と称していた。V3.0 においては、ツールは各知識領域に副知識領域として集約されている。第 11 章以降は、第 12 章はトピックスとしても新規に追加され新しい章として、その他の章は大幅に内容が拡張され章として独立している。

第 1 章　ソフトウェア要求＊
第 2 章　ソフトウェア設計＊
第 3 章　ソフトウェア構築＊
第 4 章　ソフトウェアテスティング＊
第 5 章　ソフトウェア保守＊
第 6 章　ソフトウェア構成管理＊
第 7 章　ソフトウェアエンジニアリング・マネージメント＊
第 8 章　ソフトウェアエンジニアリングプロセス＊
第 9 章　ソフトウェアエンジニアリングモデルおよび方法
第 10 章 ソフトウェア品質＊
第 11 章 ソフトウェアエンジニアリング専門技術者実践規律
第 12 章 ソフトウェアエンジニアリング経済学
第 13 章 計算基礎
第 14 章 数学基礎
第 15 章 エンジニアリング基礎

SWEBOK V3.0 では、数学を始めとする基礎知識の重要性を強調している。これは近年ソフトウェアが社会及び企業・組織に与える影響が大きくなっていることから、その開発に関与するエンジニアが実践するべき規律に関する認識を高めるのみならず、従来定義されていた基礎知識と合わせ以下を達成することを目的としている。

1）　全世界に、ソフトウェア・エンジニアリングに対する一貫した大局観を普及・促進する。
2）　他の体系化された実践規律、例えばコンピューターサイエンス、プロジェクト管理、コンピュータエンジニアリング、及び数学に対して、ソフトウェア・エンジニアリングの範囲と位置付けを詳細に示す。
3）　ソフトウェア・エンジニアリング実践規律の内容を、特徴づけして示す。
4）　ソフトウェア・エンジニアリング知識体系に対して、トピックスを通したアクセスを提供する。

【脚注】

※ 1　松本吉弘訳、ソフトウェアエンジニアリング基礎知識体系－SWEBOK V3.0 －、オーム社、2014 年 11 月 25 日、ISBN978-4-274-50521-8
※ 2　Software Engineering, Report on a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th Octo., 1968、http://homepages.cs.ncl.ac.uk/brian.randell/NATO/
※ 3　http://cisas.unipd.it/didactics/STS_school/Software_development/Trial_Version1_00_SWEBOK_Guide.pdf
※ 4　http://www.computer.org/portal/web/swebok/2004guide;jsessionid=dba3ca44da150499851e19bd752a

5） カリキュラム開発、個人の認証、及び免許取得に必要な基礎を提供する。

これらの目的達成のために、基礎知識体系として、以下の階層構造で定義されている。

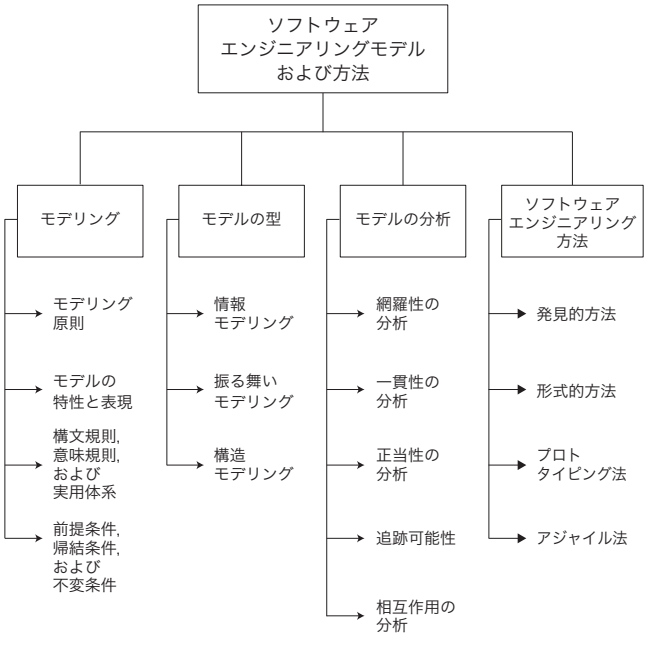
1） 知識領域―前述の章で示されるもの
2） 副知識領域
3） トピックス

今回の連続紹介３の２においては、先ず追加された知識領域の階層構造を紹介することにより、何が従来に増してソフトウェアエンジニアに知識獲得が期待されているかを紹介する。

