

# SEC journal

44

## 巻頭言

**富田 達夫** 独立行政法人情報処理推進機構 理事長

## 所長対談

**IoT は本物か？**

坂村 健 東京大学大学院 情報学環

## 論文

**IT 組織における職場リーダー像の系統的な導出**

佐々木 康浩 株式会社三菱総合研究所 東京工業大学大学院 総合理工学研究科

吉川 厚 東京工業大学大学院 総合理工学研究科 / 寺野 隆雄 東京工業大学大学院 総合理工学研究科

## 特集

**IoT 時代に求められる開発スタイル**

**多様化が進むソフトウェア開発スタイル**

山下 博之 SEC システムグループリーダー / 室 修治 SEC 調査役

**IoT 時代におけるリーンスタートアップ**

岩崎 健一郎 H2L 株式会社 代表取締役

**変化を味方につけるアジャイル開発**

平鍋 健児 株式会社永和システムマネジメント 株式会社チェンジビジョン

**IoT の時代における派生開発の対応**

清水 吉男 株式会社システムクリエイツ 代表取締役

**ユーザーイニシアティブを可能にする超高速開発**

関 隆明 超高速開発コミュニティ 会長 一般社団法人 ICT 経営パートナーズ協会 会長

## SECjournal 論文賞

SECjournal 論文賞 受賞論文発表

## 報告

CEATEC JAPAN 2015 出展報告

Embedded Technology 2015 / IoT Technology 2015 出展報告

## 技術解説

**工数に影響を与える質的変数とその影響度**

古山 恒夫 東海大学 理学部 客員教授

## 連載

**情報システムの障害状況 2015 年後半データ**

松田 晃一 IPA 顧問 / 八嶋 俊介 SEC システムグループ 主任

## 組織の活動紹介

**一般社団法人 組込みイノベーション協議会 活動紹介**

鶴保 征城 一般社団法人 組込みイノベーション協議会 代表理事・理事長

**組込みシステムの新たな展開に向けた提言**

河村 建夫 組込み IoT イノベーション議員連盟 会長 衆議院議員

## Column

「人工知能の未来」

巻頭言 ……1

情報処理推進機構理事長に就任して

富田 達夫 独立行政法人情報処理推進機構 理事長

所長対談 ……2

IoTは本物か？

坂村 健 東京大学大学院 情報学環

論文 ……8

IT組織における職場リーダー像の系統的な導出

佐々木 康浩、吉川 厚、寺野 隆雄

特集：IoT時代に求められる開発スタイル

多様化が進むソフトウェア開発スタイル ……16

山下 博之 SECシステムグループリーダー／室 修治 SEC調査役

IoT時代におけるリーンスタートアップ ……18

岩崎 健一郎 H2L株式会社 代表取締役

変化を味方につけるアジャイル開発 ……22

平鍋 健児 株式会社永和システムマネジメント、株式会社チェンジビジョン

IoTの時代における派生開発の対応 ……26

清水 吉男 株式会社システムクリエイツ 代表取締役

ユーザーイニシアティブを可能にする超高速開発 ……30

関 隆明 超高速開発コミュニティ 会長、一般社団法人 ICT経営パートナーズ協会 会長

SECjournal 論文賞 ……34

SECjournal論文賞 受賞論文発表

報告 ……36

CEATEC JAPAN 2015 出展報告

荒川 明夫 SEC企画グループ主任

Embedded Technology 2015/IoT Technology 2015 出展報告

荒川 明夫 SEC企画グループ主任

技術解説 ……40

工数に影響を与える質的変数とその影響度

古山 恒夫 東海大学 理学部 客員教授

連載 ……48

情報システムの障害状況2015年後半データ

松田 晃一 IPA 顧問／八嶋 俊介 SECシステムグループ主任

組織の活動紹介 ……54

一般社団法人 組込みイノベーション協議会 活動紹介

鶴保 征城 一般社団法人 組込みイノベーション協議会 代表理事・理事長

組込みシステムの新たな展開に向けた提言

河村 建夫 組込みIoTイノベーション議員連盟 会長、衆議院議員

Column ……59

「人工知能の未来」

鶴保 征城 IPA 顧問、学校法人・専門学校 HAL 東京 校長

書籍紹介 ……60

編集後記 ……62

SECjournal 論文募集 / 国家試験 エンベデッドシステムスペシャリスト試験のご案内

# 情報処理推進機構理事長に 就任して



独立行政法人情報処理推進機構 理事長

富田 達夫

この度情報処理推進機構の理事長を拝命いたしました。IPAはソフトウェア産業にとどまらず、IT社会全体の発展に取り組んでいる、日本のIT社会推進のための中心的な機関です。大変な重責となりますが、私のこれまでの産業界や学界での経験を活かしながら全力で職責を果たしていきたいと思っております。

## SEC 活動への期待

いま世の中では、様々な機器がインターネットにつながり、情報をやりとりするIoT化が進んでいます。このような世界においては、これまで結びつきのなかった機器やシステムなどが互いにつながり連携することで、新しい価値やサービスが創造され、誰もが手軽にその恩恵を得ることができるようになります。その反面、ひとたび機器に障害や不具合が発生するとその影響まで他の機器に伝搬して被害が大きくなってしまおうという懸念もあり、これまで以上に信頼性・安全性の確保が求められています。

このようなIoT化が進む世の中で安心・安全な生活を送れるよう、IPA技術本部ソフトウェア高信頼化センター（SEC）では、情報機器やシステムの“信頼性（セーフティ）”向上への取り組み、また、つながることを考慮した“セキュリティ”向上への取り組みなど、様々な面から、国民の皆様へ安心・安全をお届けするための活動を行っています。様々なものやサービスがつながる動きは今後ますます進むと考えられ、SECの役割は更に重要なものになると確信しています。

## IoT時代に向けて

あらゆるものがソフトウェアで制御されて動いてい

ます。それぞれの製品やサービスの機能・性能の向上に伴い、ソフトウェアの大規模化・複雑化が進むと共に、ビジネス面では開発期間の更なる短期化が求められています。こうした状況を踏まえ、今後はより一層、エンジニアの活躍の場は広がっていくと考えます。しかしながら、大学では情報分野の学生離れも目に付き、大変危惧される所です。あまりにも身近になったITは、もはや研究の対象ではなく、コモディティ化したツールと受け止められ、また、いわゆる7K問題のようにエンジニアの労働環境に悪いイメージを持ってしまっていることも原因なのかもしれません。しかし、このIoT化の流れは、第4次産業革命とも呼ばれるように世の中を大きく変え得るエポックであり、情報処理技術の進展なくしてはなし得ないものです。情報処理技術の進展が自ら作り出した課題も含め、更に取り組む研究テーマは幅広くあります。また、このIoT時代に新たなビジネスをITを活用しながら創っていくためのソフトウェア開発には大きな可能性が広がっています。SECでは、未来の技術や社会を見据えつつ、産業界と学界の知見を結集して、社会課題の解決や経済成長に貢献する取り組みを加速させます。SECの活動を通じて情報分野におけるエンジニアの魅力ある姿を伝えたいと思っております。

IPAでは、SECの活動と共に、情報セキュリティ、人材育成などの業務と合わせて、健全な情報社会の発展に尽力していきたいと考えております。

今後も皆様のご理解とご協力をお願い申し上げます。

# IoT は本物か？

東京大学大学院 情報学環

坂村 健



SEC 所長

松本 隆明

IoTという言葉があらゆるところで使われるようになってきている。確かにIoTを活用すれば私たちの生活は非常に便利になるという期待もあるが、他方で、今は絶頂期でもやがて幻滅期に入るのではないかという見方もある。30年以上前からIoT時代の到来を予見してユビキタス・コンピューティングを提唱してきた坂村健先生に、IoTの今後の方向性や可能性についてお話を伺った。

## TRON プロジェクトに集まる世界の評価

**松本：**先生は昨年、ジュネーブの国際電気通信連合（ITU）本部で開催されたITU150周年記念式典で「ITU150周年賞」を受賞されました。おめでとうございます。

**坂村：**ありがとうございます。



**坂村 健** (さかもら けん)

1951年生まれ。東京大学大学院情報学環教授、ユビキタス情報社会基盤研究センター長、工学博士。オープンなコンピュータアーキテクチャTRONを構築、デジカメ、携帯電話、家電などの組込OSとして世界中で多数使われている。ユビキタス社会実現のためIoTの研究を推進。2002年よりYRPユビキタス・ネットワーク研究センター所長を兼任。著書に『ユビキタスとは何か』、『変わる国、日本へ』、『不完全な時代』、『毛沢東の赤ワイン』、『コンピューターがネットと出会ったら』など多数。2003年紫綬褒章、2006年日本学士院賞、2015年ITU150Award。

**松本：**ITによって人々の生活向上に貢献したとして、ビル・ゲイツ氏などと共に世界で6名のひとり選ばれたわけですが、どんな感想をお持ちですか。

**坂村：**大変光栄なことで、TRONプロジェクトを支えてくださった多くの方に感謝したいと思っています。そもそもTRONプロジェクトは、30年余り前の1984年に始めたものです。勘違されている方もいるのですが、これは政府のプロジェクトではなく完全なドネーション、つまり私の考えに賛同して下さる方の寄付で成り立ってきたプロジェクトです。そういう方々のサポートがなければ、プロジェクトは始められなかったし、継続することもできなかった。30年間にわたってTRONを支えて下さったみなさんに心から感謝したいと思っています。

**松本：**先生が非常に早い時点で組込みシステムに着目され、独自のユビキタス・コンピューティングを提唱され開発されてきたことに改めて驚くのですが、当時はどういうことを考えていたのですか？

**坂村：**申し上げたように、私は1984年から組込みに注目していました。当時、大型コンピュータが米国のイニシアティブのもとに動いていましたから、新しいことをするには、まだ誰もイニシアティブが取れていない分野でやるべきではないかと考えたからです。当時はまだマイクロソフトもAppleも有名ではなく、大型コンピュータでIBMが全世界でシェア8割を独占していた時代です。マイクロコンピュータの世界は、まだ全く誰も覇権を取っていませんでした。また、80年代は日米の国力の差もまだ大きく、アメリカに正面から勝負を挑んでもとても勝ち目はなかった。そこでアメリカがあまりやってないところをやろうということで、マイクロコンピュータに集中し、組込みコンピュータの世界を研究フィールドにしたのです。更に、その頃はまだ全くポピュラーではなかったのですが、オープンアーキテクチャという、すべての技術情報を公開して、しかも私の場合にはフリーで出す、という哲学に基づいて研究を遂行してきたことが、良かったのではないかとと思っています。

**松本：**当時はオープンにすること自体、かなり抵抗があったと思います。

**坂村：**国策のプロジェクトですべてフリーなどということは到底考えられませんでした。ですから民間プロジェクトでやらざるを得なかった。1980年代というのは、そういう時代でした。

**松本：**民間でも、それなりのビジネスとして成り立たせないといけない。やはりクローズにしたがる傾向がありませんか。

**坂村：**もちろんあります。しかし、企業は常にあらゆる可能性を考えていますから、その可能性の一つとして、私の

言っていることに説得力があったのではないかと思います。基本的なインフラストラクチャーのところでお金を取るべきではないという考え方に関しては、企業を含めその当ても賛同して下さる方は多かった。とくに半導体を中心とした新しい分野のエンジニアは、多くの方が私のプロジェクトに賛同してくださったんです。それから、そのときにもう一つ私がしたことは、未来のビジョンを描くということです。単に、その当時のリアルタイム OS がどうあるべきかではなくて、もっと半導体の能力が上がっていった場合には、こういうことができるとか、ああいうことができるとか、それを色々と本に書いたりしていた。実は、ITU150 年記念で、いわば「IoT の元祖」として高く評価していただいた理由のひとつは、1989 年にヨーロッパのシュプリング・フェアラークという出版社から、TRON 構想を書いた英語の本を一冊出していたからです。その中に、色々なものがネットにつながって相互に協調動作し合いながら、色々なシステムを作っていくという分散型システムが、マイクロコンピュータの発展と共にできるだろう、ということを書いていた。それが、欧米では広く知られていたからなんです。残念ながら、日本ではそれほど認知されていないのですが（笑）。

**松本：**先生が考案されていたユビキタスの世界というのは、正に今の IoT の世界ですね。

**坂村：**ITU の受賞式でも ITU 事務局長や関係者の多くが「IoT がここまで来ているのは、坂村が哲学的構想とビジョンと、それから TRON で、今でいうネットにつなげる色々なものをやろうとしていたからだ」と言ってくれました。

## 多くの家電やデジタルカメラ、H2A ロケットまで

**松本：**実際、今、私たちの身のまわりにあるものの多くが TRON でコントロールされていますね。

**坂村：**家電製品など色々なものに入っていて、デジタルカメラでは 80% くらい、車のエンジンのコントロールとか、はやぶさとか H2A ロケットとか、そういうものにもたくさん入っています。日本だけでなく全世界で使われているので、知名度が高い組込みシステムになっています。TRON の本も、世界中の言葉に翻訳されていて、中国語訳も韓国語訳もありますし、最近ではベトナムやインドネシア、マレーシアなどにも。TRON が世界のどこからロードされているかということは、当然インターネットでわかるわけですが、最近ではアフリカがかなり増えています。一体何に使っているんですかと聞くと、やはり携帯電話に使いたいという答えが多い。とくに開発途上国ではそうです。そうした国では、今のスマートフォンは高額でとても買えない。それを国民に持たせることはできないんです。日本でも高級機は今、10 万円近くしますからね。そこで彼らは、日本の初期の携帯電話がすべて TRON で作ってあると知って、これで行こうと考えた。今、彼らが実現したいのは通信網

の確立です。別にゲームがしたいわけではありません。基本的な社会活動、経済活動のために必要なのです。

**松本：**新たに電話線を引くのは大変だけれど、携帯電話なら簡単に最低限の通信網ができますね。

**坂村：**そこで TRON が役立ちます。私もそういう人たちに技術提供することは、TRON プロジェクトの重要なミッションだと考えています。少し話が飛びますが、そもそも最先端の開発だけでは、地球は維持できません。開発途上国をどう助けるかは、今、世界の非常に重要な課題です。ところが日本は、自分達のことばかり考えている。新しいもの、新しいものと考えて、結局みんな失敗しています。現在は国連でも、ポバティ（貧困）をどう救うかということが、環境保全と並ぶくらい重要なプロジェクトになっている。私もそういうことに貢献したい、と考えているんです。おかげさまで、そういう私の研究や行動に賛同して寄付をしてくださる方が、コンピュータ業界ではないところでも大勢出てきている。非常に嬉しく、頼もしく思います。

## 今求められる「アグリゲイトコンピューティング」

**松本：**TRON のようなオープン・プラットフォームを広めていかれただけでなく、先生は 30 年も前に、当時の大型機のタイムシェアリング中心の考え方から、直接ネットを介して物と物がつながって行くような世界をコンセプトとして打ち出されたわけです。

**坂村：**はい。今はネット環境が確立していますから、クラウドを活用して更に色々なことが考えられるようになりました。もともと私は、端末は軽くして、ほとんどの高機能な処理は全部クラウド側に持っていこう、という考えなのです。今、これだけネットの環境がうまく行ったのですから、セキュリティを含め複雑な部分はトンネリングさせて、クラウド側でやろうというのが、TRON の最近のアーキテクチャです。トンネリングさせるための暗号処理やプロトコルの設計ということに重点を置いています。

この考え方なら、端末は軽



**松本 隆明**（まつもと たかあき）

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

くできる。なぜスマートフォンがここまで高くなってしまふかということ、パソコンの機能を入れてしまったからです。大体 Linux ベースのものですが、こんなことをしたら高いものになり、重くなって、10万円近くになるのは当然でしょう。しかし TRON なら、Linux に比べてメモリの消費量は何十分の一です。CPU のスピードも遅くていい。今の TRON の最先端の  $\mu$  T-Kernel という OS では、プロセスという考えはありません。すべてタスクベースです。リアルタイムのタスクベーススケジューリングだけやっていて、プロセス制御はやらない。やりたかったら、全部クラウドでやればいいわけです。だからリアルタイムのところだけに特化させている。それが、今非常に好評です。ですから例えばデジカメも、レンズだけのカメラでいいと言っているんです。撮った写真のデータはすぐネットワークに上げてしまえばいい。そうすればメモリもたくさんは要らないですよ。全部クラウドに上げてしまうんです。

**松本:** スマートフォンも、あまりにも高性能であるが故に、乗っ取られたらどうするかなど、かえってセキュリティの問題が出てきてしまっています。

**坂村:** そうです。更に、今のスマートフォンには、ものすごくたくさんのセンサが付いています。もったいないですよ。カメラはレンズだけにして、スマートフォンで見たらどうかと言っているんです。

**松本:** 逆に、スマートフォンはディスプレイだけにしてしまうということですね。ディスプレイと通信だけにします。

**坂村:** 今はどこかものの作り方が偏っています。私は、アグリゲイトコンピューティング、総体コンピューティングということを提唱して、今かなり注目されています。更にクラウドの OS については、フレーム OS ということをやっている。今までのサーバの OS とは違って、色々なところから直結してくるものを、どう連携させていくのかということに重点を置いた、今までの概念と違う OS を作るべきだということです。これもかなり注目されていて、よくリファアされています。

**松本:** それにしても、30年経って、これだけ坂村先生の研究や TRON に日が当たって来た、その大きな理由というのは何でしょうか。

**坂村:** それは、最初から言っていたことが間違っていなかったということでしょう(笑)。

**松本:** そうなんです、今、これだけ盛り上がっているというのは、通信のネットワークが広がったとか、色々な背景もあるのでしょうか。

**坂村:** 考え方が間違っていなかったというのは、大変嬉しいし、私の誇りでもあります。おっしゃるようにネットワークインフラが確実にでき上がってきたということは大きいですね。こんなものは、30年前には何もなかったのですから。インターネットもなかった。インターネットの一般開放は、1990年代ですね。私が TRON を始めた頃は、アーパネットといったものが研究されていた。実は日本でいち早くアーパネットの端末を置いたのは私の研究室なんです。東京大

学の理学部が「米国の大学からの要請で共同研究を行うためネットワークでデータの交換をやりたい。助けてくれないうか」というので、ハワイ大学とつなげたんです。私も理学部情報科学科にいましたからね。

**松本:** そうだったんですか。ネットワークの進歩もあると思いますが、それ以外では端末のコストが下がったといったことも大きいですね。

**坂村:** それはもう、マイクロプロセッサが圧倒的にポピュラーになりましたからね。

**松本:** 性能も良くなった。

**坂村:** TRON がうまくいった理由のひとつに、小さくした、ということがあります。6LoWPAN モジュールで TRON のベースのものがあります。非常に小さい物です。

**松本:** 通信機能が入っているのですか？

**坂村:** もちろん入っています。100メートルは飛びます。単三2本で1年もつ。こんなところに UNIX は入りません。

**松本:** コスト面では安くできるのですか。

**坂村:** 何個買うかです。大体、今、500円でしょうか。

**松本:** 少なくとも1ドルを切らないと、ビジネスとしては難しいという話を聞いたことがあります。

**坂村:** ものを買う人は、みんなそう言いますね。安い方がいいから。しかし、必要な機能が備わっていなければ、いくら値段が安くてもしょうがない。高かったから売れなかった、なんていうのは言いわけで、値段に見合う機能がなかったから売れなかったんです。それだけの話だと思いますね。

## 目先の利益にとらわれすぎる日本の研究開発

**松本:** 今や IoT は、幻滅期を乗り越えてある意味で安定期に入ってきたと言えないですか。

**坂村:** そう思います。しかし、そういう時にまたしても日本が乗れないわけです。理由は何かというと、地道にやらないからです。どちらかという、日本の開発会社全体が商社のようになっている。商社も必要だと思うけれど、開発会社がどんどん減ってしまっているわけだから、うまくいきません。やはり技術開発というのは、長い年月をかけ努力を続けられない限りできない。そういう意味でいうと、今日本は駄目になってきています。

どうして駄目になったかということ、これは一例ですが、電電公社を分割したからじゃないですか。原点は、そこら辺にあるのではないかと。電電公社は米国の DoD の役割をしていました。一社のまま事業をすれば、研究開発も支えられる。しかし、別会社にしてしまったらこれができない。どんどん分けていってしまうと、そこで競争するのはいいのですが、実際には競争どころか足を引っ張り合って、技術開発が停滞し、余裕がなくなって外国のものを買ってきた方が安い、ということになってしまいます。確かにビジネスだけを考えるとそうなるんです。

**松本:** 目先の利益が出るものに、すぐ飛びついてしまいま

すからね。長期的な研究開発に投資するという発想には、なかなかならない。

**坂村：**そうですね。だから駄目になって、国際標準づくりからもどんどん引き揚げてしまうし、研究投資もビジネスに近い投資しかしなくなる。昔は TRON なども「こういうのは重要だ」と思っていたからドネーションもあったけど、その後なくなった。TRON のように長期間かかって取り組む技術開発よりは、アメリカから買って来てすぐ売れるものに投資し始める。商社になってしまったら、そうですね。今から研究開発して何年か経たなければ結果が出ないというものより、買って来てすぐ売れるものの方がいいとなってしまふ。

**松本：**私自身、最近の企業の研究開発を見ていると、どうしても現場寄りの、すぐ使える技術、売りに直結する技術に走っていて、中長期的に進めたり、それをオープンにしていこうという発想がなくなっていると感じます。

**坂村：**日本の研究開発は何かが間違っている。ある程度の資金や実力のある人がいなければ、日本もここまでは来なかったとは思いますが。しかし、すべてが裏目に出ているような気がしますね。

---

## IoT の考えに基づいた未来住宅を作る

**松本：**先生は、これまで TRON 電脳住宅やトヨタ夢の住宅 PAPI などの開発を手がけられ、今度は LIXIL と組んで、IoT ハウスを作られると伺っています。住宅に注目されるのはどういう理由ですか？

**坂村：**住宅は IoT やユビキタス・コンピューティングがどういふことをやるのか、ということをつかりやすく示すことができるからです。身近にわかる IoT のショーケースとして非常にいい、と思っています。

更にいえば、実際に役に立つということです。最初の TRON 電脳住宅の時は竹中工務店グループとやり、次はトヨタグループとやりましたが、今度のパートナーである LIXIL という企業は、前の二つの企業とは性格が違って、住宅のパーツを作っている会社です。私は思うのですが、住宅はパーツで作られていますから、パーツを作っている会社が IoT という感覚を身に付けた場合、与える影響力は非常に大きいということです。

どういふことかということ、建築会社やハウスメーカーが作った場合は、すべて特注になってしまうんです。だから TRON 電脳住宅にしても PAPI にしても、非常に高かった。たくさん作らないからです。ところが LIXIL は、室内ドアだけでも、年間何十万も作って売っているわけです。トイレも何十万個という規模です。そういう企業が IoT パーツを作り始めたら、サッシだって 100 万本くらい売っているわけですから、与える影響力が全く違います。もちろんコストも下がる。それは IoT 化の、大きな起爆剤になるのではないかと思います。

LIXIL グループから IoT 部品に賭けたいと言われ、ぜひお願

いしたいと言われたので、喜んで協力することにしました。IoT の考えに基づいた未来の住宅部品を、来年立ち上げることを目標にして、今着々と研究を進めています。もちろん、私の研究所でやったものの知見もすべて入ります。この研究所でも、まだパーツを作ることはしていない。パーツを作って、そこに TRON のボードのコンピュータを入れたりするのではないんです。ソフトの研究をしたり、試作センサを付けてデータを取るといったことをやっているわけで、つまりセンサを作っているだけで扉を作っているわけではないんです。しかし、LIXIL との開発は、OS より少し上のミドルウェアレベルぐらいで、住宅パーツの中に入れてしまおうと考えてやっているわけですから非常におもしろい。コストも大きく下がるでしょう。

**松本：**今までの建設会社などではできなかったことが実現しますね。

**坂村：**コストを下げようと思ったら、部品を工業化しない限り無理なんです。その可能性が大きく開けると思いますが。今、私が興味をもっていることは、さきほどのポパティ救済だけではなく、世界の人々にどういふ影響を与えることができるかということです。その点では、部品レベルの活動をもっと強めなければうまくいかない。その意味で、LIXIL とのプロジェクトに私は非常に期待しています。

**松本：**確かに、そういう形で組み込まれたものがパーツとして世の中に出回れば、本当にそれが広がって行きますね。色々なハウスメーカーが、それを取り入れ組み合わせればいだけの話になってくる。それはわかりやすいですね、非常に。おそらく自動車も、同じような道を辿っていくんじゃないかという気がします。