3. 第9章　ソフトウェアエンジニアリングモデル及び方法

当知識領域の副知識領域は、以下の構造図における第一レベルの分割されたものである。

1） モデリング
2） モデルの型
3） モデルの分析
4） ソフトウェア・エンジニアリング方法



各副知識領域は、関連する参考文献において紹介されているトピックスをカバーしている。各章の最後に、「トピックスと参照資料の対照表」（構造図に次いで示す）及び「参照資料」のリストが準備されており、読者は更なる学習ができるようになっている。第９章の参考資料は７つ引用されているが、残念ながら日本語に翻訳されているのは 2015 年５月現在 Sommerville の「ソフトウェアエンジニアリング」が事例としてあるが、SWEBOK V3.0 に引用されている 2011 年版ではなく、1993 年版であり、これでは参照資料にはならない。以降、各章における参照資料はほとんど翻訳されていない。

トピックスと参照資料の対照表

	Budgen 2003 [1*]	Mellor and Balcer 2002 [2*]	Sommerville 2011 [3*]	Page-Jones 1999 [4*]	Wing 1990 [5*]	Brookshear 2008 [6*]	Boehm and Turner 2003 [7*]
1 モデリング							
1.1 モデリング原則	c2s2. c5s1. c5s2	c2s2	c5s0				
1.2 モデルの特性と表現	c5s2. c5s3		c4s1.1p7. c4s6p3. c5s0p3				
1.3 構文規則、意味規則、および実用体系		c2s2.2.2 p6	c5s0				
1.4 前提条件、帰結条件、および不変条件		c4s4		c10s4p2. c10s5 p2p4			
2 モデルの型							
2.1 情報モデリング	c7s2.2		c8s1				
2.2 振る舞いモデリング	c7s2.1. c7s2.3. c7s2.4	c9s2	c5s4				
2.3 構造モデリング	c7s2.5. c7s3.1. c7s3.2		c5s3	c4			
3 モデルの分析							
3.1 網羅性の分析			c4s1.1p7. c4s6		pp8-11		
3.2 一貫性の分析			c4s1.1p7. c4s6		pp8-11		
3.3 正当性の分析					pp8-11		
3.4 追跡可能性			c4s7.1. c4s7.2				
3.5 相互作用の分析		c10.c11	c29s1.1. c29s5	c5			
4 ソフトウェアエンジニアリング方法							
4.1 発見的方法	c13.c15. c16		c2s2.2. c7s1. c5s4.1				
4.2 形式的方法	c18		c27		pp8-24		
4.3 プロトタイピング法	c12s2		c2s3.1			c7s3p5	
4.4 アジャイル法			c3			c7s3p7	c6.app. A