---

## GPL ライセンスの考え方は取らない

**坂村：**自動車は難しい面があります。車の中身をオープンにすることはなかなかできないからです。私が今やっている TRON というのは、オープンソースから始まったのですが、そのほかに、最近私が力を入れてやっているものに、オープンデータとオープン API があります。あらゆるものをオープンにしようとしているのですが、自動車はオープン API のカテゴリーではないかと思っています。オープン API というのは、車をネットにつないだ時に、色々な人が、その API を使えるということです。つながった車を他の人にオープンにすれば、色々なことができるでしょう、ということですね。

車を作るところの中は、やはり秘密にしたいことがたくさんあるわけです。なぜトヨタの車が TRON を使うかというと、普通、オープンソースには GPL というのがあって、Linux などでは、OS を直したら、それも公開しなさいと言っています。しかし TRON ではその必要がない。TRON は、私たちがオープンにして出したものを、その後トヨタが中を直しても、公開する必要はないんです。TRON がいいと言われるのは、リアルタイム性だけでなく、ライセンス規

定が Linux と違うからです。

**松本**：普通の GPL ライセンスと違うんですね。

**坂村**：違います。GPL はユーザがプログラマという前提ではすばらしいが、組込みではエンドユーザは一般消費者でメリットがない。しかも、GPL は、組込みの場合には望ましくない。なぜかという、ノウハウは OS に近いところにたくさんあるんです。そこはメーカーは出したいところでしょう。そこを出さなければ使えないと言ったら、使わなくなる。しかし TRON はそうじゃない。

先ほどのカメラと同じことです。例えば、レンズだけのカメラを作ったとします。そこにはノウハウがある。カメラを作るパーツを全部オープンにして、カメラを誰でも作れるようにしようと思ったら、カメラメーカーは嫌だというはず。当たり前ですね。そんなことをしたら企業活動が成り立たない。私が求めているのはそういうことではなく、例えばネットを経由してシャッターが切れるとか、絞りの値がどうなっているのかがわかるとか、そういうことはオープンにしようと言っているのです。車でいえば、例えば今エンジンがどのくらいで回っているのかということ。エンジンをどうコントロールして、どう作るかは教えなくていい。ただ、今のエンジンの回転数がどうなのかとか、タイヤの位置がどう向いているのかとか、そのくらいは公開してください。そういうことがわかっただけでも、自分なりのカーナビを作ることができるようになります。私はそれを推奨しているんです。

そうしないと、例えば、エンジンのコントロールで、ガソリンの気化比をどうするということをリアルタイムで計算しているプログラムを公開してくれと言ったら、いやだって言われます。そこは今までずっと研究を重ねてきたところですからね。

ですから、どこのレベルでオープンにするかは、色々あるわけです。何から何まで、全部見せているのは TRON だけです。OS はもうみんなに見せている。しかし、それを使ったアプリケーションの部分で重要なところは、クローズでもいいと言っている。それが TRON プロジェクトの良いところで、全世界で評価されている理由です。メーカーからみれば、信頼性の高い TRON という OS が使えて、それでノウハウは溜まっていく。しかも、自分たちが独自に加えたノウハウは出さなくてよい。そういうすぐく「虫のいい話」になっている。私はそれを OK しているんです。そして、もしそれで利益が出たら、ドネーションしてくださいと言っているわけです（笑）。するとみなさん覚えてくれていて、少し利益が出ましたとドネーションしてくれる。それに支えられてプロジェクトを進めてきたわけです。

## 社会インフラになるデータはオープンに

**松本**：お話を伺っていると、今後 IoT が本当に根付いていくためのキーワードも、そのオープン性ということでしょうか。

**坂村**：原則はオープンなのですが、何もかもオープンにすることを求めたら、それは可哀想です。先ほども申し上げましたが、そんなことをしたら、どうやって会社経営を成り立たせていくんだ、ということになってしまいます。そうは言ってない。でも、そういう考えでやっているプロジェクトは少ないんです。

**松本**：データをオープンにすると、プライバシーの問題が出てくるかもしれないですね。

**坂村**：データそのものをオープンに、というのは、今世界の流行りですね。アメリカが言い出したのですが、その最初のステップは公共データです。公共データを完全オープンにしろ。私が提唱しているのは、社会インフラになるようなデータは、企業といえども、有料にしても構わないからオープンにしようということです。その観点から産学官共同で作った VLED（ブイレッド）というオープン&ビッグデータ活用・地方創生推進機構があり、私が理事長を務めています。

**松本**：公共データというのでしょうか？

**坂村**：まず、ガバメントのデータですね。またそれとは別に、公共交通オープンデータ協議会というのを作って、私がやはり会長をやっています。そこでは JR など首都圏乗り入れの鉄道、JAL や ANA など、交通系の会社に全部入ってもらっていて、例えば交通の時刻表のデータとか、駅のデータ、それから動的データですね。山手線が今どこを走っているのか、というようなことをただでなくても良いので、オープンにして欲しいと言っています。

**松本**：例えばどの列車がどこにいるか、という情報ですね。

**坂村**：それをリアルタイムで集めて、サーバから、無料のものもあれば有料のものもありますが、とにかく公開するわけです。非常に大きなデータです。その協議会には、東京メトロと JR、更に富士通や日本電気も理事になってもらっています。

**松本**：データをどう活用していくのですか？

**坂村**：活用策の検討のために、オープン・データ・コンテストというものを開きました。東京メトロと共同で動的データ、地下鉄の動的データをオープンにして、わずか3カ月くらいの募集期間だったのですが、実に2800件の応募があり、実際にプログラムを作って Apple Store や Android に入ったものが280件ありました。もしも企業が普通に開発を進めたら、およそ5億円はかかったのではないかとわれていますが、それを完全オープンにして、一等賞金200万円です。やったわけです。オープン化するのにももちろんお金がかかりますから、200万円だけではないのですが、5億円の仕事ができたとみんなびっくりしていました。

**松本**：200万円で済めば、本当に安いですね。

## データのオープン化で何ができるか

**坂村**：面白いアイデアがたくさんありました。動的データを出すと、どんないいことがあるかということについて、

私がよく出すたとえ話があるのですが、それを本当に作ってくれた人がいた。どういうものかという、例えば1分単位で情報を送ってくる動的データによって電車が遅延しているかどうかリアルタイムでわかりますね。そうすると、明日は出張だから7時発の新幹線に乗ることにして、そのために5時に起きれば良いと考えて目覚ましをかけたとする。ところが翌朝、何か事故があって電車が止まってしまった。既定の路線で新幹線の出発駅に行くことはできないから別の経路で行く。すると予定より1時間早く5時に家を出なければいけなくなる。そのとき、5時にセットしていた目覚ましですが、自動的に4時に変更されて叩き起こされる、というものです。つまり動的タイマーができる。電車が遅れている。復旧の見込みは立っていないという情報が、動的データが公開されていれば分単位で分かる。だから4時に起こすので、ごめんなさい。早く別の電車で行きなさいというインテリジェントな目覚まし時計ができるわけです。

今私が TRON でやっているのは、そんなことばかりです。インテリジェントビデオもそのひとつです。例えば、今日は野球の試合に出かけるから、ある TV 番組をビデオに録画しておこうとセットした。しかし、もし雨が降ったら、もう野球はできないので、うちでゴロゴロしよう決めていた。すると、雨が降った場合には、録画は自動的にキャンセルされて、その時間になるとテレビがつく。もし雨が降らないで晴れていた時には出かけているから、想定通り録画機能が働くと、そういうものです。

そのためには、データをすべてオープンにして、プログラミングが自分でできるようにすべきなんです。また、そのために小学生の時からプログラムを教えて、そういうものを自分で作って見たらどうかと言っている。

先ほどの目覚ましも録画タイマーも、初歩のプログラミング能力がありオープンデータがあったら、数十行でプログラムが書けてしまいます。プログラミングを早く必修にすべきなんです。そういうことができたなら、あらゆることで役に立つ。先ほどの例は遊びのようなものですが、例えば、スマートアグリのようなものがあります。オランダは EU に加盟した途端に、それまで農業大国だったものが、あっという間に流入したスペイン産の安い果物、野菜によって、農業が全く駄目になりました。そのとき、どういことをしたかという、農民が危機感を持ってまず農地統合をやった。それから植物生産の工場化をやらないとだめだということで、コンピュータの導入をどんどん進めた。

なぜこれがうまくいったかという、農民自身、コンピュータのプログラムが書けたからです。どこか外部の専門業者に頼むという考え方は古い。自分でプログラムを書けなければ迅速で有効な対策はとれない。日本も新しい時代を作るためには、そういう新しいビジョンと哲学をできるだけ多くの人が理解し、なぜこういうことをしなければいけないのか、ということからやらないといけなさと感じています。

## 情報教育にプログラミングを

**松本：**しかしプログラムが書けても、そういう新しいサービスを思い付くような、これとこれを組み合わせると、こういう新しいことができるかと発想できなければいけませんね。

**坂村：**そういうことをやっている人は、何かもっているんです。農業なら農業をやっている人。販売ならものを売っている人。そういう人たちは、それなりに「こういうことができるたらいいな」ということは考えています。問題意識がないのは、IT 従事者くらいではないですか（笑）。ほかの業界の人たちは、みんなそれなりに問題意識を持って、何とかしたいと考えています。

問題は教育にもありますね。情報処理の教育の仕方も間違っている。入試に出ないから勉強しない。入試の問題にも、義務教育で情報関係を必修にしているにもかかわらず、出ない。受験に出ないと、みんな勉強しない。悪循環です。そのことも言っています。受験の中に取り入れられないのはおかしいと。

**松本：**IoT をもっと広げて行くためには、教育や人材育成の話もかなり大きいだらうと思います。

**坂村：**そうですね。それと、もう一つ言われているのは、今の義務教育でやっている情報教育が間違っているということ。ワードやエクセルの使い方をいまだに教えているんです。こんなもの教える必要はないでしょう。根本的な原理を教えるべきです。アメリカは今年から小学校でプログラミングを教えることにしています。イスラエルは以前からやっていますし、イギリスも去年から始めた。日本だけ取り残されているんです。

**松本：**教える人の不足もあるでしょうね。そういうことを教えられる人間がないので、仕方がないから、ワードやエクセルの使い方ぐらいしか教えられない。そこも変えていかないといけないですね。坂村先生の後に続くような人を、どうしたら日本からどんどん輩出することができるのか、課題は大きい。今日は、大変興味深いお話をありがとうございました。



# IT組織における職場リーダー像の系統的な導出

佐々木 康浩<sup>+1,+2</sup>吉川 厚<sup>+2</sup>寺野 隆雄<sup>+2</sup>

IT組織（情報システム部門など）の職場リーダーに求められる能力要件は、組織の置かれた状況やメンバーにより様々であり、一意には決められない。しかし、既存のリーダーシップ理論は、汎用的なあるべき論が示されたトップダウン的なものであり、この課題に対して妥当解を与えない。そこで、我々はコンピテンシー（行動特性）の用語を使いながら、各組織で合意形成できる職場リーダー像を描く方法論「階層的抽出法」を開発した。

## A Systematic Approach to Identify Ideal IT Leader Images

Yasuhiro Sasaki<sup>+1,+2</sup>, Atsushi Yoshikawa<sup>+2</sup>, Takao Terano<sup>+2</sup>

We propose a novel method to identify and evaluate the characteristics of IT leaders of a company. In many companies, despite many discussions of leader images, it is difficult to find differences between the images of each individual. So far, the literature on leaders has not answered these questions. Not only skills but competencies, or behavioral traits, are important for leaders in order to improve their performance and business results. We describe the procedures to draw the ideal IT leader image from behavioral traits.

### 1. はじめに

IT組織（情報システム部門など）においては、全体を統括するリーダー（CIO）も必要ではあるが、担当分野ごとのリーダーや技術的な側面のリーダーなども必要であるなど、サブリーダー的な存在（以下、職場リーダーと呼ぶ）が欠かせない。ところが、既存のリーダーシップに関する研究を、職場リーダーに求められる能力要件に適用することは困難である。

#### 1.1. 職場リーダーの位置づけ

IT組織の職場リーダーに求められるものは、必ずしもITスキルやITに関する知識・経験に依存するものではない。ロバート・カッツの能力と職位のモデルに照らし合わせれば、ITスキルの延長線上では、職場リーダーの人

材像を充分には捉えることができないことが確認される [Katz1974]。

#### 1.2. リーダーシップについての先行研究

過去から現在に至るまで、多様なリーダーシップ理論が存在する。しかしながら、それを職場リーダーに適用しようとすると困難点がある。

##### 1.2.1. 多様なリーダーシップ理論

先行研究として主要なリーダーシップ理論を探索したところ、主なものだけでも以下のとおり多様なものとなって

#### 【脚注】

+1 株式会社三菱総合研究所

+2 東京工業大学大学院 総合理工学研究科

いる(表1) [Kanai2005].

### 1.2.2. 状況アプローチの理論

F・E・フィードラーは、リーダーシップ理論の領域でコンティンジェンシー理論(条件適合理論)を打ち出した。そこで中心になるのはLPC(Least Preferred Co-worker)尺度である。LPC尺度は対になっている形容詞18項目で構成されている。回答者は、具体的人物を思い浮かべながら1から8の物差しで回答する。

フィードラーは、本人評定と部下評定のギャップを見ることで、リーダー・マッチ理論を提案するなどしている。フィードラーの研究に対しては、いくつかの批判がある。たとえば、LPC尺度がなにを測っているのかの理解が困難で、妥当性の低い尺度であることなどである。

一方、P・ハーシーとK・H・ブランチャードは、部下の成熟性に注目した。成熟性は、能力と意欲で記述される。リーダーは課題への関心と関係への関心との2軸で分類され、4つのタイプで記述される。リーダーの型と部下の成熟性の間に、関係性を見だし、状況的リーダーシップ理論として提案している。

これらは、状況によって有効なリーダー行動が異なるという点を認識させた研究であった [Kanai2005].

### 1.2.3. 先行研究を受けた我々の見解

リーダーシップは状況によって変化すると言いながら、その違いを測定するために、固定的な物差しを使っている研究がほとんどである。

しかし、職場リーダーに求められる要件を過不足なく洗い出すことこそが重要なのであって、要件が定まれば尺度の測定は大きな問題ではない。そこで我々は、職場リーダーに求められる能力項目の適切な抽出という点を解くべき問題と設定した。

## 2. IT組織においてリーダー像を描くことの難しさ

我々が次に取り上げるコンピテンシーは、IPAから提案されている「iコンピテンシー・ディクショナリ」とは、言葉は似ているものの範囲が異なる [IPA2015].

### 2.1. コンピテンシー研究の概要

職場リーダーに求める素養は各者各様であり、しかもそれぞれの思い入れは相当強い。これは、それぞれが辿ってきた成長過程の多様性を表しているともいえ、汎用的なリーダー像を定義することの難しさを物語っている。

そこで我々は、これら内容や粒度の異なるすべての素養に対して答を用意する、あるいは万能な汎用解を用意するといった検討をするのではなく、これらの要素を“仕事の成果に結びつく”という観点で「コンピテンシー」として

表1 多様なリーダーシップ理論

理論	提唱者(発表年)	概要
ミッション型リーダーシップ	P.セルズニック(1957)	組織が単に効率よく事をなしていくだけの状態から、使命(ミッション)を帯びて、その組織ならではの個性(キャラクター)が生まれ、人々がそれにコミットするようになると、組織は制度に変わる。それを実現するのが制度的リーダー。
サーバント・リーダーシップ	R.グリーンリーフ(1964,1977)	リーダーが自分たちに奉仕する、尽くしてくれると思うときに、フォロワーは自然についてきてくれる。
変革型リーダーシップ	J.M.バーンズ(1978)	リーダーがフォロワーになにかを提供し、それを見返りにフォロワーに貢献してもらうという交換型リーダーと対比され、歴史・時代の節目で大転換を起こすのが変革型リーダー。
ビジョナリー・リーダーシップ	B.ナナス(1992)	ビジョンの創造と実現が、リーダーシップ発揮の肝心なプロセスである。
フォロワーシップ理論	R.ケリー(1992)	リーダーの立場にある人は、自分の姿を鏡の向こう側(つまり、フォロワー)の立場から見るのがいい。フォロワーのほうがリードしているという気になってもらうこと、さらに、フォロワーシップがうまくできることがリーダーシップにつながる。

[Kanai2005] から抜粋

表2 状況アプローチの理論

理論	提唱者(発表年)	概要
条件適合(コンティンジェンシー)理論	F.E.フィードラー(1967)	LPC(Least Preferred Co-worker)という尺度で測定されたリーダーシップが、業績に与える効果が、リーダーが直面する状況が好意的かどうか(課題が明確で、人間関係がよく、職位のパワーが十分にあれば最も好意的)によって異なっていくことを発見した。
状況リーダーシップ	P.ハーシー, K.H.ブランチャード(1977)	有効なリーダーシップ・スタイルは、部下の成熟度の変化に依存して推移していく。
リーダーシップの代替物アプローチ	S.カー(1977)	リーダーシップを左右するのは、リーダーの行動の代わりになるものとそれを阻害するものが、組織の内外に多数存在する。

[Kanai2005] から抜粋

集約し、その「コンピテンシー」を定義することでIT組織の職場リーダー像を明確化させるという考えをとることとした。

IPAから、2014年7月に「iコンピテンシー・ディクショナリ」が公開されている。これはITスキル標準やUISS、組込みスキル標準(ETSS)を統合した「共通キャリア・スキルフレームワーク(CCSF)」が元となっている。「iコンピテンシー・ディクショナリ」は統合されたとは言え、また一部に“ヒューマンスキル”が含まれているものの、対象はITスキルが中心である。我々が取り上げるコンピテンシーとは、コミュニケーションやネゴシエーションなど、いわばヒューマンスキルに該当する部分に焦点を当てており、「高い業績を上げるための行動特性」などと訳されるものである。

そもそもは、ハーバード大学の心理学者であるデイビッド・C・マクレランド(David C.McClelland)を中心にしたグループが、1970年代初めに米務省より、「外交官(外務情報職員)において、採用時のテスト成績や学歴・知能レベルが同等にも関わらず、配属後の実業績に差が生じてしまうのは何故か」について調査依頼を受け、行った研究が起源と言われている。

いくら高度な知識や技能を保有していても、それだけでは成果に直結せず、高い業績を上げるために必要な要因が存在する、という考えに基づいている。

## 2.2. コンピテンシーに関する先行研究

コンピテンシーに関する研究として、「GLC 調査研究会」の産学共同研究がある [Nagai2005]。

この研究の目的は、特定の企業や地域を対象とした有効なコンピテンシー尺度の開発ではなく、より汎用性と柔軟性の高いモデルを構築することである。このため、複数の先行研究を分析する手法を用いて、共通する項目をリストアップするとともに、その後のインタビュー調査と質問票

表3 グローバルマネジャーのコンピテンシー尺度

要素	No.	コンピテンシー	行動例
課題設定	1	多様性の受容	現地パートナーの状況に配慮する メンバーのアイデアや提案を傾聴する
	2	学習力	自分の業務に必要な知識・スキルを自発的に習得する 業務上の経験を通じて、自分の知識を継続的に高める
	3	達成志向	困難にあっても後ずさりしない 時間的な制約条件に注意を払う
	4	不確実性への耐性	不確実な状況や、とても複雑な状況においても臆しない あいまいな出来事や情報を整理し、新たなコンセプトを提案する
	5	先見性	政治的、社会的、文化的な面が事業に与える影響について考える 技術革新の進展を予測し、自分の役割に対する影響を考える
意思決定	6	情報収集	ビジネスに影響を与える社会経済や政治、環境問題など様々な情報を収集する 最新の情報を重要視し、直接情報がとれるような情報源を確保している
	7	創造志向	既存の考えにとらわれずに、自由な発想をする さまざまな考えや情報を統合して新しいアイデアを生み出す
	8	分析的思考	多様な情報を論理的に系統づけて組み立てる 多くの情報を体系づけることによって、顕在化していない問題を発見する。
	9	戦略立案	目標を達成する上で効果的であり、コスト効率のよい計画を立てる 自社の強みを最大限に活用するビジネスチャンスを探す
	10	危機管理	想定されるリスクを洗い出し、影響度を把握する 危機が起こった際の、対応策と役割分担を明確にしている
実行	11	組織マネジメント	メンバーが成長意欲を持てるように仕事の役割を与えている メンバーを公平に扱う
	12	成果マネジメント	組織内での業務プロセスを構築し、確実に業務を達成する 目標が未達成の場合は要因分析を行い、次の活動に反映させる
	13	変革推進	人を引き付けるアイデアで、理論を実践に変える 部門や機能の変革課題を形成し、あるべき姿に向けた施策を打ち出す
	14	コミュニケーション	あいまいな状況や誤解を解消するようにする 仕事を進める上で、関連部署の支援・理解を得ている
	15	葛藤処理	もめごとの当事者に対して両者の言い分を聞く 取引先の利害が対立したときにも、合意形成を図る
コア	16	倫理性	ミスをしたときは、素直に認める 守れないかもしれない約束はしない
	17	ネットワーキング	社外のリソースをいつでも利用できるように、幅広いネットワークを築く 他者から協力を求められたときは、できるだけ支援する
	18	説明力	いかなる聞き手にも説得力ある説明をする 相手によって適切に語句や表現を選び意思疎通する

[Nagai2005]

調査を用いて、コンピテンシーの妥当性を検証するアプローチが用いられている。

### (1) コンピテンシーモデルの構築

先行研究から得られたグローバルマネジャーの行動を研究グループ内で議論し、分類基準に適合するものを採択し、重複するものを統合するなどの手続きをとった。以上のような作業を通して、初期の仮説的なコンピテンシーとして、3次元15要素75項目のコンピテンシー尺度を構築した。

### (2) グローバルマネジャーへのインタビュー調査

30名のグローバルマネジャーに各60分程度のインタビューを行い、グローバルマネジャーの詳細な行動例を抽出し、仮説のコンピテンシー項目と照合し、仮説の妥当性を検証した。さらに、当初仮説に含まれていなかった項目を追加、修正を行った。

### (3) 人事責任者への質問票調査

グローバルマネジャーのコンピテンシー仮説の信頼性を検証するため、人事責任者に対する短答式の質問票調査を実施した。日・アジア・米・欧地域の多国籍の人事責任者138名から回答を得た。確認的因子分析および信頼性係数の解析を行い、最終的に18のコンピテンシーと57項目からなるコンピテンシー尺度を開発した(表3)。

本研究は、独自のコンピテンシー尺度を開発し5地域間で国際比較を行ったという特徴を持つが、専門研究者でないコンピテンシー項目の系統的な抽出が困難であるという点に課題が残る。

## 2.3. 認知に関する先行研究

今回提案する「階層的抽出法」の先行研究を調査したところ、我々の研究に近接する研究分野として、認知言語学という領域があることがわかった [Lee2010]。

理論的観点からの言語研究では、長年の伝統として「言語の構造的側面」と「実際の使用の問題」は分離して記述すべきだと考えられてきた。

一方、言語学の中で我々の研究に類似する用法基盤モデル(Usage-Based Model)は、伝統的見方に対して対立する考え方を示している。言語の構造的特徴は使用(usage)に応じて動的に変化するという認識のもとで提案されたボトムアップ的言語モデルである。

用法基盤モデルの発案者であるランカカー(Langacker)は「抽象度の低いスキーマ」が重要であると述べている。

図1では、「太郎が学校に行く」という実例に対して、具体性の異なるスキーマ(抽象化によって細部が欠けたパターン)が存在することを表している。それによると、「学校」や「太郎」といった名詞の部分のみが人や場所などの上位概念に抽象化された(c)のスキーマから、動詞まで上位概念に抽象化された(b)のスキーマ、さらに語彙要素がすべて抽象

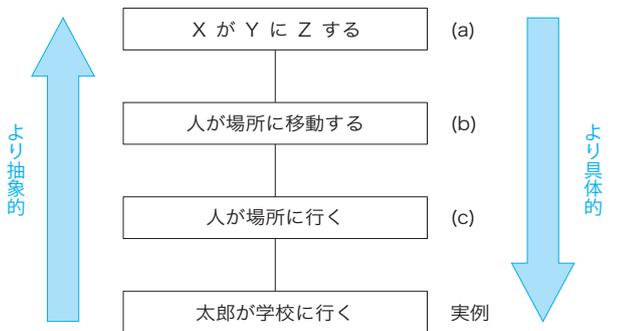


図1 スキーマの階層

[Lee2010]

化した(a)のスキーマまで、複数のスキーマが存在することを表している。この図のポイントは、実例から上に進めば進むほど、抽象的なスキーマという解釈になる。反対に下へ下へ行けば下りるほど具体的なスキーマということになる。

抽象度が高くなればなるほど、その実在可能性が薄いと考えられるため、抽象度の低いスキーマを重視するのである。

## 2.4. 先行研究についてのまとめ

自分たちの職場リーダー像を描きたいと希望したとしても、このようにリーダーシップ理論は百花繚乱であり、どの考え方に従うべきか指針はない。

コンピテンシー(行動特性)に絞って定義しようとしても、国によってあるいは企業によって、その捉え方は千差万別である。

一方、最近の認知言語学での研究によれば、言葉としての辞書的な定義だけでは人によって認識の幅が大きくなるため、抽象度の低い(具体的な)表現によって説明する必要のあることが示唆された。

すなわち、IT組織の職場リーダーに求められる能力要件の定義にあたっては、リーダーの行動レベルも含めて定義する必要がある。

## 3. 階層的抽出法の提案

IT組織の職場リーダーに求められる能力要件は、組織の置かれた状況やメンバーにより様々であり、一意には決められない。しかし、既存のリーダーシップ理論は、汎用的なあるべき論が示されたトップダウン的なものであり、この課題に対して妥当解を与えない。そこで、我々はコンピテンシー(行動特性)を使いながら、職場リーダー像を描く方法論を提案する。

### 3.1. 階層的抽出法の概要

経営環境やメンバー構成など組織の置かれている状況などにより求められる職場リーダーの能力要件が異なるという課題に対して、固定的なコンピテンシー項目を提示するのではなく、組織員にコンピテンシー項目の選択から始まり、その内容を検討させるという方法をとることで、柔軟性を持たせた。

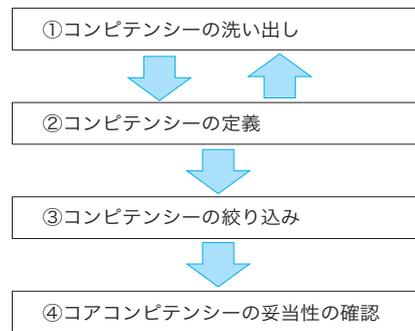


図2 職場リーダー像作成手順

コンピテンシー項目を選択しモデルを定義する際に組織全員が同じものを想起するとは限らないという認識に相違のある問題がある。そこで、認知言語学での研究からスキーマ概念を参考とし、言葉の定義→要求事項の確認→具体的な行動パターンの検討というように階層的にコンピテンシー項目の整理を行うことで認識の相違の最小化を図る。

抽出され定義されたコンピテンシー項目ごとの重要度がわからないという課題に対しては、コンピテンシー項目を定義した組織員に対して別の切り口からアンケートを行うことで、コンピテンシー項目の重要度の設定が可能であり、これをコアコンピテンシーと呼ぶこととする。

階層的抽出法の詳細は以降で述べるが、この作業プロセスをなぞれば、各IT組織で独自の職場リーダー像を行動レベルで記述することが可能となる。その手順は図2のとおりである。

## 3.2. 階層的抽出法の詳細

### 3.2.1. コンピテンシー項目の洗い出し

一言でコンピテンシーといっても、その中には多くの要素が存在する。IT組織の職場リーダーに求めるコンピテンシーを洗い出す。

**【方法】**「各IT組織で職場リーダーに求める行動」や「参加者が過去の経験などから職場リーダーに必要なと思われる行動」と、それに対応するコンピテンシーをコンピテンシー検討参加者が各々書き出す。それを集計し、類似の表現や重複していると思われる行動を参加者全員で議論を通して整理する。参考情報を与えず参加者が自由に書き出す場合もあるが、表4のように、定評のある先行研究などのコンピテンシー一覧を議論の出発点として参考にすることもある。

### 3.2.2. コンピテンシーの定義

階層的抽出法ではコンピテンシーについて、階層的に具体化することによって、用語の齟齬がないように整理できる点に特徴がある。1層目では、そのコンピテンシーを表す言葉の辞書的な意味を記述し、2層目では、そのコンピテンシーを実現するために必要な項目(要求事項)を記述する。3層目では、2層目で示された項目と認定される具体

表 4 定評のあるコンピテンシーの例

分類	項目	説明文
1	達成重視	すぐれた仕事を達成し、あるいは卓越した基準に挑む姿勢を指す。
2	達成とアクション	秩序、クオリティー、正確性への関心
3		イニシアティブ
4	情報探索	状況を、「額面どおり」に受け取らずに、さらに多くの情報を得ようとする意欲を指す。
5	支援と人的サービス	対人関係理解
6	顧客サービス重視	ほかの人たちのニーズに応え、支援し、サービスを提供したいと願う願望を指す。
7	インパクトと影響力	インパクトと影響力
8		組織の理解
9		関係の構築
10	マネジメント・コンピテンシー	ほかの人たちの開発
11		指揮命令-自己表現力と地域に伴うパワーの活用
12		チームワークと協調
13		チーム・リーダーシップ
14	認知コンピテンシー	分析的思考
15		概念化思考
16		技術的/専門的/マネジメント専門能力
17		セルフ・コントロール
18	個人の効果性	自己確信
19		柔軟性
20		組織へのコミットメント

[Spencer2011]

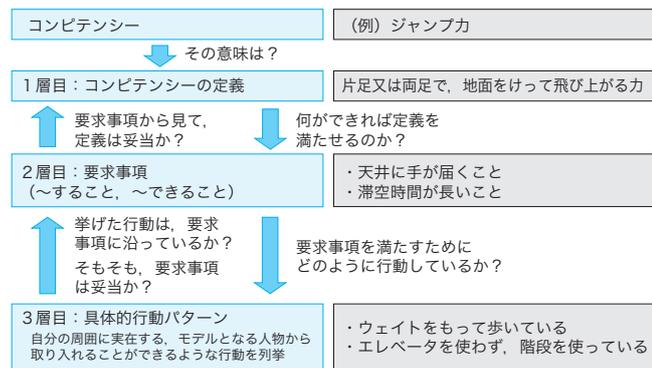


図 3 コンピテンシーの検討モデル

的行動パターンを示す。このことにより、コンピテンシーを持っているかどうかの判断は行動と紐づけられ、人によってぶれることがなくなる。2層目は3層目の行動パターンが妥当なものなのかを判断したり、行動に抜け漏れがないかを判断するのに役立つ。留意点は、この2層目、3層目を記述する際に、抽象化（一般化）せずに、具体的な人物イメージが浮かぶ形で記述するよう心掛けることである。

組織によって、4層、5層になる場合もある（図3）。

### 3.2.3. コンピテンシーの絞り込み

次に、「IT組織の職場リーダーに求められるコンピテンシー」項目の中から、特に重要と思われるコンピテンシーに絞る。これをコアコンピテンシーと呼ぶ。

【方法】参加者それぞれが100点を持ち点として、重要だと思えるコンピテンシーに各自で点数を割り振る。その後、各コンピテンシー単位で点数と点数を入れた人数を集計する。この結果を元に、回答人数の多い（参加者の3分の1以上が選ぶ）項目に絞り、次に点数の上位の項目（項目数の3分の1程度）を選ぶ。その後参加者に確認をとり、合意の得られるコンピテンシーをコアコンピテンシーとして設定する。

表 5 コンピテンシーの絞り込み (例)

コンピテンシー	管理職		社員			点数	人数
	A	B	C	D	E		
1 達成重視	30	10	10	5		110	8
2 クオリティーへの関心	10	10		5	50	150	8
3 イニシアティブ			30	5		70	4
4 情報探索				5		10	2
5 対人関係理解				5		10	2
6 顧客サービス重視	10	10		5		50	6
7 インパクトと影響力				5		10	2
8 組織の理解	10	10		5		50	6
9 関係の構築				5		10	2
10 人材育成	10	10		5		60	6
11 指揮命令				5		10	2
12 チームワークと協調		10		5		30	4
13 チーム・リーダーシップ		10		5		30	4
14 分析的思考				5		10	2
15 概念化思考			30	5		70	4
16 実務能力	10	30	5	50		190	8
17 セルフ・コントロール				5		10	2
18 自己確信	10	10		5		50	6
19 柔軟性				5		10	2
20 組織へのコミットメント	20	10		5		70	6
合計点	100	100	100	100	100		

### 3.2.4. コアコンピテンシーの妥当性の確認

最後に、もう一度「IT組織の職場リーダーに求められるコンピテンシー」項目を使い、絞り込んだコアコンピテンシーが妥当なものかどうかの確認を行う。

【方法】参加者それぞれが身近にいる（あるいは過去にいた）優秀な職場リーダーを想定する。その特定の一人を思い

浮かべながら、その職場リーダーが保有している（いた）と思われるコンピテンシーに○印をつけていく。その後、各コンピテンシー単位に○の数を集計し、コアコンピテンシーとして先に選んだ項目が妥当であったかどうかを確認する。

表 6 コアコンピテンシーの妥当性の確認（例）

コンピテンシー	管理職		社員			○の数
	A	B	C	D	E	
特定のリーダーを想定→	○さん	△さん	□さん	◇さん	▽さん	
1 達成重視	○	○	○	○		8
2 クオリティーへの関心	○	○		○	○	7
3 イニシアティブ			○	○		4
4 情報探索				○		2
5 対人関係理解				○		2
6 顧客サービス重視	○	○				4
7 インパクトと影響力				○		2
8 組織の理解	○			○		4
9 関係の構築				○		2
10 人材育成	○	○		○		7
11 指揮命令				○		2
12 チームワークと協調		○		○		4
13 チーム・リーダーシップ		○		○		4
14 分析的思考				○		2
15 概念化思考			○	○		4
16 実務能力		○	○	○	○	9
17 セルフ・コントロール				○		2
18 自己確信	○	○				4
19 柔軟性				○		2
20 組織へのコミットメント	○	○		○		6
○の数	7	9	4	18	2	

#### 4. 階層的抽出法の適用実験

三菱 CC 研究会（以下、CC 研）を実験対象として、コンピテンシー整理に関する、階層的抽出法の適用実験を行った（2008年9月実施）[Mitsubishi2009]。

【対象者】三菱グループの情報システム部門やシステム子会社に所属する 30～40 代の中堅社員 19 名。これらの社員は会社横断的に現業の IT 支援という共通部分があり、その組織内という仮想組織を想定できる。

##### 4.1. コンピテンシーの洗い出し

CC 研メンバーそれぞれが自組織の職場リーダーに求めるコンピテンシーを洗い出した。

【方法】「各社で職場リーダーに求める行動」と、それに対応するコンピテンシーを対象者が各々書き出した。それを集計し、類似の表現や重複していると思われる行動を対象者全員の議論を通して整理した。

【結果】様々な行動例とコンピテンシーが洗い出されたが、表 7 に示す 25 項目に集約された。

##### 4.2. コンピテンシー定義

CC 研メンバーは、コンピテンシーの洗い出しと平行して、

表 7 コンピテンシーの洗い出し結果

ID	コンピテンシー	行動例
1	顧客志向	お客様の視点に立った、企画・立案ができること
2	実務能力（IT スキル・業務スキル）	気付きや論理構成に必要な幅広く、適度に深い知識を有すること
3	仲間意識	プロジェクトメンバーが前向きに仕事に取り組みめるよう働きかけること
4	創造性	組織の固定概念に捕らわれない大胆な発想で企画立案できる
5	公正・誠実	公平な査定、信賞必罰を目指す
6	持論・ポリシー	自らの考えや方針を明確に打ち出し、メンバーを前向きな方向へ導いている
7	達成意欲・積極性	主体的に高いレベルの課題を設定し、達成に向けて常に全力を傾注している
8	判断力	不確実な要素がある中で、プロジェクトを推進し成功に導くことができるかどうか自分自身で判断できること
9	人材育成	職場や、プロジェクトのメンバーの特徴や持ち味に応じたアドバイスを行うこと
10	分析力・論理性	多面的な分析に基づく、実現可能性の高いプランを企画立案できる
11	構想力・ビジョン	複数案件を効率的に推進できるような開発計画等の業務計画を立案すること
12	計画的な遂行	期限をデッドエンドより少し前に設定し、逆算して進める癖を付ける
13	リスクマネジメント	少しでも問題が起きれば蓋をせず、問題が大きくなったら上にすぐに相談する
14	コミュニケーション	「聴くこと」、「話すこと」を通じて、相手との信頼関係を構築できる
15	折衝・調整力・ネゴシエーション	システムを開発することによる会社へのメリットを認識し、社内調整できること
16	チーム運営	長い目で見て、最終目的に到達するよう、うまくコントロールすることを考える
17	管理統制力・イニシアティブ	自分の定めたるルールや出した指示を守らせている
18	変革志向	日々の改善を大事にしつつ、改善の手を休めず、問題点を探し出して、次々と"少し"良くしていく
19	ストレス耐性	苦しい時こそ笑顔でいられること
20	国際性	日本語以外の言語でコミュニケーションがとれる
21	気づき・洞察力	得た情報の特異点、真因、影響度、他情報との依存関係が感じられること
22	情報活用力	内外の情報ネットワークの構築に努めている
23	目的志向	常にゴールを意識して行動できること
24	情報管理	情報管理の徹底及びコンプライアンスの遵守を行うこと
25	観察力	決まった観察点がある

表 8 CC 研における第 1 層：言葉の定義（一部）

コンピテンシー	第 1 層：言葉の定義
持論・ポリシー	日々行動していく際のぶれない判断基準と重視する価値観 IT リーダーが信頼を得る正しい考え方
達成意欲・積極性	目標を達成・成し遂げる情熱 どのような状況でも、他人や環境のせいにならず、自らが最大限に成せることを考え取り組む姿勢。他の行動特性を発揮するための動力源・エネルギー源
構想力・ビジョン	事業戦略を本質的に理解し、IT を通じて、期待を越えた未来像を設計する力
管理統制力・イニシアティブ	自らが先頭に立ち、チームやプロジェクトを一つにまとめて推進すること
折衝・調整力・ネゴシエーション	プロジェクト遂行やチーム運営にあたって発生する課題について、交渉・調整の結果、合意を導き出すこと
コミュニケーション	風通しの良い環境を作り、熱意を持って意思疎通を図ること
人材育成	【タイプ I】適切な目標を設定の上、本人に理解させ、達成に向けてフォローする 【タイプ II】仕事に誇りを持っており、仕事振りを見せ、人の目標になることで周囲の成長を促す
判断力	リーダーとして判断が必要な時に、判断材料と自らの判断基準を元に、タイムリーに（求められたタイミングで）自分の考えを定めることができる力
実務能力	IT スキル・業務スキルに長けており、使いこなすことができる能力

各コンピテンシー項目の階層的な定義を行った。

#### 4.2.1. 第1層：言葉の定義

1層目では「ユーザー企業のIT組織」の職場リーダーに求められる各コンピテンシーの辞書的な言葉の定義を定めた(表8参照)。

表8は、25項目からの抜粋である。人材育成というコンピテンシー項目については、一意に決められず2つのタイプが存在するという結論になった。

#### 4.2.2. 第2層：要求事項

次に、2層目では、「ユーザー企業のIT組織」の職場リーダーとして、コンピテンシーが高いということは何かできれば良いのか要求事項を整理した(表9)。

表9 CC研における第2層：要求事項(一部)

コンピテンシー	第2層：要求事項
判断力	①「材料」を収集する努力をしていること
	②収集した「材料」を整理・咀嚼し、自らの確固たる「基準」を作り上げていること
	③要求事項の本質を理解し、自らの判断基準と照らし合わせ判断できること
	④「タイミング」を外さず、的確な時期に判断を行う事が出来ること
達成意欲・積極性	①仕事を成し遂げたり、目標を達成しようとするやる気、やり抜く気力があること
	②遂行・達成のために周囲を巻き込み、関係者の賛同や協力を取り付ける事が出来ること
	③目的・狙いの達成に向けて、部下や関係者にやる気を起こさせるために与えられた権限を駆使して影響を及ぼすことが出来ること
持論・ポリシー	①行動や発言の目的や理由をメンバーに明示し理解させることができる ・会社の方針や、組織の方針に沿った行動をしている ・自分の判断に責任を持っている
	②上位方針や自身の役割の変更に応じ、適時見直しを図り、発展させることができる。(発展・成長)

(以下省略)

#### 4.2.3. 第3層：行動例

また、3層目では、「コンピテンシーの高いIT組織の職場リーダーが日常業務でどのような行動を取っているのか」を具体的に表し、112項目を洗い出した。ただし、メンバーが所属する各企業により、体制や仕事のやり方、人材育成方針が異なるため、職場リーダーとして定めたコンピテンシーも、そのコンピテンシーが高い職場リーダーが取る行動も様々である。今回、3層目の具体的な行動例は、教科書的な通り一遍の表現とすることを避け、実験対象であるCC研参加メンバーやメンバーが所属する組織に実在する職場リーダーが、実際に取っている行動を挙げた。

### 4.3. コアコンピテンシーの抽出

前項で洗い出した「ユーザー企業のIT組織」の職場リーダーに求められるコンピテンシー25項目の中から、CC研19名が各社共通と思われるコアコンピテンシーについて以下の手法で絞り込みを行った。

**【方法】**参加者それぞれが100点を持ち点として、重要だと思えるコンピテンシーに各自で点数を割り振った。その

後、各コンピテンシー単位で点数と点数を入れた人数を集計した。この結果を元に、回答人数の多い(6人以上)の項目に絞り、次に点数の上位10項目以内を選んだ。その後参加者全員で議論し、合意の得られた9コンピテンシーをコアコンピテンシーとして設定した。

**【結果】**各コンピテンシーの集計結果は表10のとおりである。濃い網掛けが5人以下のため最初に候補から除外され、薄い網掛けの部分が点数と議論で除外されたコンピテンシーを示している。

表10 コアコンピテンシー抽出結果

No.	ID	コンピテンシー	点数	人数
1	2	実務能力(ITスキル・業務スキル)	314	14
2	11	構想力・ビジョン	154	11
3	14	コミュニケーション	162	10
4	15	折衝・調整力・ネゴシエーション	87	8
5	7	達成意欲・積極性	87	7
6	8	判断力	71	7
7	9	人材育成	65	7
8	6	持論・ポリシー	120	6
9	17	管理統制力・イニシアティブ	100	6
10	3	仲間意識	65	6
11	19	ストレス耐性	53	6
12	18	変革志向	80	5
13	10	分析力・論理性	40	5
14	13	リスクマネジメント	55	4
15	4	創造性	52	4
16	16	チーム運営	60	3
17	5	公正・誠実	40	3
18	1	顧客志向	20	3
19	12	計画的な遂行	18	3
20	21	気づき・洞察力	16	3
21	20	国際性	5	2
22	23	目的志向	16	1
23	24	情報管理	10	1
24	22	情報活用力	9	1
25	25	観察力	1	1

## 5. 階層的抽出法の結果の妥当性の確認

### 5.1. 確認方法の概要

階層的抽出法により作成したコアコンピテンシーがあるべき職場リーダーを形成するコアコンピテンシーとして妥当と言えるのかどうかを確かめるため、CC研メンバー19名に対して表11のようなアンケートを実施した。

表11 身近なリーダーに関する個人アンケート

- アンケートの単位：個人  
身近な人で、「あのこそITリーダーだと思う」人とか「ずっと模範としてきた」人を思い浮かべてください。
- アンケート入力方法：
  - ①コンピテンシー一覧112項目をご覧頂き、アンケート回答列の一番上のセルには、思い浮かべたリーダーのタイプとして何か一言で言い表せる言葉があれば入れてください。(必須ではありません)
  - ②アンケート回答列に、その人の行動を特によく表していると思われる項目20個程度を目安に○をつけてください。

判断力

- ・時間があれば現場に足を運び、色々聞いて現状把握に努めている。
- ・ユーザーから要望が来た際には、目的・効果・採算性は必ず確認している。
- ・レビューや相談したタイミング等判断を頼まれたその場で判断し回答している（判断を先延ばししたり、曖昧にしない）。

管理統制力・イニシアティブ

- ・上位承認ドキュメントを、必ずオープンにし、メンバーに説明している。
- ・自分の口で、情熱を込めて目標を語っている。
- ・プロジェクトを実施するときは、必ず体制図（役割）、目的・目標、納期を明確にドキュメントで表している。

達成意欲・積極性

- ・つらい局面でも愚痴・弱音を言わない。
- ・計画未達や方針変更が発生した場合、一蹴決めたことでも迅速に撤回し、何故そうするのか、その理由の説明・説得を怠らない。
- ・事ある毎に「これをやればこうなる、これが出来ればあんな」と目標達成や仕事が成し遂げられたときの状況を部下や関係者に語っている。

折衝・調整力・ネゴシエーション

- ・特に譲れないポイントは明確にし、場を臨んでいる。
- ・最低3つ（松竹梅）の家を持って場に臨んでいる。
- ・組織での役職や年齢の上下を持ち込んでいると取られないように、とても丁寧な言葉使いをしている。

持論・ポリシー

- ・プロジェクト方針に合わない部下を、厳しく指導する。
- ・個別課題対策を検討する際、観点が会社レベル・社会レベルと高い。
- ・成否に関わらず、何故こういう結果になったのか、道筋を立てて語る。そして、その内容に納得感がある。



コミュニケーション

- ・表情豊かに話している。表情豊かに聴いている。
- ・何か話しがあるとき、すぐに相手のところへ足を運ぶ。フットワークが軽い。
- ・気づくと、いつもホワイトボードの前に立っている。

構想力・ビジョン

- ・メンバーの報告に対し、繰り返し問いかけている。
- ・業務部門の要求に対し、システム側の視点だけでなく業務部門や経営の視点を加え、多面的に必要性を吟味している。
- ・決定権を持っている一マンを見極め、仕事以外の付き合いでも、自分の思いを伝えている。

人材育成

- ・経験談などを織り交ぜて分かりやすい例を使い、論理的に説明している。
- ・何かが増しように質問に答えている。
- ・なぜか具体的な成果物が大小無くでてくる。またそれを惜しめない。

図4 行動パターンで記述された職場リーダー像

5.2. 確認結果

表11のアンケートを集計し、○印の多かった（重要な）行動パターンについて、コンピテンシーごとに整理し、CC研としての「職場リーダー像」を描いた（図4）。

得られた「職場リーダー像」について、CC研メンバーに意見を求めたところ、19名中12名から肯定的な意見があり否定的な意見はなかった。得られたリーダー像に対する意見を表12に、提案手法に関する意見を表13に、掲載する。表12をみると、No.2では「新鮮に感じる」という記述、No.3では「こんなに多様であることに驚きを感じ」とあり、本手法によって本人にとって目新しいものが見つかったことを示している。表13のNo.6では「自社を対象としたリサーチに活用できる」として、手法の有用性を認めた意見が出ている。

6. 結論と課題

6.1. 今回得られた知見

既存のリーダーシップ理論では適用できる解のなかった、IT組織の職場リーダーに求められる能力要件の整理という問題に対して、我々はコンピテンシー（行動特性）を使いながら、各組織で合意形成できる職場リーダー像を描く方法論「階層的抽出法」を開発した。しかも、本手法は、行動レベルにまで結びついていることから、適用者ごとの解釈が異なることがないことはもちろん、リーダーの育成や評価にも使える枠組みとなっている。本方法を適用した結果、身の回りにないリーダー像の発見も報告されており、また、平易に使えることから適用したい旨の報告もあった。

今回は、ユーザー企業のIT組織を実験対象としたが、この方法論は専門性の高い仕事の要求される職場において広く適用可能性があると考えている。

6.2. 今後の課題

今後の課題として、コンピテンシー項目の設定について、わかりやすく実用性の高い方法を考案していきたい。

表12 得られたリーダー像に対する意見

No.	CC研メンバーの意見（一部）
1	自らの行動パターンを分析することで、よく出来ている「強み」や苦手な「弱み」が、明確になり、リーダーとしての自己強化ポイントを分析するのに役立つと感じました。
2	整理した行動パターンには、身近に感じるものと新鮮に感じるものがありましたが、それだけ目標となるリーダーには色々なパターンがあるものだと感じました。自分なりのリーダー像を持ち、それに必要な行動パターンを選択して切磋琢磨する、そのように心がけて行動する参考書の一つになると思いました。
3	ユーザー企業のITリーダー像がこんなに多様であることに驚きを感じています。これは企業ごとに環境・文化が違い、それに伴い使用される言葉、言い回し等が違いため、さまざまなリーダー像が語られているのではないかと推察します。そういう意味では画一的なリーダー像よりそれぞれの企業ごとに具体的な行動パターンを考えることに大きな意義があると思いました。
:	:

表13 提案手法に関する意見

No.	CC研メンバーの意見（一部）
4	自分の周りには先輩・上司だけだと数人の行動しか出てきませんが、メンバー19人から出た112個の行動パターンは、周りではないタイプの行動もあり勉強になりました。自分の尊敬する上司をイメージして○付けを行いました。全てのコンピテンシーを網羅する行動をバランスよく取っているのだと発見できました。
5	自分のチームのリーダー候補生に、明日から始められる育成方法として活用したいと思います。そのためには、教えた側から大事だと思うことを、この手法を使って示すとともに、教えられる側が、このアンケート表を眺めて、自分で目標を設定する二つのアプローチが有益だと思います。
6	今回、リーダーの行動特性アンケートでは、回答者の所属する組織風土の傾向が反映されそうなのが、大変興味深いです。これは、「自分の組織にとって、どういう人材がリーダーで、どんなリーダーシップの発揮が求められるのか」というような、自社を対象にしたリサーチに活用できると思いました。
:	:

表14 階層的抽出法の特徴

	課題	方策
柔軟性	組織の置かれている状況（経営環境やメンバー構成など）により、求められる職場リーダーの能力要件が異なる	固定的なコンピテンシー項目を提示するのではなく、組織員にコンピテンシー項目の選択から始まり、その内容を検討させる
認識の一致	コンピテンシー項目を選択しモデルを定義する際に組織全員が同じものを想起するとは限らない	階層的にコンピテンシー項目の整理を行う（言葉の定義→要求事項の確認→具体的行動パターンの検討）
コアコンピテンシー	抽出され定義されたコンピテンシー項目ごとの重要度がわからない	コンピテンシー項目を定義した組織員に対して別の切り口からアンケートを行うことで、コンピテンシー項目の重要度を設定する

得られた結果を、親しみやすく表現するため、マーケティングで適用されているペルソナ手法の適用も検討していきたい。

【参考文献】

[Katz1974] Robert Katz : Skills of an effective administrator, Harvard Business Review, September-October, 1974  
 [Kanai2005] 金井壽宏：リーダーシップ入門，日本経済新聞出版社，2005  
 [IPA2015] IPA：iコンピテンシー・ディクショナリ解説書，2015  
 [Nagai2005] 永井裕久：パフォーマンスを生み出すグローバルリーダーの条件，白桃書房，2005  
 [Lee2010] 李在鎬：認知言語学への誘い—意味と文法の世界，開拓社，2010  
 [Spencer2011] ライル・スペンサー，シグネ・スペンサー：コンピテンシー・マネジメントの展開（完訳版），生産性出版，2011  
 [Mitsubishi2009] 三菱CC研究会：ユーザー企業のIT人材に関する研究（次世代リーダー育成の研究），2009

# 多様化が進むソフトウェア開発スタイル ～サステイナブルなデジタルビジネスのために～

SEC システムグループリーダー  
山下 博之

SEC 調査役  
室 修治

開発スピードや品質などに関する要求の高まりに応えるために、ソフトウェア開発の現場では様々な開発スタイルが採用されている。本稿では、IPA/SEC の調査・検討の成果や内外の公開調査結果などを通して、このような状況を概観すると共に、今後ますます進展する IoT 時代に向けた課題について考える。

## 1 ICT の進展とシステム開発

コンピュータがこの世に出現して以降、当初は単なる業務支援のために使われていた情報通信技術 (ICT) は、図 1 に示すように、その技術自身の進展と時代の要請とが相まって、企業の基幹業務や社会生活のコモディティなどへと、その利用が質・量ともに拡大してきている。ICT の持つ優れた能力を認めた私たちは、ビジネス戦略の武器や社会サービス、生活必需品などへの利用と、ますます ICT への依存を高めている。このような流れの中で、ICT の中核をなすソフトウェアはその重要性が増し、ソフトウェア開発への要求事項もまた変遷してきた。業務支援システムに使われ始めた頃は、品質、とくに信頼性や操作性が厳しく評価されていた。その後、ビジネス戦略の要として利用されるようになると、競争優位を維持するために開発スピードが重視されるようになってきた [1]。そして最近では、いわゆる QCD (品質、コスト、納期) の目標達成よりも、ビジネスに価値をもたらすことがソフトウェア開発プロジェクトの成功を意味すると思われるようになってきている [2]。

ただし、ソフトウェア開発への要求は画一的なものでは

なく、ICT システムの対象やそれを使うビジネスの状況に応じ、重視する評価ポイントは異なる。例えば、システム種別の観点では、企業の基幹系システムでは品質や継承性が、Web・フロント系システムでは開発スピードや変更容易性が、それぞれ重視されるという調査結果が報告されている [3]。また、ビジネス・ステージの観点では、黎明期には短い間隔で新サービスを継続的にリリースして激しい競争を勝ち抜くための開発が、安定期には利用者の評価に対応した更新版を定期的にリリースするための開発が、それぞれ求められる [4]。更に、IPA/SEC が公開したソフトウェア開発データの分析結果によれば、金融・保険業は他業種に比べ、生産性が低く信頼性が高い傾向が見られる [5]。

## 2 ソフトウェア開発スタイル

ソフトウェア開発に対する要求の変遷に伴い、それに対応した開発スタイルの採用や各種技術の利用が行われてきている。システム全体の要求を確定した後に設計、製造、試験と順に開発を進める従来のウォーターフォール型開発に加えて、少しずつ作って確かめながら開発を進めるアジャイル開発、作らないで使うことを基本とするクラウド・コンピューティング、パラメータを変更するだけの自動コード生成やビジネスルールマネジメントシステム (BRMS) の利用などである。このように、ICT システムの対象やそれを使うビジネスの状況に応じ、最適な開発スタイルは異なる。もちろん、1つのシステム全体を単一の手法で開発することが適切ではない場合もある。本特集では、それらの中から、アジャイル開発、リーン開発、超高速開発、派生開発を取り上げ、それらの動向についてそれぞれの専門家に寄稿していただいた記事を掲載する。

### (1) アジャイル開発

アジャイル開発は、ビジネス環境の変化に対する俊敏なシステムの対応が求められる場合の開発手法である [6]。シ

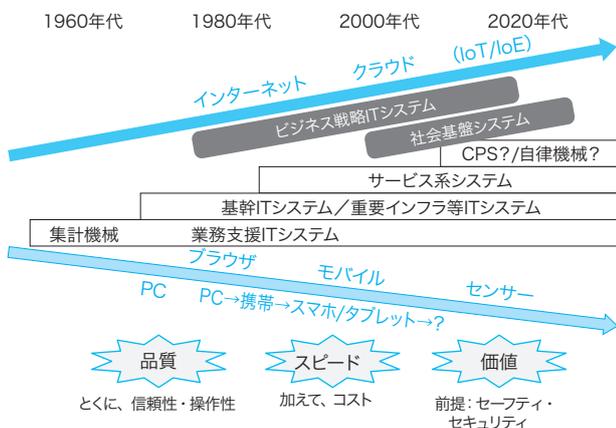


図1 ICT活用とICTシステムの変遷

システム全体の機能を小さな単位に分割し、ビジネス要求に従って優先度の高い機能から短い周期で繰り返し開発し、順次リリースしながらシステム全体を構築していく。

## (2) リーン開発

リーン開発は、トヨタ生産方式を源流とし、ソフトウェア製品を素早く提供するために、「ムダの徹底的な排除」や「品質の作り込み」、「全体を最適化する」などの原則に加え、現場で働く人々が考える環境の醸成、人間性尊重の概念も重要な基盤としている [7]。ただし、固有の手法またはプロセスがあるわけではない。

## (3) 超高速開発

超高速開発は、自動コード生成や BRMS などのツール類を活用してソフトウェア開発期間を著しく短縮することにより、企業のスピード経営の実現を目指す考え方や取り組みを指す。その代表的な推進団体は、超高速開発コミュニティ [8] である。

## (4) 派生開発

派生開発は、既存のソフトウェアに変更・追加・削除を行うことにより新たなソフトウェアを作るという保守開発を指し、主に組込みシステムの分野で採用されている。代表的な手法では、適切なドキュメンテーションとトレーサビリティを利用する。

# 3 IoT 時代に向けた課題

IPA/SEC では、主にソフトウェア開発時における先進的な設計・検証技術の適用事例を数多く収集し、それらを整理して公開している [9]。筆者らは、これらの情報及び内外の公開情報を通して、システム開発・運用時に考慮すべき要素として、アーキテクチャ、エンジニアリング、環境・ツール及びマネジメントの 4 (3+1) つを挙げる (開発スタイルはエンジニアリングに含まれる)。そして、これらの要素は相互に深く関連することから、それぞれ整合していることが重要である (図 2 参照)。

例えば、複数の小さな“サービス”を組み合わせてアプリケーションを構築する「マイクロサービス」[10] と呼ばれるソフトウェア・アーキテクチャでは、各サービスは独立にデプロイされ、疎結合で独立動作することから、俊敏な変更が可能であると共に、独立チームによる開発でチームの特徴を活かすことができるため、アジャイル開発に向くと言われる。また、アジャイル開発の生産性を高めるためには、とくに、頻繁なリリースごとに必要となる回帰テストの効率化 (自動化など) が必須であることから、そのためのツールの活用が欠かせない [1]。

このように、効率的なシステム開発・運用のためには、前述の各要素間の整合を図るマネジメントが求められる。適切なマネジメントに必要とされるスキルは、「デザイン

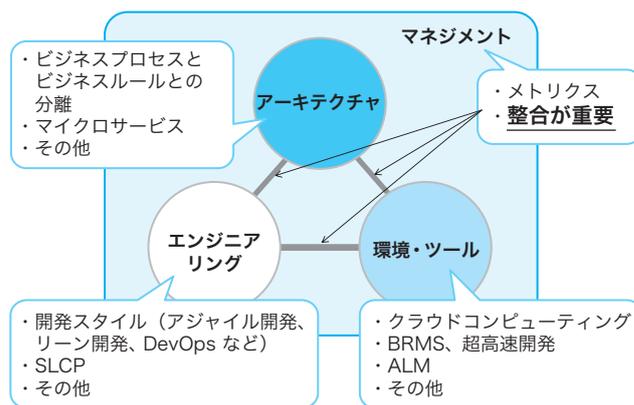


図 2 システム開発・運用時に考慮すべき 3+1 要素

力」である。これは、ICT システムの対象や、それを使うビジネスの状況などを俯瞰し、採用するアーキテクチャ、開発スタイルやプロセス/プラクティスの組み合わせ、及び環境・ツールを決定する能力のことである。

あらゆるモノがインターネットに接続する IoT (Internet of Things) / IoE (Internet of Everything) の時代が始まり、今後ますます進展すると予測されている。このような時代で生き残るには、デジタルビジネスを一層推進しなければならない。デジタルビジネスを持続的に発展させるためには、ICT システムの開発・運用に、優れた「デザイン力」によるマネジメントが重要となる。

最後に、組織・人材について考えを述べておきたい。IoT 時代におけるビジネスレイヤの人材には、IT 知識を背景に革新的なビジネスを創出するスキルが求められる。一方、IT レイヤの人材に求められるスキルは、更に 3 層に分けられる。ビジネスレイヤに近い層から順にそれぞれ、ビジネス知識を背景に適切な ICT システムをデザインするスキル、ビジネスニーズに対応して開発スタイルをデザインするスキル、エンジニアリングとツールなどを駆使してシステムを開発するスキルである。IoT 時代には、全体を見極めてビジネスやシステムについて適切にデザインできる、柔軟な組織・人材が求められる。

### 【参考文献】

- [1] ソフトウェアメトリクス調査 2014, 一般社団法人 日本情報システム・ユーザー協会 (JUAS) .
- [2] CHAOS MANIFESTO 2014, The Standish Group International.
- [3] IT 動向調査 2014, 一般社団法人 日本情報システム・ユーザー協会 (JUAS) .
- [4] Ram Chillarege: The Marriage of Business Dynamics and Software Engineering, IEEE SOFTWARE, November/December 2002.
- [5] ソフトウェア開発データが語るメッセージ 2015, IPA/SEC, <http://www.ipa.go.jp/sec/reports/20150925.html>, 2015 年 9 月 25 日 .
- [6] 山下博之, 柏木雅之: 非ウォーターフォール型 (アジャイル) 開発の動向と課題, SEC journal, Vol. 8, No. 4, IPA/SEC, 2012 年 12 月 14 日 .
- [7] 非ウォーターフォール型開発 WG 活動報告書, IPA/SEC, <http://www.ipa.go.jp/sec/softwareengineering/reports/20110407.html>, 2011 年 3 月 31 日 .
- [8] 超高速開発コミュニティ, <https://www.x-rad.jp/>.
- [9] 「先進的な設計・検証技術の適用事例報告書 2015 年度版」を公開, IPA/SEC, <http://www.ipa.go.jp/sec/reports/20151118.html>, 2015 年 11 月 18 日 .
- [10] James Lewis: Microservices, ThoughtWorks, <http://martinfowler.com/articles/microservices.html>, 25 March 2014.

# IoT時代におけるリーンスタートアップ ～ UnlimitedHand の事例から～

H2L 株式会社 代表取締役

岩崎 健一郎

Lean Startup とは、①作る ②測る ③学ぶのステップを少ないリソースで高速に繰り返す事業開発手法である。これは IoT 時代の製品開発手法と相性が良く、①作る においてはオープンソースを、②測る においてはクラウドファンディングを活用することができる時代となった。弊社事例を交えながら紹介する。

## 1 はじめに

IoT の時代において、製品や事業の開発をどのように進めるべきか。弊社では、Lean Startup = Rapid Prototyping の手法を活用して、新製品 UnlimitedHand の開発と、クラウドファンディングを用いた資金調達及びマーケティングを行った。そこから得た学びを、次の製品や事業の開発につなげる試みを行っている。

このような継続的な「カイゼン」を行うには時間もお金もかかると思われがちだ。しかし、プロトタイピングの手法を用いれば短期間、低予算で開発とマーケティングを行うことができる。あとは、いかに素早く学びを得るかにかかっている。

世界を変革する製品には、失敗がつきものである。独創的な発想を「失敗」の一語で終わらせず、市場ニーズに沿う形まで継続的に進歩させる努力が必要である。市場の声を聞きながら、ぜひ製品と事業の開発を続けていきたい。

## 2 製品開発と事業開発における共通点

### 2.1 製品開発 : Rapid Prototyping

Rapid Prototyping (ラピッドプロトタイピング) とは、製品開発で用いられる用語で、実働するモデルを短期間で製作する手法のことである。概念実証 (Proof of Concept) となる最小限の製品開発を行い、ユーザーからのフィードバックを得て柔軟にプロトタイプを改良する

ことで、最終製品に近づけていく開発手法である。①開発 ②発表 ③研究の3つを繰り返すことで、本質的な機能に注力した製品を目指すことができる。

### 2.2 事業開発 : Lean Startup

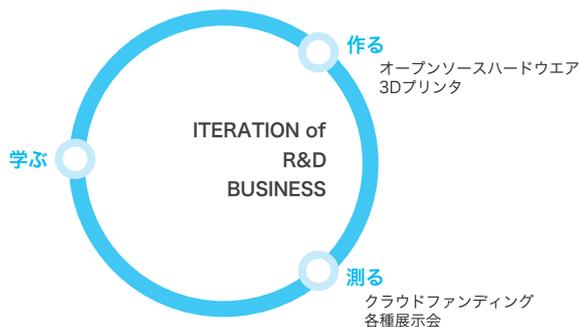
一方で、事業開発で用いられる用語に Lean Startup (リーンスタートアップ) がある。これは必要最低限の機能を持った製品 (Minimal Viable Product) を素早く開発し、ユーザーの反応を見て当初の事業アイデアの軌道修正を行う事業開発手法である。①サービス試作 ②ユーザー評価 ③ビジネス検証 の3つを素早く繰り返すことで、需要のある事業を無駄なく開発することができる。Lean とは「無駄がない」という意味で、1980 年代にアメリカで体系化された Lean Manufacturing (リーン生産方式) から名付けられたものである。もともとはトヨタ生産方式に代表される、日本の自動車産業における生産方式が研究されたものである。このように製造業の盤石な下地が日本にあることは、IoT 時代においてとくに留意すべきだろう。

### 2.3 Rapid Prototyping と Lean Startup の共通点

製品開発の Rapid Prototyping、事業開発の Lean Startup いずれも共通して言えることは、「①作る ②測る ③学ぶ」の3工程をできるだけ迅速に、継続して繰り返すことである。①作る においてはオープンソースハードウェアや3D プリンタの登場により、数日～1週間程度のリードタイムで製品試作ができる環境が整っている。また、②

測る においてはクラウドファンディングの登場により、非常に低いリスクでマーケティングができるようになっている。以下では、仕様変更や市場ニーズに合わせた柔軟な開発ができるという意味合いを込めて、このようなサイクルによる製品と事業の開発を総称して「プロトタイプイング」と呼ぶことにする。<図1>

Prototyping Product and Business  
IoT時代の製品と事業のプロトタイプイング



IPA/SEC, 岩崎健一郎, 2015.11.27, COPYRIGHT © H2L Inc.

図1 IoT時代の製品と事業のプロトタイプイング

### 3 事例紹介: UnlimitedHand

#### 3.1 PossessedHand から UnlimitedHand へ

弊社では2012年より、前腕に電気刺激を与えることで望みの指を動作させる装置 PossessedHand の開発と販売を行っている。これを用いることで、ユーザに手指の動作や擬似的な触感を伝えることができる。本装置は研究用の機材として販売しており、脳科学、基礎心理学、



図2

人間工学を中心とする様々な分野の研究者に活用いただいている。<図2>

PossessedHand は研究者だけでなく、新しい技術に興味のある層からもニーズがあった。しかし、予算の限られたベンチャー企業で、市場性の無い製品を事業化することはできない。そこでターゲットとしたのが、近年急速な高まりを見せるバーチャルリアリティ市場である。Oculus社がFacebook社に20億ドルでM&Aされてから、各社から様々なHead Mounted Display (HMD) が発表された。これらは「見る」ためのデバイスであり、バーチャルリアリティ空間を「触る」ことができない。PossessedHand を応用すれば、バーチャルリアリティ空間に手指の動きを入力し、電気刺激により指を動かすことで、仮想的な触覚を与えることができる。これをUnlimitedHandと名付け、プロトタイプイング手法による製品と事業の開発を行った。

#### 3.2 UnlimitedHand ダーティモデリング

IoT時代の開発は、変化する技術仕様や市場ニーズに対応するため、柔軟なものである必要がある。アジャイル開発手法を参考にして頂ければご理解いただけと思うが、開発が柔軟であるためには、設計や仕様といった概念からいったん自由になり、ユーザのニーズや課題のみに集中する必要がある。

そのため、我々は最初のステップとしていったんコンピュータから離れ、ダーティモデリングと呼ばれるプロトタイプ手法を実践した。これは紙とペンを使った「経験のモデリング」である。まずはユーザの利用シーンをスケッチに描き、そこから必要な機能を抽出した。次に紙を切り、時に粘土やスタイロフォームを使って、電気の流れないモックアップを作成した。このモックアップを完成品とみなして、実際に使っているつもりになる「ごっこ遊び」を徹底的に行ったのである。

これはIoT時代において、センサや外装といった仕様を決めていくのに不可欠なステップであると考えている。モックアップを使って実際の使用感を試しているうちに、当初想定していたセンサを用いる必要がなく、より安価なセンサで代替できることがわかった、という例は数多い。また外装について、とくにウェアラブル機

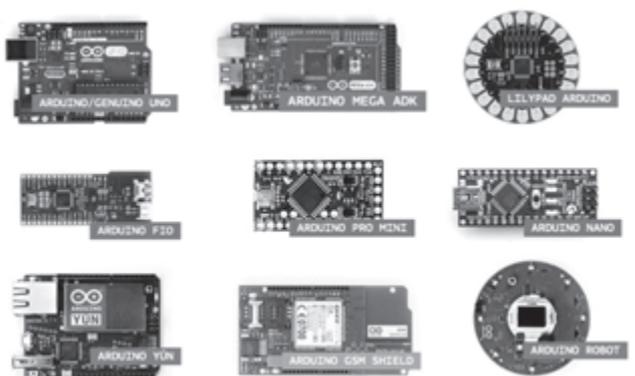
器では実際に身体に装着してみて得られる知見は多い。UnlimitedHandは当初 PossessedHand で使われていたようなベルト状となる予定であったが、ゲームに必要な筋肉では腕の全周を覆う必要が無いことが分かり、パッド状が最適であるという結論に至ったのである。

このように、ダーティモデリングに最も小さな形でプロトタイピングの①作る②測る③学ぶのステップが集約されていることが分かる。「小さく」「素早く」というのが Lena Startup = Rapid Prototyping の本質である。本格的なプロトタイピングのループに入る前に、このような「経験のモデリング」を行っておくと後々の手戻りや失敗を最小限に抑えることができる。

### 3.2 UnlimitedHand ①作る

必要な機能と形状がある程度固まったところで、開発に入った。基礎となる PossessedHand に、ハンドジェスチャの入力をするための回路を追加した。ここで用いたのがオープンソースハードウェアである。オープンソースハードウェアとは、オープンソースソフトウェアの概念をハードウェアに援用したもので、回路図、プリント基板のレイアウトを無償で利用することができるハードウェアである。最も有名なのは Arduino である。これは回路図、プリント基板レイアウトに加え、プロセッサ上で動作するソフトウェアや、ソフトウェア開発環境までをすべてオープンソースにしているハードウェアである。改変が可能のため、様々な派生基板が登場している。<図3>

Open Source Hardware : Arduino  
オープンソースハードウェアの代表 Arduino



出典: <http://arduino.cc>

図3 オープンソースハードウェアの代表 Arduino

このようなオープンソースハードウェアを用いることで、開発の低コスト化と高速化を図ることができた。概念実証となる試作を1週間単位で繰り返し、最終的な製品アイデアを3カ月程で試作した。<図4>



図4

### 3.3 UnlimitedHand ②測る

試作がある程度形になった段階で、市場の声を聞くため、クラウドファンディングと展示会への出展を行った。クラウドファンディングは、不特定多数の人々から資金を集める方法である。多くの場合は目標額と期間を設定し、インターネット経由でプロジェクトの宣伝を行う。集まった資金が設定した目標額に到達すればプロジェクトが開始できる。到達しなければ返金が行われるため、非常に少ないリスクで資金調達及び事業性の評価を行うことができる。

また展示会への出展は、IoT時代のマーケティングにおいては非常に重要である。それは実際にユーザが手に取って試せるだけでなく、1対1でユーザの声を聞くことのできる貴重な機会だからである。

UnlimitedHandの場合、2万ドルの目標額を設定した。クラウドファンディングを開始する前に展示会（東京ゲームショー）でポストカードを配布し、ピッチイベント（TechCrunch Disrupt San Francisco）にてクラウドファンディングのローンチをアナウンスした。目標額は20時間程で達成し、60日の期間中に7万5千ドルの資金を集めることができた。支援者は北米を始め欧州からアジアまで、世界中から300名を超える人々が集まり、

資金調達だけでなく、製品開発にあたって様々な助言をいただくことができた。製品をこのようにして欲しい、という具体的な改良提案もあれば、製品を海外に輸出するにあたって必要な手続き(CEマーキングやFDAといった認証取得)に関する助言もいただくことができ、大変有益であった。

またこの間、様々な展示会に出展してデモを行い、クラウドファンディングの宣伝やユーザーの調査を行った。デモを始めると長蛇の列ができるほどであり、この製品が様々な方の興味を引いていることが分かった。

### 3.4 UnlimitedHand ③学ぶ

このようにクラウドファンディングや展示会で ②測る を進めていく中で、「バーチャルリアリティを触ることができない」という課題は世界共通で共感されるものであることが分かり、事業化を進める意思決定を行うことができた。また、海外展開にあたりどの地域をどの順序で開拓するか、という調査も行うことができた。ここまで使った予算は製造業としては驚く程少ないものである。