参照資料

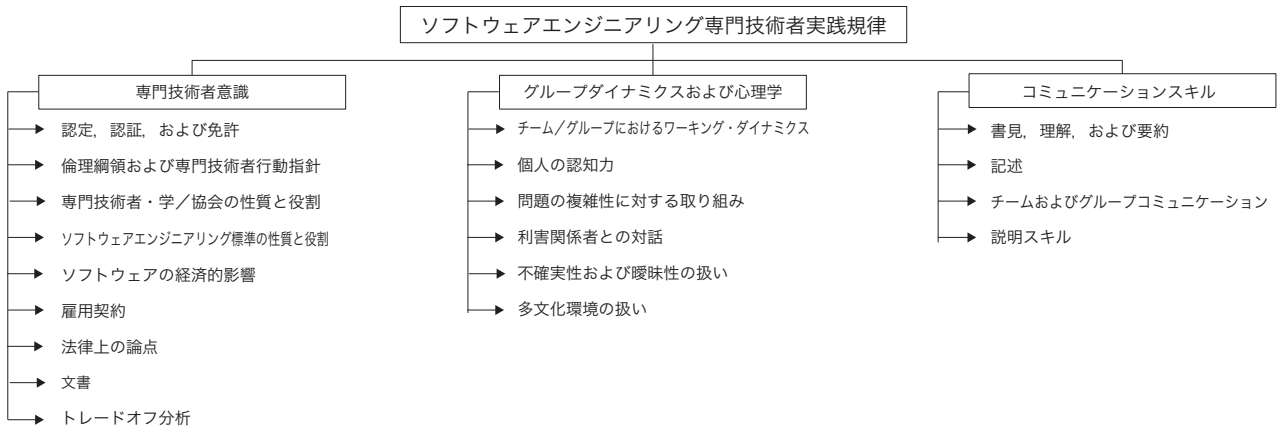
[1*] D.Budgen,Software Design,2nd ed.,Addison-Wesley,2003.
[2*] S.J.Mellor and M.J. Balcer,Executable UML:A Foundation for Model-Driven Architecture,1st ed.,Addison-Wesley,2002.
[3*] I.Sommerville,Software Engineering,9th ed.,Addison-Wesley,2011.
[4*] M.Page-Jones,Fundamentals of Object-Oriented Design in UML,1st ed.,Addison-Wesley,1999.
[5*] J.M.Wng,"A Specifier's Introduction to Formal Method s,"Computer,vol.23,no.9,1990, pp.8,10-23.
[6*] J.G.Brookshear,Computer Science:An Overview,10th ed.,Addison-Wesley,2008.
[7*] B.Boehm and R. Turner,Balancing Agility and Discipline:A Guide for the Perplexed ,Addison-Wesley,2003.