最も大きかった学びとして、クラウドファンディングの支援者から、追加機能の要望や製造にあたっての注意点をいただくことができた。一方で、ユーザーにとって必要と我々が当初考えていた機能は、反響を見ると必ずしも必要とされていないことがわかり、無駄な開発を避けることができた。

現在はこういった学びを、最終製品や次の事業の開発に活かしている最中である。

## 4 おわりに ～IoT 産業クラスター形成に向けて～

### 4.1 シリコンリーフとプロトタイピング協会

シリコンバレーのような産業クラスターを日本に創出するのは、起業家のみならず日本人全体の夢であろう。100社に1社しか成功しないと言われる中で、失敗を恐れない起業家が次々と現れて生態系を構築する社会、これを私たちは珊瑚礁になぞらえて「シリコンリーフ」と呼び、2012年から官民でのプロジェクトを進めている。

その一環として、「プロトタイピング講座」という市

民講座を開催している。これは半年間、12回の講座で電子工作未経験の方をIoTエンジニアにする講座である。①電子工作とプログラミング ②展示会出展 ③ブレインストーミングを学び、①作る ②測る ③学ぶの一連の流れを経験してもらうことを特徴としている。毎年1回開催しており、老若男女問わず、下は高校生から上は退職後の方まで、4年間で50人以上が受講している。

有識者と講座卒業生からなる「プロトタイピング協会」も2014年に設立し、プロトタイピング手法の普及活動を行っている。(http://proto-typing.org)

### 4.2 挑戦のすすめ

UnlimitedHandは、はじめは紙と粘土によるダーティモデリングに始まり、世界中から反響をいただくような大きな製品に成長した。このように、世界的に訴求する製品を生み出すためには、目先の利益にとらわれない長期的な観点からの研究開発が必要である。

海外のピッチコンテストに出場すると、日本のベンチャー企業のプレゼンスの低さに驚かされる。原因の1つは、昭和後期からの「失敗が許されない文化」があるようだ。クラウドファンディングを行う際に、外部の方から「失敗したらどうするのか」という戒めもいただいた。しかし、製品開発も事業開発も、人真似でないことをやる以上は失敗して当たり前である。「①作る ②測る ③学ぶ」を意識して、いかに素早く学びを得て、次のサイクルに活かすかが肝要であろう。学びを得て再挑戦し続ける限り、失敗することはリスクではない。挑戦せずに人真似の製品開発で済ませ、少しずつ凋落していくのが最も大きなリスクであることを、忘れてはならない。

ついに日本の強みである製造業が活かせる、IoTの時代が到来した。若い世代は失敗を恐れずに、ぜひ世界初のIoT事業に挑戦していただきたい。またリーダー世代の方々には、ぜひ積極的な研究開発投資に挑戦して頂きたい。失われた20年から得た学びを活かし、積極的に①挑戦し、②失敗し、③学びを得て、また挑戦しよう。

# 変化を味方につけるアジャイル開発

株式会社永和システムマネジメント  
株式会社チェンジビジョン

平鍋 健児

「つながる時代」が到来し、新たな知識を獲得しながらシステム開発を進めるアジャイル開発の有効性が注目されている。要求の事前固定が難しい開発、新技術を含みトライ＆エラーを必要とする開発、使いながらユーザー要求を発掘していく開発などに適した手法としての認知を得てきた。欧米手法の輸入と考えられがちであるが、リーンの源流である TPS（トヨタ生産方式）、スクラムのオリジナルコンセプトである 80 年代の日本の新製品開発手法など、アジャイル開発には日本発の考え方も多く入っている。本稿ではアジャイル開発の概要や求められる背景、スクラムを例とした開発手法などを紹介する。

## 1 アジャイル開発とは

アジャイル開発は 1 回の開発期間（イテレーションやスプリントと呼ぶ）を短く設定した、繰り返し型の開発手法で、優先順位に基づいて動くソフトウェアを徐々に成長させていくという特徴がある。アジャイル開発それ自身は具体的な開発方式ではなく、スクラムや XP といった手法を傘下に置く大きな括り言葉であり、「アジャイル宣言」<sup>※1</sup> をよりどころにしている。

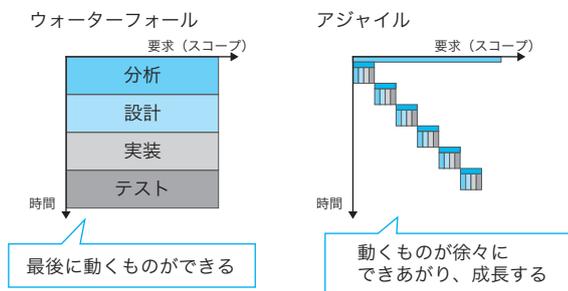


図1 繰り返し型プロセスとしての Agile

単純化して比較すると、ウォーターフォールでは最初に必要なすべての要求を定義し、設計、実装、テストと順に開発する。そのため、動くものが確認できるのは開発の最後である。これに対しアジャイル開発では短いサイクルでテストまで終了した利用可能な状態のシステムを、徐々に積みあげていく。伝統的な V 字プロセスに対応させると、「要求を一掴みとり、高速に V 字を移動させ、受け入れテストまで通過させてしまう」という動きを繰り返すことになる。

アジャイル開発の特徴はその開発サイクルが短いというだけでなく最大の利点として、ビジネス環境の変化に合わせて変化するシステムへの要求に対し、開発する製品やサービスに柔軟に素早く対応することが可能な点にある。また、短期開発から得られた経験に基づき、次の計画を見直ししていくことを繰り返すため、製品もチームも改善が継続的に実施されるのがアジャイル開発の特徴である。

## 2 アジャイル開発が求められる背景

アジャイル開発の有効性が注目されている背景にはビジネスからの要請がある。これまでシステム開発は、ゴールを想定し要求をまとめて仕様書と共に発注し、請け負った側は仕様書通りに作る。開発時点でシステムやサービスの仕様が明確であり、未知のものが存在しないのであれば問題はない。しかし、ビジネススピードが速くなり、グローバルに競合企業とサービスを競うような状況にあって、開発期間内に要求が変わらないことを保障するほうが難しい状況にある。そのため、ビジネススピードに合わせてソフトウェアを提供する、また、変化に素早く対応してソフトウェアを提供することが求められる。その一つの解決策となっているのがアジャイル開発である。ユーザーと連携しながら要求変更を取り入れながら開発を進め、ユーザーのニーズを理解した上でユーザーの想いに近いソフトウェアを実現するアジャイル開発の特徴が昨今のビジネスからの要請と親和性が高いのである。

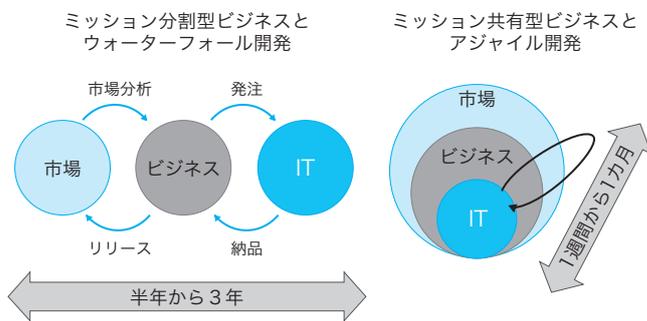


図2 ビジネス構造としての Agile

そのためには、ビジネスの構造にも変革が求められる。市場調査後に要求をまとめ、それを発注し、納品されたも

【脚注】

※1 アジャイル宣言 : <http://agilemanifesto.org>

のを市場提供する、というのが従来型のビジネス構造である。ここではシステムの要求を定義すること（発注側）、仕様書通りに作ること（受注側）、にミッションが分断され、アジャイル開発は機能しない。これは、ユーザ側とベンダ側に分かれた日本のシステム開発でのアジャイル採用の難しさを示している。ミッションを共にした一体感を持ったチームづくりが難しい。自社サービスを保有し自社にエンジニアを雇用している Web サービス事業者でアジャイルが浸透しやすい理由もここにある。自社サービスの成功ために、ビジネスとエンジニアが一体となった開発チームを作りやすい<sup>※2</sup>。

### 3 スクラムとは

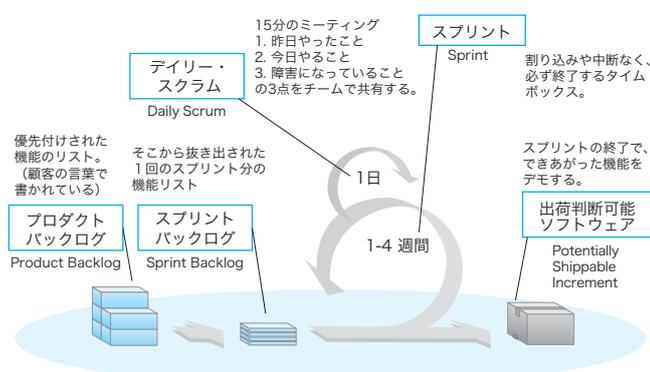


図3 スクラムのフレームワーク

アジャイル開発の代表的な方法論であるスクラムに基づいて開発の進め方を紹介する。

スクラムでは、優先づけされた要求のリストを「プロダクトバックログ」と呼び、1週間や1カ月といった比較的短期間（スプリントと呼ぶ）で開発のループをまわす。プロダクトバックログから1スプリント分を先頭から抜き取って、これを「スプリントバックログ」と呼ぶ。1スプリント分のアウトプットが「出荷判断可能ソフトウェア」、すなわちテストまで済んだ動くソフトウェアである。スプリントをまわすごとにどんどん成長していく。短期間での開発なので、毎日朝会（「デイリー・スクラム」）を実施し情報共有すると共にチームで話しながら日々の問題点を解決していく。

スクラムチームの構成は、(1) プロダクトバックログとその優先順位に責任を持つ「プロダクトオーナー」、(2) 開発者、テスター、Web デザイナー、品質保証といった開発に関係する「開発チーム」、そして、(3) 全体のプロセスをうまく動かしていくための支援をする「スクラムマスター」、これら3つの役割から成る。言い換えるとスクラムマスターが従来のプロジェクトマネージャを置き換え、プロダクトオーナーがビジネス側の仕様決定者、開発チームが開発者となる。実際にはチームは自己組織化されており、例えばスクラムマスターはタスクを個々の開発者にアサインしたりしない。チームが自分たちで決めて実行する。更にペアを

組んで開発をする、タスクボードを使って日々の開発を透明化するなど、様々なプラクティスがそこには含まれている。

スクラムマスターは開発全体を俯瞰し開発を成功に導く。プロダクトオーナーは、開発のROIを最大にすべく、単なる要求の優先順位づけだけでなく、製品やサービスがどういう姿になりたいのか、その最終目的は何なのかというビジョンを語り、チームとコミュニケーションして要求とユーザに関する知識をチームにインプットする大切な役割を持つ。

スクラムではプロジェクトの進行と共に、フィードバックを受けながらプロダクトバックログに入っている機能の詳細化、見積もり、追加や削除、優先順位の見直しが順次進められる。並び順が上のアイテムほど明確で詳細化された状態になっている。スプリントを開始する直前にスプリント計画会議を開催し、今回のスプリントで開発する内容を確定させ、プロダクトオーナーと開発チームで合意してスプリントが始まる。

スプリント期間中は、15分のデイリースクラム（朝会）というミーティングを毎日開催し、情報共有の場を持ち、メンバ全員が互いの状況を共有する。フェイス・トゥ・フェイスで全員参加のミーティングには、作業進捗だけでなく、その日の体調やチームを取り囲む様々な状況を把握できるというメリットがある。上司への報告ではなくチーム全体への報告というスタイルとなるのが特徴である。この時、バックログを壁と付箋で表現して視覚的に短時間で最新状況を共有するためのタスクボードなどが用いられる。

スクラムでは、スプリント中にどのような作業プロセスで仕事を進めるかは定義されておらず作業のやり方は開発チーム自身で決める（後述の「プラクティス」参照）。

スプリント完了時にスプリントレビューを開催し、開発した結果についてステークホルダにデモを行い、フィードバックを得る。フィードバックされた結果はプロダクトオーナーを中心に検討され、必要があればプロダクトバックログへ反映する。また、このスプリントレビューとは別に、次のスプリントをよりうまく、楽しく進めるためにふりかえり（スプリント・レトロスペクティブ）を行う。開発したソフトウェアに関してではなく、スプリントで得られた知識を集約して開発プロセスやプラクティスを改善する。実際に開発作業をしている現場の人が、手を動かして気付いたことを自分の活動の中に取り入れていく、プロセス改善がプロセスの中に含まれており、自分たちのプロセスを作っていくことがアジャイル開発の大きなポイントである。

### 4 アジャイル開発のプラクティス

このように、スクラムには「プロセスの枠組み」のみが規定されており、実際に現場で行われるソフトウェア開発

**【脚注】**

※2 IPA 調査参照 <http://www.ipa.go.jp/sec/softwareengineering/reports/20120611.html>

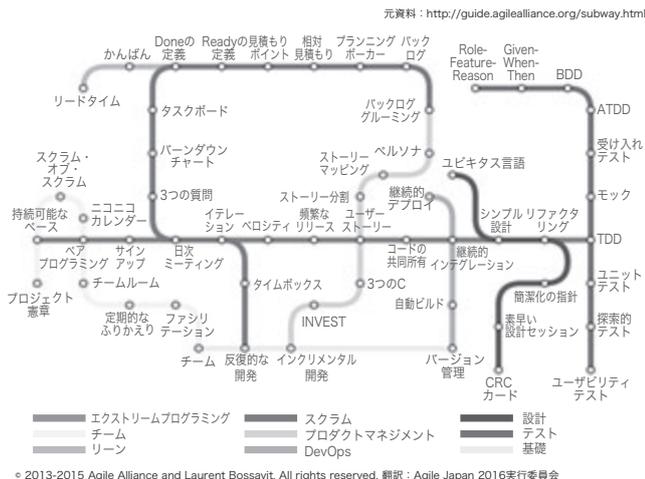


図4 アジャイルプラクティスの地図

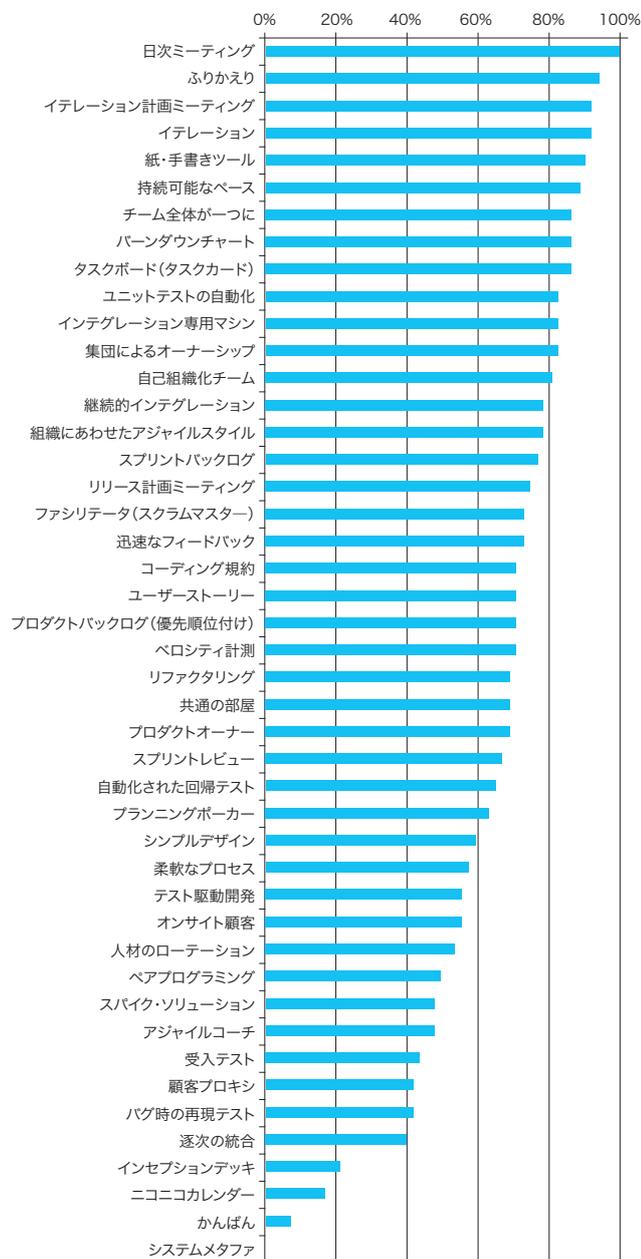


図5 日本でのアジャイルプラクティスの採用状況<sup>※4</sup>

中大規模特有の工夫		小規模と同様だがとくに注意して実施する工夫
<b>組織体制</b> <ul style="list-style-type: none"> <li>チーム間ローテーション</li> <li>コミュニケーション</li> <li>段階的朝会</li> <li>チーム跨ぎのふりかえり</li> </ul>	<b>品質</b> <ul style="list-style-type: none"> <li>第三者テスト</li> <li>部分適用</li> <li>必要な部分のみ適用</li> <li>疎結合なチーム</li> <li>工程の見える化</li> </ul>	<b>コミュニケーション</b> <ul style="list-style-type: none"> <li>完全透明性</li> </ul> <b>展開</b> <ul style="list-style-type: none"> <li>パイロット導入</li> <li>認定研修・コンサル</li> <li>タントの利用</li> </ul> <b>分散拠点開発</b> <ul style="list-style-type: none"> <li>チャットシステム</li> <li>リアルタイムチャット</li> </ul> <b>アーキテクチャ</b> <ul style="list-style-type: none"> <li>アーキテクチャの改善</li> </ul> <b>システム分割/インテグレーション</b> <ul style="list-style-type: none"> <li>疎結合で分割</li> <li>早期からのインテグレーション</li> <li>継続的インテグレーション</li> </ul> <b>品質</b> <ul style="list-style-type: none"> <li>重視するビジネス価値</li> <li>ビジネス価値の変化</li> <li>タイムボックス優先の品質</li> <li>自動単体テスト</li> </ul>
<b>展開</b> <ul style="list-style-type: none"> <li>漸進的な人数増加</li> <li>漸進的な展開</li> <li>社内勉強会</li> </ul> <b>分散拠点開発</b> <ul style="list-style-type: none"> <li>同一拠点から分散へ</li> <li>TV会議</li> </ul> <b>アーキテクチャ</b> <ul style="list-style-type: none"> <li>組織の共通基盤アーキテクチャの利用</li> <li>アーキテクチャについての教育</li> </ul> <b>システム分割/インテグレーション</b> <ul style="list-style-type: none"> <li>同じリズム</li> </ul>	<b>小規模とは逆のアプローチをとる工夫</b>	<b>アーキテクチャ</b> <ul style="list-style-type: none"> <li>最初のアーキテクチャ構築</li> <li>アーキテクチャ専門チーム</li> <li>運用保守チーム</li> </ul> <b>品質</b> <ul style="list-style-type: none"> <li>テスト・フェーズ</li> </ul>

図6 中大規模事例での工夫<sup>※5</sup>

の設計手法、開発環境、見積手法、会議の進め方、テスト技法、などは定義されていない。「アジャイルプラクティス」と呼ばれるこれらの個々の活動は、これまでに方法論ごとに多数提案されており、それに加えて個々の現場で生み出されたローカルなものも多い。実際のアジャイル開発では、開発チームごとにプラクティスを選択、あるいは、編み出して、自分たちで自分たちのプロセスを作り、維持・改善していく（そのエンジンとなるのが、スプリントの最後に行う自己改善活動「ふりかえり」なのである）。

図4は、Agile Allianceが掲載している「プラクティス地図」である<sup>※3</sup>。様々な手法からプラクティスが相互乗り入れしている様子が、地下鉄のマップのように示されている。また、IPAによる日本でのプラクティス活用事例<sup>※4</sup>（図5）には、日本でのプラクティス採用データが掲載されている。スクラム（やXP）の基本的な枠組みに、様々なプラクティスが組み合わされていることが分かる。

また、図6にはチームや組織ごとに工夫したプラクティスが抽出されている。中大規模では難しいと言われているアジャイル開発であるが、こういった現場ごとの工夫があったアジャイル開発が実際に機能するようになる<sup>※5</sup>。

## 5 アジャイルの歴史

アジャイルという言葉は、2001年にユタ州のスノーバードで前述のアジャイル宣言が書かれたのが最初だ。その前には、90年代後半に出てきた手法であるXP（Extreme Programming）、スクラム、更に前にはEvoという、PDCA

### 【脚注】

- ※3 <http://guide.agilealliance.org/> (Agile Japan 2016 実行委員会翻訳)
- ※4 IPA「アジャイル型開発におけるプラクティス活用事例調査」の報告書とリファレンスガイドを公開 <https://www.ipa.go.jp/sec/softwareengineering/reports/20130319.html>
- ※5 IPA中・大規模な開発プロジェクトで非ウォーターフォール型開発を成功させるポイントなどを紹介 <http://www.ipa.go.jp/about/press/20120328.html>

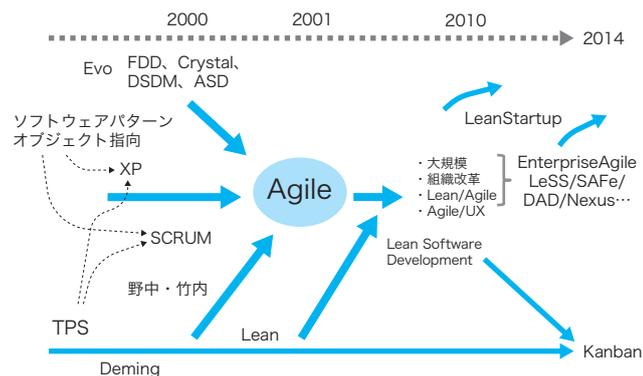


図7 アジャイルの現在位置

を基礎にした繰り返し手法が存在した。これがアジャイルのメソッドとして提唱された最古だろう。後に FDD (Feature Driven Development)、Crystal Clear、英国コンソーシアムの DSDM などがあった。ASD (Adaptive Software Development) は後に『アジャイルプロジェクトマネジメント』を書いた Jim Highsmith の考え方である。このようなメンバがスノーボードに集まって、「アジャイル」(Agile) という言葉を作った。

リーンソフトウェア開発は、Mary と Tom Poppendieck が提唱したが、この夫妻は TPS への造詣も深く、「リーン」と「アジャイル」を結び付けたことで、それまでソフトウェア開発活動とのみ捉えられていたアジャイルが、経営的な視点で説明された。Kanban、というアジャイル手法もその後提案されている。リーンとは他産業適用へ抽象化された TPS であり、著者なりに一言で言うと「顧客の目を見た価値を定義し、その価値を顧客からのプルで流れ化し、改善活動を現場参加で行う」となる。リーンの言葉で経営者に対してアジャイルを語れるようになったというのは大きな変化であり、アジャイルが爆発的に普及する契機になった<sup>※6</sup>。大きなリーンの流れの1つにアジャイルも位置付けて理解できるようになったのだ。

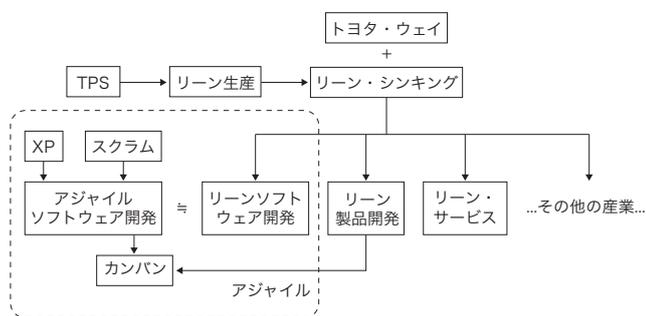


図8 リーンの流れからみたアジャイル<sup>※7</sup>

また、スクラム、という名前は 1986 年に竹内弘高と野中郁次郎による Harvard Business Review の論文「The New New Product Development Game」に由来している。そこでは日本の新製品開発が、職能の垣根を取り払ったチームで行われていること、成果物をバトンパスするリレーでなく、

全員が一丸となってボールを運ぶラグビーのようにプレーしている、という意味で「スクラム」という用語が使われている<sup>※8</sup>。「知識はどこからやってくるのか」という深淵な問いに答えようとする SECI モデルがスクラムの中核にあるのだ。

そして、2011 年には、「リーンスタートアップ」が書かれた。ここでは、米国西海岸のスタートアップ企業が顧客開発と製品開発のループを1つにして、資金を有効に(リーンに)活用しながらスタートアップする手法を記述した。これは、アジャイルの最適な利用法としてのひとつの理想型と言えるだろう。