4. 第 11 章ソフトウェアエンジニアリング専門技術者実践規律から第 15 章エンジニアリング基礎の知識領域構造図

以降、新しい第 11 章から第 15 章までの 5 つの章の構造図を紹介する。

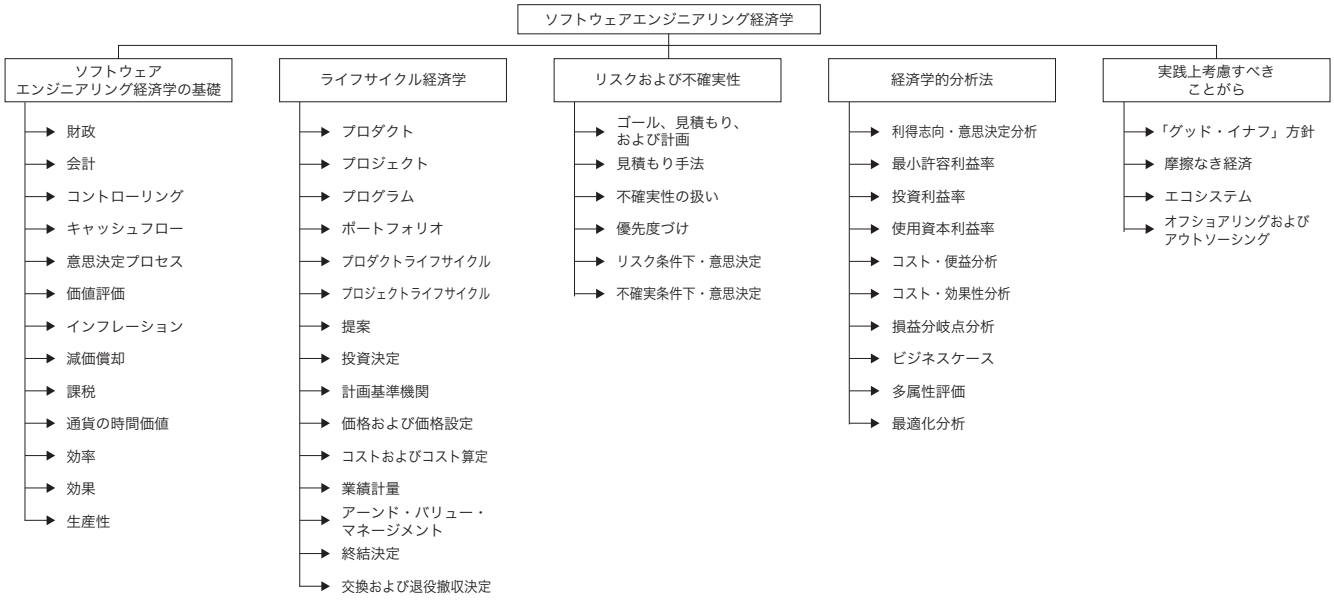
1） 第 11 章　ソフトウェアエンジニアリング専門技術者実践規律

トピックスの一つとして留意されなければならないものとして、「倫理要綱及び専門技術者行動要綱」がある。エンジニアが専門職として社会的責任を担うためには本トピックスは必須事項であろう。ソフトウェア開発には多くのステークホルダが関与し、「コミュニケーションスキル」というものが副知識領域として特出している。その他、各トピックスにはものにより副トピックスも説明されている。例えば、「法律上の論点」というトピックスには、標準、登録商標、特許、著作権、企業秘密、専門職責任、法定要求、通商規則遵守及びサイバー犯罪、の副トピックスがある。



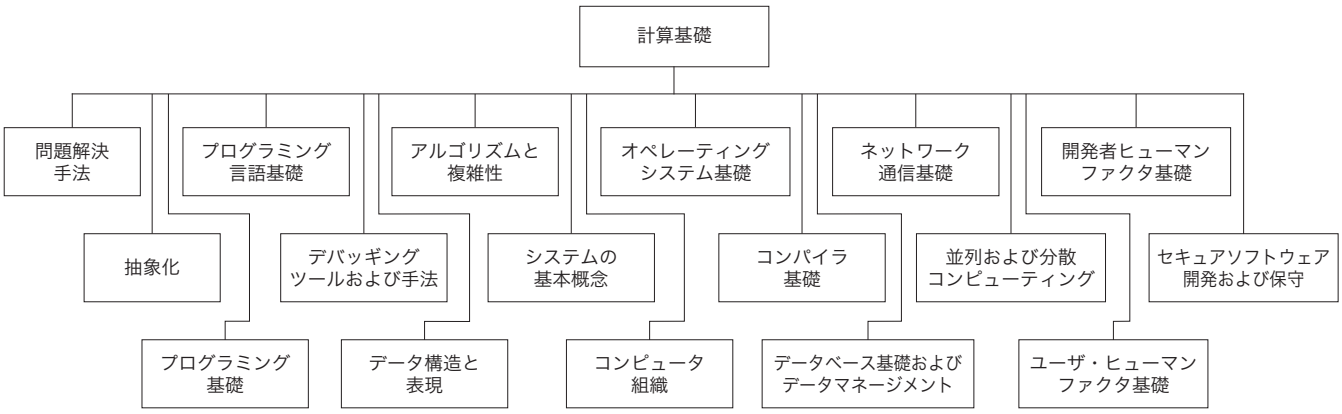
2） 第 12 章　ソフトウェアエンジニアリング経済学

とくに日本におけるソフトウェアビジネスにおける利益率は他国に比して高いとは言えない。健全な利益無しには業界としての発展は期待できず、エンジニアはともすれば技術論に関心が向きがちであるが、経済学の基礎知識を持つことも期待されている。ソフトウェアが出荷された後のみならず、出荷前までの投入労力に対して収入が発生するが、開発期間は言うに及ばず出荷後当該製品あるいはサービスが使用されなくなるまでのライフサイクルにおける経済学の知識が必要である。利益率が低い時にはなおさら、ライフサイクルを通したコスト推定と実際コストのモニタリング、リスクの事前予測及びその回避等に関わる副知識領域で述べられるトピックスが必要であり、更に「実践上考慮すべきことがら」という副知識領域では、「グッド・イナフ」の方針、摩擦なき経済、エコシステム、オフショアリング及びアウトソーシングというトピックスがカバーされている。



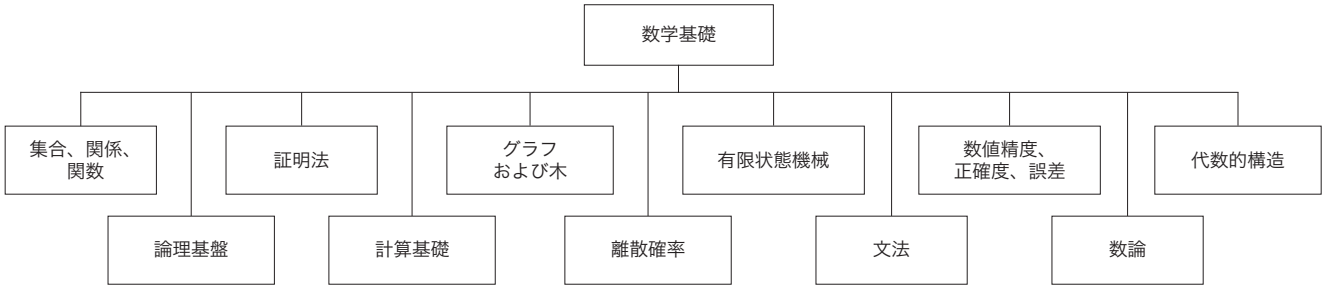
3） 第 13 章　計算基礎

この章は、コンピュータサイエンスという領域にかかわる知識領域をカバーしている。ソフトウェアはコンピュータで動くものであるが故に、効率的なソフトウェア開発にはコンピュータサイエンスの背景が必要である。例えば、抽象化という副知識領域は、抽象化水準、カプセル化、階層及び代替抽象というトピックスをカバーしている。これらトピックスは McConnell の Code Complete という大書により説明されており、幸いにして第 2 版が上下二巻で日本語に翻訳されており、良い参考文献となっている。



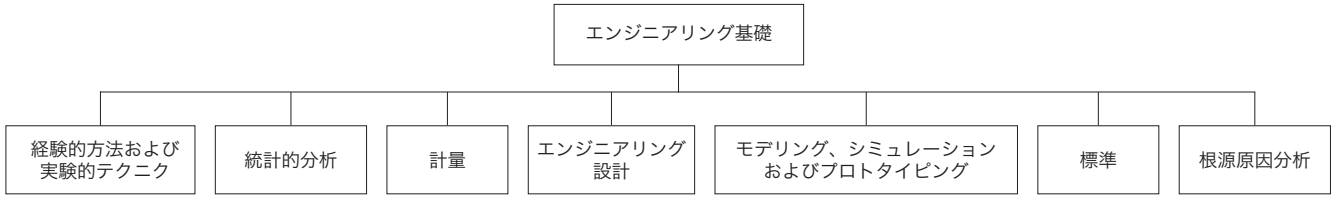
4） 第 14 章　数学基礎

今までのソフトウェアエンジニア育成において、比較的手を抜かれていたものが、数学という知識領域であろう。それもあってか、この章は比較的丁寧に各副知識領域を説明している。数学は形式手法と結びつけられて語られることがあるが、形式手法を採用しない場合でも、ソフトウェアエンジニアは精密な手段を持つことが期待されている。プログラミング言語は最終的には、極めて数学的なものであるからである。



5） 第 15 章　エンジニアリング基礎

ソフトウェア開発がエンジニアリング的に実施されるためには、対象とするシステムに体系的にアプローチし、実践規律に基づき定量可能な手法を用いることが期待されている。これは、エンジニアリングの定義そのものである。



5. 終わりに

連載 2 回目の今回は、SWEBOK V3.0 になって新しく紹介された知識領域をそれらの全体像を見る形で説明した。説明は全体を鳥瞰することを目的としており、各知識領域については「ソフトウェアエンジニアリング基礎知識体系—SWEBOK V3.0—」を本解説の読者の皆さんが読まれることを期待している。次回の連載最終回においては、従来からカバーされていた知識領域に関し、改定された部分を中心に紹介する。