## 6 アジャイルの課題とエンタープライズアジャイル

現在議論になっているのは、大規模化、分散チーム、オフショア開発、をどう扱うか、更に、企業の予算システムや人事評価との整合どうするのかといった、アジャイルと組織との関連である。それに伴い、「エンタープライズアジャイル」という言葉も生まれた。海外では、エンタープライズアジャイルフレームワーク、と呼ばれる幾つかの手法も提案されている。LeSS (Large-Scale Scrum)、SAFe (Scaled Agile Framework)、DAD (Disciplined Agile Delivery)、Nexus、の4つが現在よく知られている。どれも、アジャイルチームをスケールアップする試みだ。

## 7 最後に

この記事では、アジャイル開発の概要、ビジネス変化の要請、歴史的背景を概観した。とくに、アジャイルはプロセスとして定義されているものではなく、「プラクティス」をチームで選択、工夫しながら自分たちのやり方に仕立て上げる活動が重要であることを強調したい。2016年5月31日には、毎年恒例となっている「アジャイルジャパン2016」が開催される。今年のテーマは「あなたとつくるアジャイル」とした。日本でアジャイルが普及するには、現場での工夫とその事例の共有が必要であるとの思いである。

アジャイルは海外から普及が始まった手法ではあるが、トヨタ生産方式や野中郁次郎の80年代の新製品開発の研究など、コンセプト部分には日本からのアイデアがたくさん入っている。アジャイルは本来、日本が得意とするそのチーム開発のやり方なのだ。どこまで行っても、「ソフトウェアは人が人のために作っている」のであり、そのソーシャルな活動総体のデザインがアジャイルなのである。

### 【脚注】

- ※6 「リーン」と「アジャイル」の関係とは?  
<http://www.atmarkit.co.jp/ait/articles/1311/15/news015.html>
- ※7 『リーン開発の現場』前書きより
- ※8 『アジャイル開発とスクラム』(野中郁次郎、平鍋健児)

# IoT の時代における派生開発の対応

株式会社システムクリエイツ 代表取締役

清水 吉男

世の中は IoT に絡んでビジネスチャンスを探ろうと躍起になっている。だが現実はそのように甘くはない。これまでの派生開発の中で劣化したソースコードは、今後、IoT の要求に継続的に応えていくには新規開発が大幅な作り変えは避けられないと思われる。だがオフショアなどによる派生開発を長く続けたことの副作用として、組織の開発能力は空洞化しており、とてもすぐに新規開発できる状況ではない。

そこで、派生開発に特化した開発プロセスである「XDDP<sup>※1</sup>」を活用して、現状を派生開発で凌ぎながら新規開発の準備をする方法について提案する。

## 1 IoT は時代のうねり

IoT の時代は突然現れたのではない。高速通信の環境が整い、ビッグデータを高速に処理できるクラウドの環境が実用的になった。スマホの裏でクラウドの環境が稼働していることに気づいていない人も多い。

ソフトウェアの開発方法も、2000 年頃から Agile など QCD の同時達成の方法が定着し、それまでの機能の競争から価値の競争に移行してきた。これらの状況の上に、図 1 のように機器とクラウドと IT がつながることで新しい価値を提供しようというアイデアが出るのは自然の流れである。

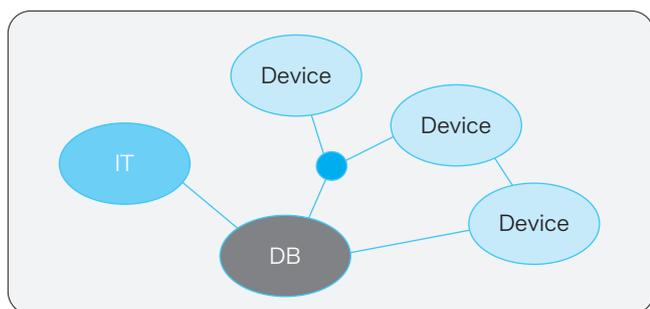


図 1 IoT の構成イメージ

IoT は時代の巨大なうねりである。それを象徴するかのようには、エジソンと共に歩んできた GE が、航空機エンジンなどの製造を関連会社に移し、本体を「ソフトウェア会社」に転換させてしまうほどである。「ソフトウェア」によって価値を提供し続けるのである。

果たして、日本の製造業のトップにこれだけの認識ができていだろうか。これまでの日本では、ソフトウェア（この場合プログラムコードを指す）はハードウェアが提供する機能を稼働させるための手段という認識であった。ハードウェアが「主」でソフトウェアが「従」の位置付けである。

### 1.1 早く買った顧客に後悔させない

だが、IoT の中ではソフトウェアが「主」となり、ソフトウェアの構想を実現するために、ある部分をハードウェアが担当する。もしかすると、既に世の中に出回っているハードウェアから得られる情報を組み合わせて、新しい価値を提供するようなソフトウェアだって出てくるだろう。

ソフトウェアを変更することで提供する価値を変えることもできるし、価値を変化させながら継続的に提供し続けることもできる。それが「Software Defined」である。一部の製品は既にこれを実現しており、自動車だって実現し始めた。これからは「早く買って後悔する」ということはなくなるだろう。

初期の製品には少々使いにくい部分があっても、しばらく待っていれば改良されたソフトウェアがダウンロードできる。買い換える必要はない。買い換えてもらえないということは、競争相手にとって厄介である。しかもプラットフォームを無償提供して「デファクトスタンダード化」すれば、世界中の企業や個人がその製品の上で稼働する新しいソフトウェアを開発してくる。

### 1.2 新しい市場の要求に対応できるか

こうした中では、新規開発時にその後の機能追加や変更を受け入れやすいアーキテクチャを選択すること、多様な要求に応えるためにプログラムコードを劣化させることなく速やかにバージョンアップを提供する仕掛けが求められる。このとき、不適切な変更を加えて混乱させてしまえば他社のソフトウェアに乗り換えられてしまう。

#### 【脚注】

※ 1 XDDP：eXtreme Derivative Development Process の略で、筆者が派生開発に特化した開発したプロセスとして考案したもの。[1]

このように、IoT は新しい時代の到来を実感させる。だが現実問題として、日本の既存の製造業のソフトウェアの開発現場は手放して歓迎できる状態だろうか。このような要求に対応できるのだろうか。

## 2 日本の開発現場が犯した選択ミス

このIoT への対応を考えると、障害になるのは日本の開発組織に「要求にプロセスで対応する」という発想がないことである。DFD<sup>※2</sup>のようなツールを使って、今回のプロジェクトを成功させる合理的なプロセスを設計し、しかも途中で適切に変化させるという発想がないのである。

また、自分たちのプロセスを変化させて、より良い開発手法を取り入れるという発想も見当たらない。その原因を歴史の中で振り返ってみる。

### 2.1 プロセスで対応することに気づかなかった

1980年代、市場は「品質」に対する要求を發した。発端は日本の製品である。その中で、モトローラが日本のページャ(ポケベル)の品質に対抗するために編み出したのが「6Σ」である。ISO や CMM の考え方もこの時期に生み出されている。フィリップ・クロスビーの「Quality is Free (邦題:クオリティマネージメント)」には「最初から正しく仕事をすれば高品質は確保できる」とあり、更に、「品質コスト(予防コスト+評価コスト+失敗コスト)は、ものごとを正しく行わないことの代償である」と喝破した。

このとき日本では、品質は「テスト工程」で工数を投入して実現しており、結局、「プロセス」で対応するという考え方は受け入れられなかった。この選択ミスが、その後に繰り出される市場の要求に対してことごとく的外すことになる。

### 2.2 「自分たちの作業に適合しない」という判断

90年代に入って納期の短縮要求が出たとき、日本では「人海戦術」で対応した。ちょうど、バブルで新入社員が大量に採用されており、必要な教育もそこそこに一人の担当範囲を狭めて現場に投入した。

このとき、世界では市場の要求を満たすために次々とツールが作られてきた。そこで求められていたのは、自分たちの開発プロセスを変更してツールを効果的に使うことであつた。だがこのようなツールも日本ではうまく使えなかった。例えばCASE ツールも、「このツールは我々の開発現場には適合しない」と判断されて導入を見送った組織が多い。

IT 部門でも同じことが起きている。パッケージソフトを導入する際に大幅な「カスタマイズ」を要求した。つまり、自分たちの作業プロセスを変更せずにパッケージソフトを変更させたのである。そこには「プロセスを変化させて納

期を短縮する」という発想は見当たらない。

90年代後半にコストの削減要求が出たときも、単純に「人件費の削減」という発想から、自社のソフトウェア開発部門を子会社化したり、更にオフショア開発へとシフトさせてしまった。「製造」の文化が邪魔をしたと思われる。ソフトウェア開発では、プロセスの工夫と適切なトレーニングによって生産性を何倍も上げることができるが、それを実践した事例がないのだろう。

### 2.3 オフショア開発の代償

派生開発は新規開発よりもはるかに難しい。その派生開発を不用意にオフショアに出したことでソースコードが激しく劣化し、軽微な変更を依頼したつもりでもテストでバグが多発する。そればかりか、それまで開発要員だった人もオフショア先の中継担当になってしまった。新入社員も実際にソフトウェアを開発することなく年数が過ぎていく。組織によっては20年間、まともにソフトウェアを開発していない。当然、そのような組織ではソフトウェアの開発技術力が空洞化しているはずだ。

IT 領域でも、コストの要求から外部のベンダ(SIer と呼ばれている)に丸投げし始めた。その結果、自分たちの業務の要求仕様書も書けないという状態に陥った。更に多段階の下請け構造も助長した。この代償は高くつく。

このように市場の要求への対応のズレは、80年代の選択ミスに起源を發しており、IoT の時代を前にして大きな障害となる。このことを企業のトップは認識しているだろうか。

ロバート・コール氏は「日本の製造業のトップにソフトウェアがわかる人がいないのはなぜか」と問いかけている。コール氏は同志社大学に席を置いて長く日本の製造業を研究されている。確かに、大手の製造業では「プログラムコード」を開発してきた人は、既に役員クラスにいる。だがその人が「ソフトウェアをわかる」人かどうか。

## 3 オフショアからの回帰へ

IoT においては、IT を含めたトータルな開発が必要になる。更に、要求に対して機敏な対応が求められることを考えると、今までのようにオフショアで対応することは難しい。相変わらず、IoT だってオフショアでできるだろうと考えている企業のトップがいるとすれば、時代錯誤も甚だしいと言わざるを得ない。

#### 【脚注】

※2 Data Flow Diagram の略。構造化分析で使われるプロセスの表記ツールで、これを作業や開発プロセスの表現に応用する。筆者は、構造化分析を知らない人にも使えるように表記法を改良して「PFD (Process Flow Diagram)」として公表している。筆者のHP ([http://homepage3.nifty.com/koha\\_hp](http://homepage3.nifty.com/koha_hp) または <http://kohablog.cocolog-nifty.com/>) の「PFDの書き方」を参照してください。

### 3.1 開発力の回復に着手すべき

既に述べたように、日本の製造業では派生開発をオフショアで対応してきた組織が少なくない。オフショアに出していない組織でも長く派生開発で回してきた。その結果、IoT向けの機能を組み入れるにも、ソースコードが劣化していて新しい機能の追加や操作を変更する際にバグが吹き出す。また、RTOSすら搭載されていないものも少なくない。

新規開発と派生開発では目的も違うし、実現できることも違う。一言で言えば、新規開発は勝つための仕掛けを組み込む機会であり、そのためのアーキテクチャ設計のような技術が求められるのに対して、派生開発では多様な変更要求に機敏に対応するための技術、あるいはリファクタリングを織り交ぜて劣化を遅らせる技術が求められる。IoTの時代に漕ぎ出すには、新規開発と派生開発の両方の技術を確保することが求められるのである。

## 4 XDDPでソフトウェアの開発力の回復へ

ソースコードが劣化し、開発技術が空洞化している状況を考えてすぐには新規開発に取り組みない。現実には、XDDPの持ち味を生かしながら、しばらくは現状のソースコードで派生開発に対応しつつ、並行して新規開発の準備を進めるしかない。

### 4.1 XDDPの特徴

XDDPは、派生開発専用のプロセスとして開発されたものである。その特徴を幾つかピックアップする。

- ① 機能追加と変更をそれぞれ異なるプロセスで対応する (図2参照)  
 一般には、機能追加の中で変更作業をしている。しかしながら、機能追加と変更とは求められていることが違う。そのため、変更にとっては不適切なプロセスで作業を行っていることになり、そこからバグが発生している。XDDPはこれを根底から改善する。

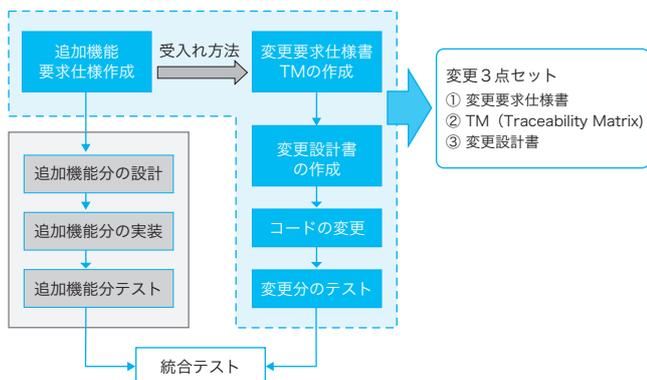


図2 機能追加と変更を別のプロセスで対応する

成果物	カバー範囲	観点が違う 記述の内容	レビュー機会	
変更要求仕様書	What (Why)	今の振る舞いをどのように変更するか なぜ、変更するのか? before/afterで記述する	○	○
TM	Where	変更する仕様がどこにあるか?	○	
変更計算書	How	具体的な変更方法を記述する 原則として before/after で記述	○	○

- ・「変更箇所」を「before/after」で表現することで担当者の理解が見え、レビューによって「部分理解」の状態を緩和する
- ・さらに、「影響箇所」に気づく機会を得てバグも減少する

### 図3 変更3点セットがレビューを可能にする

- ② 「変更3点セット」ですべての変更情報を記述する (図3参照)  
 XDDPでは変更を扱う最小限の成果物として、「変更要求仕様書」「TM (Traceability Matrix)」「変更設計書」の3種類を想定している。もちろんこのほかに必要な成果物があれば、プロセスを設計する段階で追加する。追加機能は「追加機能要求仕様書」にまとめるが、この追加機能を受け入れるための変更<sup>※3</sup>も変更要求仕様書で扱う。これによって、変更3点セットは今回のすべての変更を扱う文書となる。
- ③ 変更の成果物は「before / after」を表現する  
 変更要求仕様書と変更設計書は、変更する個所に対して「before」と「after」の情報を記述することで、担当者が変更箇所を理解し認識している様子を表現することになり、レビューの効果を高める。
- ④ 「変更3点セット」でレビューを可能にする  
 派生開発では「部分理解<sup>※4</sup>」の状態に変更作業が強いられるため、思い込みや勘違いが混じる。だからこそ、レビューが必要なのである。XDDPは「変更3点セット」という成果物を、時間差をもって作成する。そのタイミングに合わせてレビューすることで、ソースコードを変更する前に多くのバグを未然に防止する。

XDDPは、部分理解から発する多くの問題を、「変更3点セット」の作成や、それを使ったレビューなど、最小限の秩序を持ち込むことでバグを大幅に減少させる仕組みを提供する。事前にトレーニングなどの適切な準備を伴うことで、工数も従来比で1/2～1/5にもなり、余った時間で新規開発の準備を並行させることができるのである。

【脚注】

- ※3 一般には、この追加機能を受け入れるための変更がどこにも記述されていないことが多くの問題を生み出している。
- ※4 新規開発と違って、派生開発の変更の担当者は「全体」を理解できる状況にはないことを表している。

空洞化したソフトウェアの設計技術の多くは、派生開発の中でも機能追加の単位で習得する機会がある。この技術は後の新規開発に役に立つ。また、ソースコードを読んで変更箇所や影響箇所を調査する際も、調査結果を表現する方法として設計の表現方法を逆に使用する。これも設計技術の習得につながる。こうして、XDDP の取り組みに目的を持ち込むことで、現状の派生開発の品質を確保しながら、並行して新規開発の技術習得を補っていくことができる。

## 4.2 リファクタリングも XDDP で対応する

一般にリファクタリングは難しいと言われている。それでもソースコードの劣化の状態によっては、今後の変更作業に支障をきたすので「リファクタリング」を行うことになるが、このとき、XDDP が役に立つ。XDDP の枠の中で、今回対応するリファクタリングのパターンを指定し、それを変更要求仕様の形で記述することで、秩序を維持した形でリファクタリングを行うことができる。これは新規開発への時間稼ぎにもなる。

## 4.3 USDM<sup>※5</sup>で要求仕様の作成技術を習得する

派生開発の追加機能に対しては通常のリクエスト書が必要になる。このときの要求仕様書の記述方法を支援するのが「USDM」という表記法である。(変更要求仕様も USDM の表記法を応用している)

以下に USDM の主な特徴を列記する。

- ① 要求と仕様を階層構造で表現する
- ② 要求には必ずそれを必要とする理由をつける
- ③ 機能要求はイベントに始まる一連の振る舞いを一つの要求として扱う
- ④ 要求に含まれる「動詞」をすべて表現することを目指し、動詞に対して仕様グループを設定する
- ⑤ 仕様グループに対して一気に仕様化する

派生開発の追加機能に対して短時間でかつ精度の高い要求仕様を書けることは、ベースライン設定後の仕様変更を減らすことになり、更に、リリース後にもこの機能に関して仕様変更が減ることにつながる。もちろん、この技術は新規開発時の備えになる。

## 4.4 アーキテクチャ設計の準備

派生開発の中では、機能単位の設計技術は習得できるが、アーキテクチャレベルの設計技術は別に習得するしかない。そこで、XDDP で余った工数を充てる。

アーキテクチャの設計は重要である。一般にアーキテクチャスタイルは5種類<sup>※6</sup>ほど知られているが、自社の製品の特性を考慮しながら、IoT の今後の要求にも対応しやすいアーキテクチャを選択したり、組み合わせたりして試作す

る必要がある。また、市場の要求も予測しきれないところがあり、新規開発も比較的短い間隔で必要になる可能性がある。そうすると、「部分的」に作り変える要求に対応できるアーキテクチャも有効である。

保守性のような品質は新規開発時の「設計の中で織り込む」必要がある。USDM では保守性のような「作り方の品質要求<sup>※7</sup>」は作業への指図として記述する方法を提供している。適切に仕様化することで、アーキテクチャと一緒に設計に組み入れることができる。このことは、IoT の時代において有利に作用するはずである。

## 5 遷宮の発想で両方の開発技術を持つ

派生開発に並行して新規開発の準備を進めていく中で、新規開発のプロジェクトを立ち上げる目処が立ったところで、選ばれたメンバを中心に新規開発チームを作って取りかかってくる。新規開発のメンバは何度かに分けて派生開発のチームから補充していく方法で対応できるだろう。

ソフトウェアシステムも、これからの時代の要求に応え続けるには、派生開発だけで回すことは無理だろう。そのとき、新規開発ができる技術者を確保できなければ、その時点で市場からの撤退を余儀なくされる。だから、しっかりした新規開発の技術と機敏で変更ミスをしない派生開発の技術をバランス良く保有していることが勝ち続ける組織の条件になるだろう。

そこで「遷宮」の考え方が参考になる。日本の神社では、壊れていなくても一定の期間が過ぎれば、その時代の技術を織り交ぜて建て直す。新規に建てる技術を失わないための計らいである。若い建築家は解体する際に外から見えなかった技術を学ぶ。ソフトウェア開発でも、派生開発を知ることで、新規開発の「あるべき姿」が見えてくるし、新規開発を仕掛けることで技術の積み上げを図れる。そのためには派生開発と新規開発の人材を交流させることである。

### 【脚注】

- ※5 USDM : Universal Specification Describing Manner の略。要求仕様が無難な表現方法として筆者が開発したものです。[2]
- ※6 「実践ソフトウェアエンジニアリング」ロジャー・プレスマン、日科技連 第10章参照 [3]
- ※7 USDM では、品質要求を「機能を補完する品質要求」と「作り方に関する品質要求」に分けている。

### 【参考文献】

- [1] 『派生開発』を成功させるプロセス改善の技術と極意、清水吉男、技術評論社
- [2] <改訂第2版>要求を仕様化する技術・表現する技術、清水吉男、技術評論社
- [3] 実践ソフトウェアエンジニアリング、ロジャー・S. プレスマン、日科技連

# ユーザーイニシアティブを可能にする 超高速開発

超高速開発コミュニティ 会長  
一般社団法人 ICT 経営パートナーズ協会 会長

関 隆明

## 1 我が国の IT 利活用レベル

日本生産性本部の「日本の生産性の動向」によると、2014年の労働生産性は、OECD加盟34カ国中21位、主要先進国7カ国中最下位と報告されている。

その主たる原因は「規制の多さ」と「ITの利活用レベルの低さ」だと指摘されている。

このような状況下でありながら、設備投資全体に占めるIT投資の割合は、米国、英国が45%に対し、日本は20%に過ぎない。

またJISAの調査ではSE・プログラマの人数は50万人と2005年以降伸び悩んでいる。ITを利用した経営に対する日米企業の比較を見ても、

- ① 情報システム投資の重要性について、米国企業が「極めて重要」と考えている比率が75%に対し、日本企業は16%に過ぎない。
- ② CIOについて、米国企業では経営課題と先進ITを良く理解し、戦略的活用に積極的に挑戦しているのに対し、日本ではCIO自身が不足している状況にある。
- ③ IT投資の対象が、米国では「製品・サービス開発強化」、「ビジネスモデル変革」など「攻めのIT投資」の比率が上位を占めているのに対し、日本では「業務効率化・コスト削減」など「守りの投資」の比率が圧倒的に多い。
- ④ 更に注目すべきは、「新たな技術／製品／サービス利用」に対し、米国企業は積極的活用の意欲が強いのに対し、日本企業は安全性重視で、その活用には消極的な傾向が強い。

今や世界の中での自らの位置を直視し、「日本はIT利活用の後進国」なのだとしっかり自覚し、後れをキャッチアップし、更に一步先んずるための施策を果敢に実行して行かなければならない時だと痛感する。

## 2 日本における IT 化の大きな問題点

企業経営者から「多くの資金と人をつぎ込んで開発しながら、経営に役立っていない」「システム開発の期間が長く、稼働後の経営環境の変化にも改修作業が行けない」などと指摘されている。

筆者はこのようなことが起こる原因は多々あると思うが、大きな本質的な原因として、「ユーザ側のイニシアティブの欠如」と「ウォーターフォール型スクラッチ開発への偏重」を挙げたい。

### (1) ユーザ側のイニシアティブの欠如

日本の経営者の多くはIT投資を経費と見がちで、先行投資とは考えない傾向が強い。従ってIT要員はできるだけ少なく、CIOも置かないケースが多い。米国ではユーザ企業にいるIT人材の数とITサービス企業にいるそれとの比が約2.5対1に対し、日本のそれは逆に1対3となっている。

そのため米国では経営、IT両面の知識と経験を持ったCIOのリーダーシップのもと、自主的にSIベンダやソフト開発企業を選定し、自社の意図するシステム開発に向けて使いこなして行ける。それに対し日本ではユーザ側のIT人材不足と後述するシステム開発の難しさから、システム開発でのSIベンダ依存が強まり、ひどい時は丸投げまで起る。

その結果、ユーザ側の経営からの要求とシステム開発のベースになる要件定義との乖離が起り、結果的にユーザ要求に的確に応えられるシステムの構築ができないということになる。

### (2) ウォーターフォール型スクラッチ開発への偏重

ユーザ企業の個別要求を尊重するあまり、過剰な機能

の取り込みとあまりにも網羅的なシステム開発の必要性から、此の開発方法への拘りが増したと考えられる。

人手による精緻な開発法の特徴から属人性が高く、開発期間の長期化や所要工数の増大、結果として初期投資の増大を招き、さらに手作りソフトの硬さから来る稼働後のシステムメンテナンス費用が高み（年間予算の70～80%）、前向き投資の余裕が無くなってしまった。

このような大きな問題を無くし、システム部門のみにシステム開発を委ねるのではなく、プログラミング技術を持たない管理スタッフや業務遂行者もシステム開発に参加でき、限られたIT要員で有用なシステムを構築できるようにする方法が強く求められている。

### 3 多様化するIT調達法の使い分け

近年のIT技術の急速な進歩により、パッケージソフトの活用、クラウドの利用、BPOなど大変多様化している。個別開発する場合も、従来のウォーターフォール型一辺倒からの脱却が、真剣に考えられるようになって来ており、急速に超高速開発への関心が高まってきた。

フロントエンド系の情報システムには積極的にクラウドでのパッケージソフトを活用すべきであり、バックエンド系の一般の業務でも可能なものは極力クラウドでのパッケージソフトの活用を進めて行くべきだと思う。

問題は基幹業務システムの調達である。図1はITベンダ側から見た「日米のITビジネスモデルの違い」を表したものではあるが、ユーザ側から見た「日米のシステム調達形態の違い」と見ることができる。

- ハードウェア販売でサービス無償が崩壊し、ハードウェア、ソフトウェア、サービスが分離した価格体系へ移行
- パッケージに合わせて利用するより、自社のやり方に合わせ自主開発にこだわる思想
- Sier (System Integrator)、ベンダーに全て任せると、最適なシステムを構築してくれるとの過信
- アウトソーシングの台頭

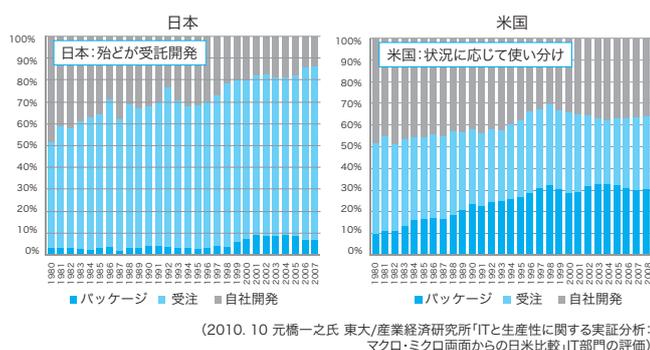


図1 日米のITビジネスモデルの違い

この図中の右端(最新年)の棒グラフを数字で表現すると、下記のごとくなる。

	<パッケージ利用>	<受注=ベンダ開発>	<自社開発>
米国	30%	33%	37%
日本	6%	80%	14%

米国では各形態の比率がほぼ3分の1ずつで、自社開発が37%で最も高く、パッケージ利用も30%に達している。それに比し、日本はベンダによる開発が80%と突出しており、パッケージ利用に至っては6%と極めて低い。日本ではパッケージに合わせて作業するよりも、自分のやり方に合わせ個別開発に拘る傾向が強く、開発はベンダに委ねるという特徴がはっきり出ている。しかしIT要員が限られている日本では、極力業務の標準化を進め、パッケージ利用の比率を大幅に引き上げて行くべきである。そしてプログラミング知識の無いエンドユーザでもシステム開発のできるツールを積極的に取り入れ、一体となってシステム構築のできる体制を作って行くべきだと考える。

### 4 超高速開発を大きく取り上げる理由

当然のことながら私たちもクラウドでのパッケージソフトの利用を大いに推進しようと思っている。

しかし前述のごとく我が国ではあまりにも個別開発されたシステムが多く、パッケージソフトの種類があまりにも少なく、例え使用する場合でもカスタマイジングの度合いが高くなるのが通例である。

本来パッケージソフトはそのまま使うのが原則であり、カスタマイズすればするほど、導入後のリビジョンアップに支障が出てくる。人為的に決められたルールに従って作業する比率の高い会計・財務や人事管理の分野ではその利用度が高いが、人の作業、物の動き、工作機械や運搬機械などの稼働に直接かかわるシステムになればなるほど、パッケージソフトの活用が難しくなってくる。その場の特性に合った個別システムをより早く、安く開発でき、導入後も常に作業のやり方の変化に、スピーディかつ柔軟に対応する必要がある。このような状況下で威力を発揮できるのが、超高速開発ツールである。

### 5 超高速開発ツールとは

図2に示すごとく、業務のデザインから運用・保守工

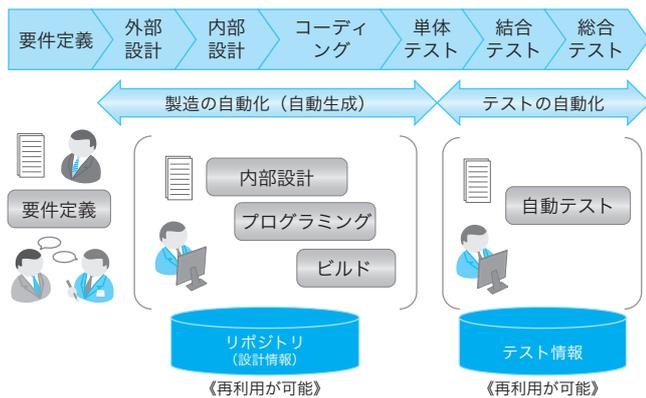


図2 超高速開発ツールの自動化

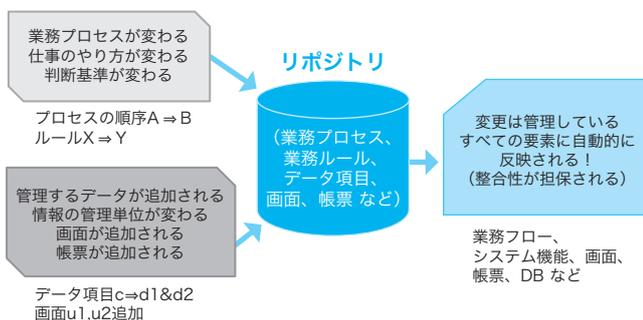


図3 超高速開発ツールはなぜ問題を解決できるのか？  
設計要素の情報をリポジトリに保持

程を含めたライフサイクル全般にわたる生産性向上と継続的品質改善ができるツールである。

システム設計情報をリポジトリと呼ばれるDBに蓄え、それをベースにプログラムの自動生成、データベースの自動作成を行い、できたプログラムの単体テスト、結合テストまで自動的に行える一連のツールを指している。一般的には自動プログラミング・ツールとして注目されているが、それ以上にリポジトリを持つ効果が絶大である。

図3に示すごとく、リポジトリには業務プロセス、業務ルール、データ項目、画面、帳票などすべての業務設計情報が蓄えられている。これをベースに自動プログラミングが行われるのみならず、プログラム作成後、業務プロセスや業務ルールの変更、データ項目の修正や追加などが発生した場合、すべてリポジトリに対して業務用語で修正、変更をすればプログラム、データベースへ自動的に反映され、整合性が担保される。このような機能が単なるプログラムの自動生成以上に、持続可能な業務アプリケーションシステム構築手法として高く評価される所以となっている。

		WF <sup>*1</sup>	アジャイル	xRAD <sup>*2</sup>	アジャイル/WF	xRAD/WF
総費用/JFS	平均	112.19	135.45	40.7	1.21	0.36
	係数	28.2	57.65	6.4		
工数/JFS	平均	1.28	2.15	0.48	1.68	0.37
	係数	0.44	1.6	0.26		
工期/JFS	平均	0.31	0.24	0.1	0.77	0.32
	係数	0.13	0.04	0.03		

・費用・工数・工期ともに超高速開発はウォーターフォール法の3分の1である

※1 ウォーターフォール ※2 Extremely Rapid Development、超高速開発備考

JFS:JUAS Function Scale。画面数と帳票数から規模を換算した値。既存WFのデータを基に画面数+帳票数×2/3を算出(ユーザー発注者が明確に判るのは画面数、帳票数である)。また、係数はグラフにプロットしたときの傾きを示す。xRADは係数が小さいので、規模による生産性の変動が少ない。なお、件数が少ないことと製品価格は考慮していないので注意が必要。

JUASソフトウェア・メトリックス調査2014より

図4 適用の効果

## 6 超高速開発ツールの効果

効果を羅列的に挙げると

- ① 業務ベースでシステム開発ができるため、プログラミング技術を持たない管理スタッフや業務担当者でもシステム開発が可能になる。
- ② 低付加価値のプログラム製造作業からSEを開放し、より付加価値の高い上流工程作業に向けて行くことができる。
- ③ 少人数で効率良く開発ができるため、アジャイル開発の有効なツールとなる。
- ④ 図4に示すごとく、「工期の大幅短縮」や「工数削減」、更に「品質向上による手戻り削減」などにより、総費用の大幅削減が可能(平均3分の1)となる。
- ⑤ 図5に示すごとく経営環境変化にスピーディかつ柔軟に対応できるため、システムライフサイクルの長期化が可能となる。

以上のような効果により、ユーザ企業主体の開発が可能となり、「ユーザイニシアティブ」の確立が可能となる。

## 7 超高速開発ツールの課題

2年半前に超高速開発ツールの啓蒙及び活用促進を図るため、「超高速開発コミュニティ」を発足した当時は、その有効性に疑問を持つ企業が多かった。しかしその後のツールベンダのチャレンジ精神に富んだビジネス展

- 再構築（IT基盤変更等）の都度費用が増大する。古いシステム改修に時間とコストが増大している。
- バックログが増大すると、システムが陳腐化して、利用者がエンドユーザコンピューティングで対応して「見えない」コストが増大している。

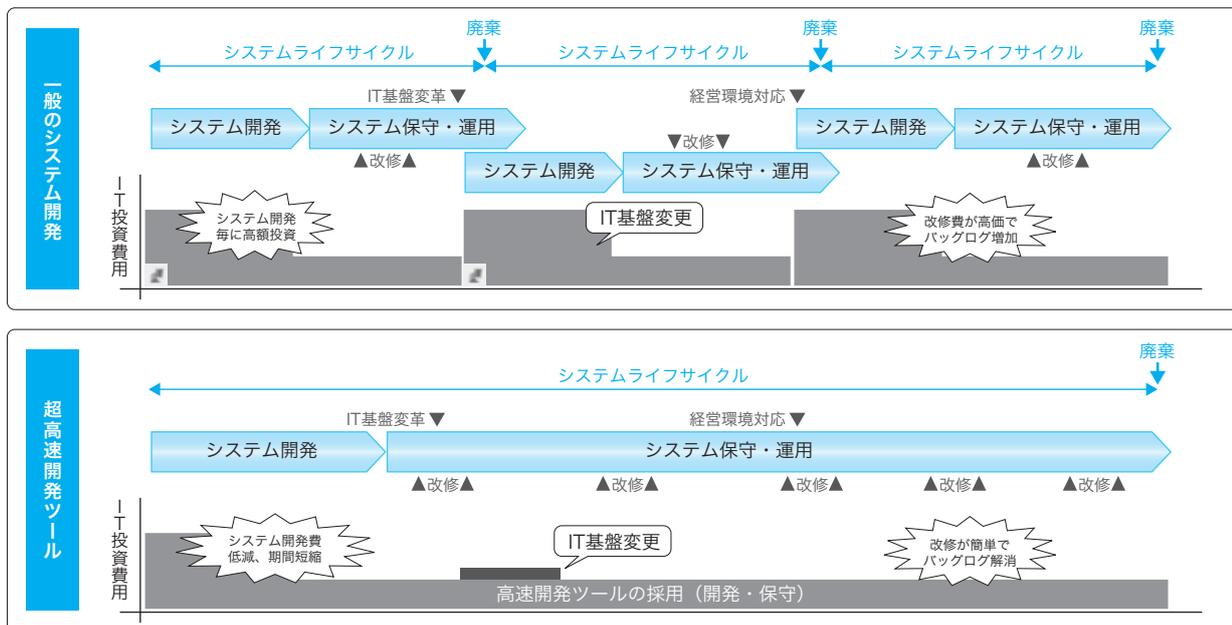


図5 システムライフサイクル

開、当コミュニティによる啓蒙活動や事例による有効性の訴求、さらに図4のごとく、JUASのような中立の団体による他の開発手法との客観的比較評価結果の公表などにより、急速にその評価が高まって来た。

とは言え、まだまだ知名度は低く、これらの活動を引き続き精力的に継続しなければならないと思っている。

これからはITサービス提供企業や各種業界団体も含めて、経営課題解決からITによるソリューション提供まで、連携の取れた支援活動を展開し、超高速開発ツールがいかに有効なツールであるかをアピールして行くことが必要だと思っている。超高速開発ツールそのものの課題としては、まだまだ機能範囲の拡大が必要であることと、それぞれのツールが独立して開発されて来たり、海外から導入されて来たりしているため、各ツール間の連携が取れていないことである。

今後その連携を一段と強化して行く必要があると思う。

又まだクラウドでの使用ができないツールはプログラム開発はもちろん、開発されたアプリケーション・ソフトのクラウドでの運用を可能にしなければならないと思う。

## 8 IoT時代を迎えて検討すべきこと

IoTによる「つながる工場」を例にとれば、従来の生

産管理、製造実行システム（Manufacturing Execution System: MES）及び工作機械やロボットの制御システムを連結することが極めて重要になって来よう。生産管理のごとき情報システムを対象としてきた超高速開発の考え方を、MESや制御システムのアプリケーション・ソフトウェアの開発にまで適用できないものかと考えている。

既に複数の工作機械やロボットを一体的に監視・制御する手作りのMESやセンサーから各種のデータを取り込んでDB化し、そのデータを用いて制御システムや各機器に必要な指示をするためのアプリケーション・ソフトウェアを自動生成する、テンプレート活用型のツールなども出現して来ている。

機械やロボットの制御用組込みソフトは無理としても、センサーからのデータ取り込み以降、MES、生産管理をすべて対象にした超高速開発ツールができれば、極めて有効であろうと思っている。

「もの作り」の世界では「3DCAD + 3Dプリンター」の組み合わせが革命を起こしつつある。

同じように、「アプリケーション・ソフトウェア開発」の世界で、超高速開発ツールの「リポジトリ + 自動プログラミング」の組み合わせで革命を起こせたら素晴らしいと思っている。

# SECjournal 論文賞 受賞論文発表

SECは、我が国ソフトウェア産業発展のための様々な取り組みを実施しておりますが、その取り組みの一つとして、ソフトウェア工学に関する論文を募集し、優秀な論文に対し、表彰を行っております。

今年度のSECjournal論文賞は、2014年7月から2015年6月までに投稿された合計16編のうち、査読者による審査を経て、SECjournalに採録された6編の論文を候補とし、そこから更に選考委員会と表彰委員会による厳正な審査の結果、1編を選出いたしました。

各賞の発表と表彰式は2015年11月19日にEmbedded Technology 2015内で実施いたしました。本年は最優秀賞、優秀賞は該当なし、所長賞1編が選出されました。



上段左より、鶴保 征城 (IPA 顧問)、大原 茂之  
立石 譲二 (IPA 理事)、角田 雅照、片山 卓也、松本 隆明

(敬称略 ※肩書きは  
2015年11月当時)

## SECjournal 論文賞表彰委員会 委員

委員長	片山 卓也	北陸先端科学技術大学院大学 名誉教授
委員	有賀 貞一	AIT コンサルティング株式会社 代表取締役
	岩野 和生	国立研究開発法人 科学技術振興機構 研究開発戦略センター 上席フェロー
	大島 啓二	一般財団法人 日本科学技術連盟
	大原 茂之	一般社団法人 スキルマネージメント協会 理事長
	土井 美和子	国立研究開発法人 情報通信研究機構 監事
	松田 晃一	独立行政法人 情報処理推進機構 顧問
	松本 隆明	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

選考委員会では、全委員の査読結果を含め、対象論文の審査を行った。ただし、委員が著者の論文や委員の関係者の論文については、該当委員は審査を行っていない。

※委員は50音順に掲載。敬称略 ※所属、肩書きは2015年11月当時

## SECjournal 論文賞選考委員会 委員

委員	飯泉 紀子	株式会社日立ハイテクノロジーズ 科学・医用システム事業統括本部 医用システム設計開発本部 医用システム第三設計部 主管技師
	兼本 茂	会津大学 コンピュータ理工学部 教授
	神庭 弘年	一般社団法人 PMI 日本支部 理事
	楠本 真二	大阪大学 大学院 情報科学研究科 教授
	紫合 治	東京電機大学 情報環境学部 情報環境学科 教授
	新谷 勝利	新谷 IT コンサルティング 代表
	寺中 勝美	NTT ソフトウェア株式会社 監査役室 常勤監査役
	古山 恒夫	東海大学 理学部 客員教授
	水野 修	京都工芸繊維大学 情報工学・人間科学系 准教授
	神谷 芳樹	みたに先端研合同会社 本店 代表社員
	峯 恒憲	九州大学 大学院 システム情報科学研究院 准教授
	山城 明宏	東芝ソリューション株式会社 ソリューションセンター ソリューション品質保証部 ソリューション品質企画担当 主幹
	山本 雅基	名古屋大学 大学院 情報科学研究科 附属組込みシステム研究センター 特任教授
	山本 里枝子	株式会社富士通研究所 システム技術研究所 所長
	鷲崎 弘宜	早稲田大学 理工学術院 基幹理工学部 情報理工学科 准教授

# SECjournal 論文賞 表彰委員会審査報告



SECjournal 論文賞  
表彰委員会委員長  
北陸先端科学技術大学院大学  
名誉教授  
片山 卓也

SEC journal は我が国のソフトウェア産業政策の一環として発行されているジャーナルであり、ソフトウェア開発現場での先端技術の実践や開発の報告、論文の掲載などを通して我が国のソフトウェア産業、IT 産業の技術力向上に貢献してきました。そして、そのような論文の中からとくに優れたものを毎年選り表彰を行ってきました。今回は、2014 年 7 月からの 1 年間で SEC journal に掲載された論文を対象に論文賞選考委員会、表彰委員会で審査を行い、以下の論文を優秀論文として表彰することを

決定いたしました。優れた内容のものであると同時に、実際の開発現場における有効性などを評価の主な観点といたしました。

## 「組み込みソフトウェア開発における設計関連メトリクスに基づく下流試験欠陥数の予測」

ソフトウェア開発の初期において、完成したプログラムの中にどれくらい欠陥があるかを、設計文書（上流ドキュメント）や品質保証に関するメトリクスだけを見て予測しようとしたものである。これはできあがったコードを調べて行うものではなく、完成後のテストにかけるべき工数を事前に予測するのに利用できるものである。実際の企業で計測されたデータを用いて検証を行い、設計文書の中でも、基本設計ドキュメントのみを用いた場合でも予測精度にさほど差がなかったことを示している。いかに上流の設計が重要か、その後の工程でいくら品質強化施策を実施しても効果が薄いかということを角田氏は数値的に明らかにした。この点が高く評価された。

## 所長賞

# 組み込みソフトウェア開発における 設計関連メトリクスに基づく下流試験欠陥数の予測

角田 雅照、門田 暁人、松本 健一  
(SEC journal 42 号掲載)

## — 受賞者 コメント —

## 組み込みソフトウェア開発における設計関連メトリクスに基づく下流試験欠陥数の予測

角田 雅照 (奈良先端科学技術大学院大学 情報科学研究科 / 近畿大学 工学部情報学科)

本研究では、近年大きな課題となっている組み込みソフトウェアの品質確保に着目し、開発の設計工程から得られるメトリクスを用いて、テストの後半（結合テスト、総合テスト）における欠陥数の予測を試みた。基本・詳細設計におけるレビュー工数、検出欠陥数に関するメトリクスに加えて、試験仕様書を含む各仕様書のドキュメント量に関するメトリクスを用いて予測モデルを構築した。

分析では、ある組み込みソフトウェア開発企業で収集されたデータを用いて予測モデルを構築した。その結果、どのメトリクスを優先的に用いるべきかは一概にいえなかった



角田 雅照



門田 暁人



松本 健一

が、どのメトリクスを組み合わせただけの場合でも、モデルの予測精度は比較的高かった。このことから、上記のようなメトリクスを収集している企業では、欠陥数の予測を試みる価値があると考えられる。また、基本設計関連のメトリクスのみを使った場合の予測精度が高かったことから、基本設計終了後に予測を試みても、ある程度予測精度が高いことが期待される。

本研究は、共同研究先の企業のご協力により、分析を進めてその結果を発表することができた。深く御礼申し上げます。

# CEATEC JAPAN 2015 出展報告

SEC 企画グループ 主任

荒川 明夫

IPA は、2015 年 10 月 7 日～10 日の 4 日間、幕張メッセで開催された「CEATEC JAPAN 2015」に初出展した。IPA の展示ブースや関連カンファレンスには、延べ 7,300 名の方に足を運んでいただき、大盛況のうちに会期を終えた。その模様の一部を紹介する。

## 1. 展示会概要

CEATEC JAPAN とは、IT・エレクトロニクスを核とする産業のシンボルイベントとして国内外から注目を集め、産業の活力、方向性を表す場として、一般社団法人電子情報技術産業協会（JEITA）、一般社団法人情報通信ネットワーク産業協会（CIAJ）、一般社団法人コンピュータソフトウェア協会（CSAJ）の三者が CEATEC JAPAN 実施協議会として主催し、開催している展示会である。

## 2. 出展概要

IPA は、事業成果の普及・啓発を目的として、様々な展示会に出展しているが、これまでリーチしていなかった新たな事業成果普及対象者層へのアプローチを試みるべく、本展示会に初めて出展し、展示ブースを構えた。



展示ブースでは、主にソフトウェアの信頼性向上、情報セキュリティ対策、IT 人材育成など、安全・安心な

IoT 時代を支えるためのコンテンツのほか、未踏スーパークリエイータのプロダクトなど、IPA の事業全般について、「展示（パネル・デモ・関連資料配布）」、「ブースプレゼン・SEC 先端技術入門ゼミ」にて広く紹介した。

## 3. 展示ブース

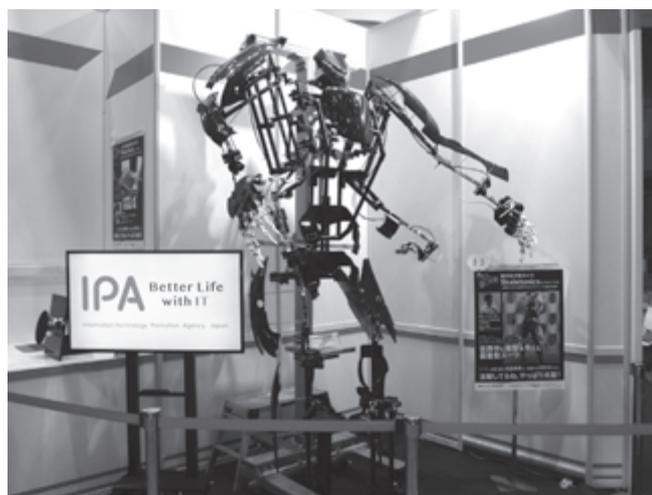
展示ブース内には、それぞれの事業成果を紹介する展示コーナーと 30 分間のプレゼンテーションや講義を行うためのブースプレゼン・SEC 先端技術入門ゼミステージ（32 セッション / 4 日間）を構え、運営した。

展示コーナーでは、「IoT」をテーマに掲げ、IPA で行っている「ソフトウェア高信頼化」「情報セキュリティ」「IT 人材育成」の中から、見た目に分かりやすい事業成果を配置した。中でも、原宿文化展パネルの認知度が高く、来場者がパネルを背に記念撮影をする場面も見られた。



展示コーナーで、とくに注目を集めたのは、未踏スーパークリエイータのプロダクト「スケルトニクス」である。ブースプレゼンの中でデモンストレーションを行った

際、多くの来場者が展示ブースを取り囲んだ。また、報道機関からの取材も殺到していた。



#### 4. CEATEC コンファレンス

本コンファレンスは、国際会議場にある会議室にて、4部構成で実施し、約800名の方に参加いただいた。プログラムは、「安全・安心なIoT社会の実現を目指して」「IoT社会の情報セキュリティ」「IoT時代における創造的人材の発掘と育成」「つながる世界の天使と悪魔」という構成で、こちらも「IoT」をテーマとした構成とし、立ち見が出るほどの盛況であった。



受講者からは、「IPAの取り組みがよくわかった」や「IoTについて理解を深める良い機会だった」という意見が寄せられた。

#### 5. オープンステージ

展示会場内オープンステージでは、IT人材育成事業で

認定している、未踏スーパークリエイターである、筑波大学助教 落合陽一氏とスケルトニクス株式会社 代表取締役 CEO 白久レイエス樹氏が登壇し、自身の事業について、プレゼンテーションを行った。