Column

デジタル社会の公用語はプログラミング言語

IPA 顧問
松田　晃一

「IT 業界の公用語はプログラム言語」と題したコラムを数年前の SEC メルマガに書きました（2012 年 4 月号）。それは、こんな内容でした。

IT 業界の公用語はプログラム言語

「会社の公用語を英語にする」という企業が話題です。グローバルなビジネスを展開する企業にとっては、英語がコミュニケーションの基本となるのは必然なのでしょう。では、IT 業界はどうでしょうか？私は、IT 業界の公用語はプログラム言語にすべきだと思います。IT エンジニアは広く使われているプログラム言語を何か一つでよいから、読み書きが自由にでき、自在に操れることが必要だと思います。それは、動くプログラムを完成させるためというよりは、むしろ IT エンジニア同志のコミュニケーションの言葉としてプログラム言語を使うということです。

チームで行われるシステム開発では、自分の考えを相手に伝えたり、他人のアイデアを理解したりすることは本質的なことです。その手段として、日本語や英語はもちろん大切ですが、IT エンジニア同士のコミュニケーションにはプログラム言語がより有効だと考えるからです。日本では往々にしてプログラム言語を操る能力はプログラマに任せておけば良いと、あまり重要視されてこなかったように思います。しかも、プログラミングは若い頃の一時期にやることで、できるだけ早くシステム・エンジニアやプロジェクト・マネージャ、システム・アーキテクトへ進むのが成功のキャリアパスという風潮もそれを助長したようです。確かにプログラミングの作業はオフショアや、開発ツールの進歩によって、その役割が減ってきたことは事実です。しかし、そのこととプログラム言語能力の重要性とは無関係です。

IT エンジニア相互のコミュニケーション、とくにこれから進むグローバル化に際して海外のエンジニアと対等のコミュニケーションを図るためには、プログラム言語が自在に操れることはますます重要になってきていると思います。

こんな趣旨のコラムでしたが、今振返ってみると IT エンジニアや IT 業界に限るのではなく、もっと幅広い人々にとっての公用語にすべきだ、と思います。

デジタル社会はソフトウェアでできている

システムを求める利用者とその開発を請負う開発者の間の意思疎通が旨く行かずに、システム開発が失敗する例が後を立ちません。利用者と開発者の両方に共通のプログラミング言語の素養があれば、このようなコミュニケーションギャップの問題はすぐに解消です。そもそも利用者にその能力があれば、わざわざ開発者に依頼などせずに、自分のアイデアを思う通りに自らプログラミングして直ぐに実現することができます。新しいビジネスモデル、斬新なサービスのアイデアを思い付けば、直ぐに自分で実現して見せて、世の中に問うことができるのです。

日本のお家芸の工業製品も同じです。製品の価値を決めるのは、ハードウェアではなく、ソフトウェアへ急速にシフトしています。いわゆるソフトウェア・デファインド・マシンです。ここでも、製品のアイデアをいかに素早くソフトウェアで実現し製品に組み込むかが勝負です。

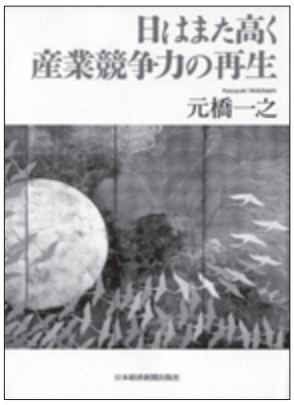
このようなイノベーションは様々な幅広い分野で起きていますが、それを本当にリードできるのは「プログラミングの専門家」ではなく、「プログラミングができるその分野の専門家」です。だからこそ、幅広い分野に携わる人達自身が自らプログラム言語を自在に操れることが、イノベーションを生む大きな力になるのです。

デジタル社会の公用語はプログラミング言語

デジタル社会の公用語であるプログラミング言語を皆が身に付けるには、初等教育からプログラミング教育を取り入れていくことが必要だと思います。欧米では既にそんな動きが始まっています。例えば、米国ではオバマ大統領をはじめ多くの著名人、企業が先頭に立って子供たちへのプログラミング教育の推進に取り組んでいるようです。日本では、英語についてやっと初等教育に取り入れる動きが始まっていますが、残念ながら日本国民の英語力は世界的に見ても非常に低いレベルにあります。プログラム言語ではその轍を踏まないように、急いで教育を始める必要があります。そして、デジタル社会の公用語を自由に操れる新しい市民を育てることが必要です。

現代のデジタル社会はソフトウェアでできています。そのデジタル社会を生きていく私たちにとって、ソフトウェアを読み書きする力を持つことは、生きる力を持つことと同じだと言うのは言い過ぎでしょうか。デジタル社会の公用語はプログラミング言語です。

「サイエンス経済」に向けて戦略の舵を切れ！



日はまた高く 産業競争力の再生

元橋一之　著

ISBN: 978-4532355906
日本経済新聞出版社刊
四六版・318 頁
定価 2,200 円（税抜）
2014 年 2 月 25 日刊

著者はサイエンス経済にいかに対応できるかを歴史データを用いて説明している。以下に本書における説明の一部を紹介する。

スイス IMD による「世界競争力年鑑」によると日本の国際競争力は 1993 年までは 1 位でその後低下し 2013 年には 24 位。なぜこうなったかについては戦略論で著名なマイケル・ポーター教授は、1）日本が競争力の源泉としてきたオペレーショナル能力の優位性が薄れた、2）変化に対するスピーディな経営戦略が欠如、としている。経済成長モデルは、1700 年までの産業革命期、1980 年までのサイエンス革命前、1990 年以降の「サイエンス経済期」と変化が見られる。

1990 年代以降、IT 投資が企業の生産性を向上させるという分析結果が公表され、とくに米国においては IT により経済構造が大きく変わったとするニューエコノミー論も発表され、インターネット更にはビッグデータの普及が特記されている。

「サイエンス経済」とは、IT 革命やライフサイエンス革命によりもたらされる経済やビジネスという社会現象に限定せずに世の中で起きている原理について明らかにし、様々な事象に応用可能な理論体系であり、これに基づく知的生産活動を競争力の源泉とするものである。ここにおいて、イノベーションが生まれるための技術プラットフォームが形成される。

「サイエンス経済」においては、その競争力の源泉は人材であり、この人材は「既存の技術を活用して新しいビジネスモデルを組み立てるクリエイティビティの高い者である。この人材を一企業内のみにではなくオープンにかつレバレッジが得られるように求める必要がある。

（新谷　勝利）

人はなぜ誤った意思決定をするのか

失敗は「そこ」から はじまる

フランチェスカ・ジーノ　著
柴田裕之　訳

ISBN: 978-4478025383
ダイヤモンド社刊
四六版・356 頁
定価 1,800 円（税抜）
2015 年 1 月 17 日刊

著者はハーバード・ビジネススクール経営学准教授にして、経済学・経営学・交渉学・社会心理学・行動経済学・組織行動学など多彩な研究を行う研究者である。組織行動や意思決定の知見をもとに、企業や非営利団体のコンサルタントとしても活躍している。

本書は誤った意思決定により生じた大手企業の失敗事例を紹介している。企業は顧客のためを想い十分な検討・準備をして行動を起こす。しかしながら、顧客が予想外の行動をとり施策は失敗に終わってしまう。顧客がなぜこのような行動をとるのか、日常の意思決定において何に影響されるのか、その行動心理を多くの実験結果から明らかにし解説している。

一例をあげると「社会的比較は、人々の心に不快感や嫉妬心といった感情を引き起こし、素直な判断を阻害する。」という行動の実験として、「仲間内の投票で最も多く支持された者が表彰されるという実験で、賞にふさわしい仲間がいたとしても、人々の心には仲間が受賞している姿を思い浮かべると嫉妬心により、その仲間を推薦する可能性が低くなる」ということを比較実験により明らかにしている。顧客がこのような行動をとることをあらかじめ理解・予測し、企業側は自らの意思決定のインプットにする必要があるとしている。

IoT の時代、大規模で複雑なシステムを開発・運用するには従来のソフトウェアやハードウェアといった視点だけではなく、人や社会までを含めて捉える必要がある。人がどう行動するか、社会がどう反応するか、そのために意思決定するべきポイントは何かを紹介している本書は技術者にも手に取ってもらいたい一冊である。

（遠藤　秀則）

編集後記

今号は 2014 年度の SEC 活動概要を特集しています。2014 年度は IPA の第三期中期計画の 2 年度目にあたります。前年度スタートさせた取り組みを確実なものにすると共に、「つながる世界」において、利用者が安全・安心にシステムやサービスを利用できるよう、それらを提供する事業者に向けた取り組みもスタートさせました。詳細については本誌特集記事をご覧ください。

これまで国内外を問わず数々のソフトウェアに関する有識者を迎えて実施してきた「所長対談」も、今年で 10 年目を迎え、この節目に初の公開形式での対談を行いました。2015 年 2 月に開催した SEC 特別セミナー「IoT 時代のソフトウェア・エンジニアリングとビジネスイノベーション」会場にて、ドイツ フラウンホーファー研究機構 実験的ソフトウェア工学研究所（IESE）の所長ディーター・ロンバック博士を迎えた対談の模様をお届けします。初の試みを実施するまでは、「所長対談」としてまとめることができるのか大変心配でしたが、会場からの質疑も含め本号に掲載することができました。セミナー会場にお集まりいただいた皆様にも感謝いたします。（編集長）

編集部より

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

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

SEC journal　編集委員会

ETWest2015 の様子（出展報告は次号に掲載予定です）（撮影：IPA）

SEC journal 第 11 巻第 1 号（通算 44 号）　2015 年 7 月 1 日発行

編集兼発行人　独立行政法人情報処理推進機構
技術本部　ソフトウェア高信頼化センター
所長　松本隆明
〒 113-6591　東京都文京区本駒込 2-28-8　文京グリーンコート センターオフィス 16 階
Tel：03-5978-7543　Fax：03-5978-7517
URL：http://www.ipa.go.jp/sec/
e-mail：sec-journal_customer@ipa.go.jp

SEC journal 論文募集

独立行政法人情報処理推進機構（IPA） 技術本部 ソフトウェア高信頼化センターでは、下記の内容で論文を募集しています。

論文テーマ

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

論文分野

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

応募要項

締切り　：1月・4月・7月・11月　各月末日
査読結果：締切り後、約1カ月で通知。「採録」と判定された論文は SEC journal に掲載されます。
応募方法：投稿は随時受付けております。応募様式など詳しくは HP をご覧ください。
http://www.ipa.go.jp/sec/secjournal/papers.html

SEC journal 論文賞

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

---

iパス ITパスポート試験 のご案内

ー ビジネスにＩＴを活用する すべての社会人のための「国家試験」ー

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

ｉパスのメリット

ＩＴを活用した業務効率化とビジネス拡大に！

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

情報セキュリティ対策・コンプライアンス強化に！

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

経営全般に関する知識など幅広い知識がバランスよく習得できる！

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

詳しくは、ｉパス Web サイトをご覧ください。https://www3.jitec.ipa.go.jp/JitesCbt/index.html
※企業の活用事例、企業の声、合格者の声など魅力的なコンテンツがご覧になれます。

IPA Better Life with IT

SEC journal No.41
第 11 巻第 1 号（通巻 44 号）
2015 年 7 月 1 日発行

ISSN 1349-8622

VEGETABLE OIL INK

R70 古紙パルプ配合率70％再生紙を使用

リサイクル適性Ⓐ この印刷物は、印刷用の紙へリサイクルできます。

SEC journal No.41

独立行政法人情報処理推進機構