各講演が15分間と短いながらも、200名近い聴講者が集まって、熱心に耳を傾けていた。

#### 6. 記者発表

10月7日には、国際会議場内プレスブリーフィングルームにて、「IoT時代における『セーフティ設計・セキュリティ設計・見える化』導入のためのガイドブックを発行」と銘打って、IoT時代におけるIT製品やサービスを提供する企業の経営層にセーフティ設計・セキュリティ設計の重要性を啓発すると共に、現場の技術者に対して、セーフティ設計・セキュリティ設計とその見える化を実現するための手法を紹介した「つながる世界のセーフティ&セキュリティ設計入門」の発行について、記者発表を行った。

#### 7. 本出展を振り返って

CEATEC JAPANは、IPAとして初出展ということもあり、今までアプローチできていなかった方に事業成果を届けられたのではないだろうか。また、来場者が我々の事業成果のどんな部分に関心があり、興味を持っていたのかを聴ける貴重な機会でもあった。

展示ブースで来場者の方から多くのご意見をいただいたので、次回出展の参考とし、今後のIPA事業活動に反映していきたい。

CEATEC JAPAN 2015 IPA ウェブサイト

<http://www.ipa.go.jp/sec/events/20151007.html>

- ・ブースプレゼン・SEC 先端技術入門ゼミの講演資料がダウンロードできます（一部）
- ・ブースプレゼン・SEC 先端技術入門ゼミ（一部）の動画を公開しています

# Embedded Technology 2015 / IoT Technology 2015 出展報告

SEC 企画グループ 主任

荒川 明夫

IPA/SEC は、2015 年 11 月 18 日～20 日の 3 日間、パシフィコ横浜で開催された「Embedded Technology 2015 (ET2015)」及び同時開催の「IoT Technology 2015」に出展した。また、隣接会場のアネックスホールでは、IPA セミナーを 2 日間 8 部構成で併催した。

## 1. 展示会概要

Embedded Technology (ET) とは、一般社団法人組込みシステム技術協会 (JASA) が主催する最新テクノロジーの専門展示会であり、組込みシステム開発にかかわる技術者や開発者向けに最新技術などの情報を発信している。

また、今回から IoT 技術の最新動向を発信する展示会として IoT Technology が同時開催された。

## 2. 出展概要

IPA は、2006 年より事業成果の普及・啓発を目的として、ET に出展している。

本年は、IPA 展示ブースでのブースプレゼンを 3 日間で 39 回、SEC 先端技術入門ゼミブースでの講義を 3 日間で 18 回、別会場では IPA セミナーを 2 日間で 8 回、計 65 回のセッションを実施した。

また、SEC の事業を中心に IPA で取り組んでいる IoT や組込み系に関連する事業のパネル展示や資料配布、デモなどを実施した。

## 3. 展示ブース

本出展では、展示ブースでブースプレゼン・SEC 先端技術入門ゼミ・パネル展示・資料配布などを実施し 3 日間運営した。



展示コーナーでは、「つながる世界のセーフティ&セキュリティ設計について」や「高信頼化基盤構築についての取り組み」など、IoT に関する展示のほか、10 月に開催した CEATEC JAPAN 2015 で展示した IPA の各センターの取り組みについても紹介した。



とくに、組込み業界初となる「組込みソフトウェア開発データ白書 2015」を初披露し、来場者から高い関心が寄せられた。



#### 4. IPA セミナー

展示会場に隣接するアネックスホールでは、IPA セミナーを2日間8部構成で開催し、延べ800名の参加があった。

今回のIPAセミナーは、1日目に「IoT時代に求められるソフトウェアエンジニアリング」や「組込みソフトウェア開発データ白書の公開と狙い」と題した講演を行ったほか、平成27年度「情報化促進貢献個人等表彰」（経済産業大臣賞）を受賞された、IPA ソフトウェア・エンジニアリング・センター（現ソフトウェア高信頼化センター）所長などを歴任された鶴保 征城氏による「受賞記念特別講演：2020年に向かっての課題」を実施した。



2日目は、「組込み機器のためのハードウェアセキュリティ」、「最新システム技術の海外動向」、「IoT時代の

システム開発」など、IoTに関する最新情報について、講演を通じて発信した。

#### 5. 組込みIoTハッカソン

提示されたペルソナ（想定ユーザ像）に対して、そのペルソナにとって有効なサービスを提供するシステムを展示会の会期中に組込みIoT技術を活用して開発し、デモとプレゼンテーションで優劣を競う競技である「組込みIoTハッカソン」が実施された。11月18日（水）に課題を発表、展示会会期中に実装して、最終日の20日（金）に参加チームによるプレゼンテーションを実施、審査と優秀チームの表彰を行った。IPAからは、株式会社豆蔵の「進撃の豆ちゃん」チームにIPA賞を贈賞した。

#### 6. 本出展を振り返って

昨年に引き続き、過去の出展の中で、最も多くの方にIPA展示ブースに足をお運びいただいた。その要因の一つはIoT Technologyの併催だと推察され、IoTへの関心の高さが伺えた。

今回は、過去最大規模（小間数）の出展ということもあり、ブースプレゼンとSEC先端技術入門ゼミの同時開催、展示コーナーでは事業成果のパネル展示のみならず、小型ロボットのデモンストレーションや大型パネルの展示など、視覚的にも分かりやすい展示を心がけ、これまでIPAブースを御覧いただいた方にも新鮮に映ったのではないだろうか。来場者アンケートでも高い満足度をいただくことができた。

今後も、効果的で分かりやすい展示を心がけ、取り組んでいく。

ET2015 / IoT2015 IPA/SEC ウェブサイト

<http://www.ipa.go.jp/sec/events/20151118.html>

- IPA セミナー・ブースプレゼン・SEC 先端技術入門ゼミの講演資料がダウンロードできます（一部）
- IPA セミナー・SEC 先端技術入門ゼミの動画を公開しています（一部）

# 工数に影響を与える質的変数とその影響度

東海大学 理学部 客員教授

古山 恒夫



エンタープライズ系ソフトウェアプロジェクトの開発工数に影響を与える質的変数を、新規開発及び改修・保守それぞれについて明らかにした。また業種別の分析を行い、製造業の新規開発プロジェクトの影響要因も明らかにした。質的変数の選択にあたっては2群の平均値の差とそのP値だけでなく効果量も考慮に入れた。

## 1 はじめに

プロジェクト計画時の見積もりや組織レベルのプロセス改善には、考えられる様々な要因の影響を考慮する必要がある。どのような要因がどの程度影響を与えているかを定量的に把握する方法として、専門家の知見を用いた方法や蓄積データの分析による方法がある。

蓄積データから工数に与える影響要因とその程度を調べる方法にも様々な方法があるが、古くから研究され、現在でも多くの研究発表がなされているのが、回帰分析によるものである [Jørgensen 2007]。

実際には、上記の方法を単独で用いることは少なく、幾つかの方法を組み合わせることでより精度の高い結果を得ることが多い。例えば、ソフトウェアの開発工数に影響を与える要因を数式モデルにしたCOCOMO [Boehm 1981]では、専門家の知見から得られた結果を63のプロジェクトの蓄積データで検証し、定量化している。

回帰分析に基づく方法に限らず、コストモデルの構築では蓄積データが重要な役割を果たす。蓄積データのデータ項目数が多いほど、また収集したプロジェクト数が多いほど、更に収集データの値の信頼度が高いほど分析結果の精度は高くなる。IPA/SECが2004年から収集しているエンタープライズ系ソフトウェアプロジェクトデータには3,000件を超えるプロジェクトの多くの質的変数のデータが揃っていて、国際的によく知られているISBSGのデータと同様に多くの欠損値を含むものの統計的に処理できる十分なデータ量がある。

本稿ではIPA/SECが収集したエンタープライズ系ソフトウェアプロジェクトデータ [IPA2014] から、開発工数に大きな影響を与える質的変数を選択し、その影響度を定量化した結果を報告する。質的変数はそれぞれ個別に分析をしてお互いの相関は考慮していないが、それでも最初のステップとしては幾分かの情報を読者に提供できるものと考えている。

質的変数の選択にあたっては、これまで重視されてきた平均値の差の大きさとその有意確率 (P 値) に加えて、効果量も判断基準とした。効果量は各要因単独の目的変数に与える影響度の大きさとそれ以外の要因の影響度の大きさを比較した量であり、たとえ平均値の差が大きくても効果量が小さければ、もっと大きな影響要因がほかにあることになる。今回用いた効果量は、平均値の差を標準偏差で正規化したもので、平均値の相対的な差を表すものである。

今回の分析結果はIT Confidence Conference 2014で報告 [Furuyama 2014] したものと大きな差はない。選択基準によって影響要因としてリストアップされなかったものもそれぞれあるが、各変数の与える影響の傾向が異なるものはない。今回、目的変数を生産性ではなく工数にしたこと、効果量を選択基準のひとつに加えたこと、対象データを1世代新しいものにしたこと、業種別の分析を加えたことが新しいことである。

2章で分析対象データを紹介する。3章で分析方法を、4章で分析結果を示す。5章で分析結果に対する考察を、6章でまとめを述べる。

## 2 分析対象データ

### 2.1 分析対象プロジェクト

分析対象プロジェクトは、IPA/SECで収集したエンタープライズ系ソフトウェアプロジェクト3,546件 [IPA2014]のうち、次の条件を満たすものを対象とする。

- ① 開発5工程 (基本設計、詳細設計、製造、結合テスト、総合テスト (ベンダ確認)) が揃っている。
- ② FPの実測値 (5001\_FP実測値\_調整前) が報告されている。

なお、工数のデータはすべてのプロジェクトが報告している。

この条件を満たすプロジェクトは、新規開発で 574 件、改修・保守では 217 件である。

## 2.2 目的変数と説明変数

### (1) 目的変数

目的変数は工数とする。工数は様々な項目名で収集しているが、工数の値としては「開発 5 工程工数」すなわち、開発 5 工程に含まれる、開発工数、管理工数、その他の工数をすべて加えた総工数をとる。

### (2) 説明変数

ソフトウェアプロジェクトで扱う変数には、比尺度に従うもの、名義尺度または順序尺度に従うものがある。前者を量的変数、後者を質的変数と呼ぶことができる。

量的変数には、工数に最も影響を与える「規模」以外に、工程進捗に伴って得られる中間生産物や作業結果に関するもの、例えば仕様書枚数、レビュー指摘件数、テストケース数、バグ数などがある。しかし、規模以外の変数の値は開発が進んでから得られるものであること、また多くは規模と強い相関があることから、本稿では取り上げない。規模については、開発すべき機能量としての FP 規模を変数とする。

質的変数のうち順序尺度に従うものは、実績の評価などプロジェクト終了後に実施されて値が得られるものを除くと、代表的な変数は 45 個と考えられる。これらの 45 個の変数は、「要求レベル (の高さ)」のようにプロジェクト計画段階で早期にわかるものか、またはツールの利用や作業スペースのようにプロジェクトの遂行にあたってプロジェクト管理者や企業レベルでコントロールできるものである。質的要因の候補としてとりあげる 45 個の質的変数を表 1 に示す。表中の変数の頭に付いた番号はデータ白書で定義されたものである。データ白書では類似の変数を 100 番台が同じものでグループ化しているので、これを参考にしながら、表 1 では変数をその変数の意味を考慮して 5 つのグループに分類している。

名義尺度に従うもののうち、代表的な変数として業種とアーキテクチャを取り上げる。名義尺度に従う変数では、分析に耐えられるだけのデータが揃っている名義のもののみを対象とする。名義尺度に従う質的変数を表 2 に示す。ただし、これらの変数の値は計画段階ではわかっているもののプロジェクト管理者や組織レベルではコントロールできないものであり、分析結果は参考値に過ぎない。

質的変数を説明変数とする場合は、分析結果の頑健性を高めるために、次の条件を満たすものとする。

- ① データ数 (回答数) が 50 件以上ある。
- ② 各レベルに属する回答数はすべて 10 件以上ある。
- ③ 回答の内容が極端に偏っていない。具体的には、偏

表 1 影響要因の候補 (質的変数: 順序尺度に従うもの)

分類	変数 (*1)
開発プロジェクト全般 (11)	111_ 新技術利用 / 112_ 役割分担_ 責任所在 / 113_ 達成目標_ 優先度_ 明確度合 / 1011_ 定量的出荷品質基準_ 有無 / 1013_ 第三者レビューの有無 / 5241_ 品質保証体制_ 基本設計
	114_ 作業スペース / 115_ プロジェクト環境_ 騒音 計画の評価 (120_ コスト / 121_ 品質 / 122_ 工期)
ツールの利用 (11)	302_ 業務パッケージ / 403_ 類似プロジェクト / 404_ プロジェクト管理ツール / 405_ 構成管理ツール / 406_ 設計支援ツール / 407_ ドキュメント作成ツール / 408_ デバッグ_ テストツール / 409_ CASE ツール / 411_ コードジェネレータ / 412_ 開発方法論利用 / 422_ 開発フレームワーク
ユーザ側 (9)	501_ 要求仕様_ 明確さ
	ユーザ担当者 (502_ 要求仕様関与 / 509_ 受け入れ試験関与 / 503_ システム経験 / 504_ 業務経験 / 507_ 設計内容理解度) / 505_ ユーザとの役割分担_ 責任所在_ 明確度合 / 506_ 要求仕様_ ユーザ承認有無 / 508_ 設計_ ユーザ承認有無
要求レベル (8)	要求レベル (512_ 信頼性 / 513_ 使用性 / 514_ 性能・効率性 / 515_ 保守性 / 516_ 移植性 / 517_ ランニングコスト要求 / 518_ セキュリティ)、519_ 法的規制
開発担当者 (6)	601_ PM スキル
	要員スキル (602_ 業務分野経験 / 603_ 分析・設計経験 / 604_ 言語・ツール利用経験 / 605_ 開発プラットフォーム使用経験) 1010_ テスト体制

(\*1)「/」が項目の区切りを表す。番号はデータ白書で定義されたもの

表 2 影響要因の候補 (質的変数: 名義尺度に従うもの)

変数	とりうる値 (*1)
201_ 業種 1	製造業、情報通信業、卸売・小売業、金融・保険業、左記以外の業種
308_ アーキテクチャ 1	スタンダード、メインフレーム、2 階層クライアントサーバ、3 階層クライアントサーバ、インターネット・イントラネット

(\*1) 値ひとつとそれ以外の値すべての 2 つのレベルで比較する。

り率が -0.7 ~ 0.7 の範囲である。ただし、偏り率は、偏り率 = (上位レベルの回答数 - 下位レベルの回答数) / 総回答数、で定義する。偏り率のとり得る範囲は -1 から 1 であり、上位レベルの回答数が多い場合はプラス、下位レベルの回答数が多い場合はマイナスとなる。上位レベルと下位レベルの回答数が等しい場合、偏り率は 0 となる。

## 2.3 変数変換

### (1) 目的変数

工数は一般に対数正規分布に従うので対数変換する。本稿では直観的に理解しやすい常用対数で変換する。

### (2) 順序尺度に従う質的変数

順序尺度に従う変数の多くは 4 つのレベルの値を持つが、各レベル間が等間隔であるという保証はない。今回の分析では 3 レベル以上の値を持つ変数は隣り合うレベル同士を合併して全体で 2 レベルにする (2 値化する)。このときレベルの若番の方を上位レベル、老番の方を下

位レベルと呼ぶ。2レベル化する分割点は複数個あるが、すべての可能な分割点で分割したものを分析対象とする。

### (3) 名義尺度に従う質的変数

名義尺度に従う変数に対しては、着目する分類項目とそれ以外の分類項目すべて、の2つの分類にまとめる。例えば、「製造業」と「製造業以外のすべての業種」のように2値化する。

## 3 分析方法

### 3.1 分散分析と回帰分析

質的変数の影響を調べるための分析方法として次の3つの方法が考えられる。

**[方法1]** 質的変数の2つのレベルを2つの群とし、2つの群で工数に差があるかどうかを分散分析で調べる。

**[方法2]** 質的変数の2つのレベルを2つの群とし、2つの群で生産性(=工数/FP)に差があるかどうかを分散分析で調べる。

**[方法3]** 工数を目的変数とし、質的変数とFP規模を説明変数とする重回帰分析を行う。ただし、ここでは2つの群の誤差分散(回帰からの残差の分散)、回帰係数ともに等しいという仮定を設ける。

工数への影響要因の最大のものが規模であることはよく知られている。[方法1]は規模の影響を無視しているため、分析の精度に問題があると考えられる。例えば、質的変数のレベルに応じた2つの群のうち群1は相対的にFP規模が小さいプロジェクトが集まり、群2ではFP規模が大きなプロジェクトが集まっているとすると、正確な結果は期待できない。[方法2]は工数がFPに比例するという仮定が設けられている(情報量を減らしている)ことから[方法3]より劣ると考えられる。

以上のことから今回は[方法3]で分析することとする。質的変数は上位レベルと下位レベルの2値しかとらないため、質的変数の偏回帰係数の値はFP規模が等しいと仮定した場合の2つのレベルそれぞれの平均値の差に相当する。

### 3.2 説明変数の選択基準

回帰分析により、目的変数に影響を与える説明変数を選択する場合、説明変数の係数のP値が統計的に有意であるかどうかで判断することが多い。しかし、P値はデータ数Nに依存し、Nが大きくなれば真の傾きがゼロでない限り必ず有意となる。従ってP値の有意性だけで説明変数を選択すると実際には意味のないような係数の小さなものまで選択してしまう可能性が生じる。

それを避けるために係数の絶対値の大きさも考慮する必要があるが、単に係数の絶対値の大きさだけ見てい

てもわからないことがある。例えば、誤差分散が大きい場合は誤差変動も大きくなり、係数の絶対値が大きくてもその変数の総変動に寄与する割合は相対的に小さくなる。そのため、説明変数が目的変数に与える影響の度合いを示す指標を導入する必要がある。それが一般に効果量と呼ばれるものである。

効果量としては様々なものが定義されている。例えば回帰分析における寄与率(決定係数)もそのひとつである。しかし、プロジェクト計画時には規模はおおむね定まっているため、規模を除いた効果量の方が実際的であると考えられる。このことから本稿では母集団における平均値の差を標準偏差で正規化した次の量 $\delta$ を用いる[Cohen 1977]。

$$\delta = \frac{\mu_2 - \mu_1}{\sigma} \quad (1)$$

ただし、 $\mu_1$ は上位レベルの群の平均値、 $\mu_2$ は下位レベルの群の平均値である。式(1)の分子は質的変数の偏回帰係数から、分母は回帰からの誤差分散Veの平方根から推定する。

各選択条件の選択基準を次のように定め、すべての基準を満たすものを目的変数に対する影響要因とする。

#### (1) P値

係数がゼロでないかどうか(2つの群の平均値に差があるかどうか)を判断するための有意水準(P値)は統計学で標準的に使われている5%とする。

#### (2) 回帰係数

本稿では、回帰係数の値が $\log 1.5 = 0.176$ 以上のものを選択基準とする。これは2つのレベルそれぞれのプロジェクト群の工数に関する中央値に相当する値の比が1.5倍以上であることを意味する。なお、本稿では、上位レベルを0、下位レベルを1として分析しているため、係数がプラスの場合は、上位レベルの群の方が下位レベルのものよりも工数が少ないことを表す。

#### (3) 効果量 $\delta$

$\delta$ の値が0.5以上のものとする。これはCohenの提案した「大」「中」「小」の基準のうちの「中」程度の基準である[Cohen 1977]。

## 4 分析結果

質的変数(名義尺度及び順序尺度に従う変数)のうち、3.2で述べた影響要因の選択基準を満たした質的変数を表3(新規開発の場合)と表4(改修・保守の場合)に示す。表3と表4の工数比率と実現確率の欄の意味については考察で議論する。

表3と表4のすべての群について、FP規模と工数それぞれの歪度と尖度を求めたところ、新規開発(表3)で

はすべての群で±1の範囲に入っていて正規分布から逸脱している可能性は低い。改修・保守では、歪度が±1を超えているものが1件(407\_ドキュメント作成ツールの上位レベルの工数:1.03)あった。また、尖度が±1を超えているものが14件あったが、そのうち1.2

を超えているものは5件、絶対値が最大のものでも-1.5(1010\_テスト体制)である。この結果から、改修・保守でも正規分布からの大きな逸脱はないと考えられる。

以下の結果はすべてFP規模が同一と仮定した場合のものである。

表3 影響要因として選択された質的変数(新規開発)

分類	変数名	内容		データ数 N			質的変数		誤差分散 Ve	効果量 δ	工数比率	実現確率 (%)
		上位レベル	下位レベル	総数	上位レベル	下位レベル	係数 (*1)	P 値 (%)				
業種	201_業種1	製造業	左記以外	513	160	353	0.28	0.0	0.115	0.81	1.88	72
		金融・保険業	左記以外	513	135	378	-0.28	0.0	0.116	-0.82	1.91	72
プロジェクト全般	112_役割分担_責任所在	a:非常に明確	b:概ね明確+c:やや不明確+d:不明確	255	96	159	0.23	0.0	0.136	0.63	1.71	67
	113_達成目標_優先度_明確度合	a:非常に明確	b:概ね明確+c:やや不明確+d:不明確	232	81	151	0.27	0.0	0.130	0.76	1.88	70
	115_プロジェクト環境_騒音	a+b:(小さい)(*2)	c+d:(大きい)(*2)	211	179	32	-0.29	0.0	0.134	-0.79	1.94	71
	5241_品質保証体制_基本設計	a:プロジェクトメンバが実施	b:品質保証の専門スタッフが実施	213	150	63	0.25	0.0	0.130	0.68	1.76	69
ツール利用	404_プロジェクト管理ツール利用	a:有り	b:無し	201	128	73	-0.20	0.1	0.152	-0.51	1.58	64
	407_ドキュメント作成ツール利用	a:有り	b:無し	179	70	109	0.37	0.0	0.115	1.10	2.36	78
	408_デバッグ_テストツール利用	a:有り	b:無し	178	66	112	-0.23	0.0	0.146	-0.60	1.69	66
ユーザ側	509_ユーザ担当者_受け入れ試験関与	a:十分に関与+b:概ね関与	c:関与が不十分+d:全く関与していない	185	148	37	-0.28	0.0	0.141	-0.74	1.89	70
要求レベル	512_要求レベル_信頼性	a:極めて高い+b:高い	c:中位+d:低い	192	96	96	-0.22	0.0	0.146	-0.58	1.66	66
	518_要求レベル_セキュリティ	a:極めて高い+b:高い	c:中位+d:低い	178	80	98	-0.38	0.0	0.118	-1.11	2.41	78
	519_法的規制有無	a:業務レベルの規制あり+b:一般法レベルの規制あり	c:規制なし	159	78	81	-0.22	0.0	0.131	-0.62	1.67	67
開発担当者	602_要員スキル_業務経験	a:全員が十分な経験	b+c+d(左記以外)(*2)	244	59	185	-0.22	0.0	0.148	-0.57	1.65	66

(\*1) 2つの群の平均値の差に相当。 (\*2) 具体的なレベルの内容は付録を参照のこと。

表4 影響要因として選択された質的変数(改修・保守)

分類	変数名	内容		データ数 N			質的変数		誤差分散 Ve	効果量 δ	工数比率	実現確率 (%)
		上位レベル	下位レベル	総数	上位レベル	下位レベル	係数 (*1)	P 値 (%)				
アーキテクチャ	308_アーキテクチャ1	a:スタンドアロン	左記以外	217	33	184	0.46	0.0	0.144	1.21	2.89	80
		e:イントラネット/インターネット	左記以外	217	73	144	-0.22	0.0	0.161	-0.55	1.66	65
プロジェクト全般	5241_品質保証体制_基本設計	a:プロジェクトメンバが実施	b:品質保証の専門スタッフが実施	74	48	26	0.38	0.0	0.154	0.97	2.40	75
ツール利用	403_類似プロジェクト_有無	a:有り	b:無し	55	39	16	0.30	1.9	0.175	0.72	2.00	69
	404_プロジェクト管理ツール利用	a:有り	b:無し	111	54	57	-0.38	0.0	0.170	-0.91	2.38	74
	407_ドキュメント作成ツール利用	a:有り	b:無し	109	34	75	0.40	0.0	0.168	0.98	2.52	76
ユーザ側	501_要求仕様_明確さ	a:非常に明確+b:かなり明確	c:ややあいまい+d:非常にあいまい	107	76	31	0.25	0.8	0.185	0.58	1.78	66
要求レベル	512_要求レベル_信頼性	a:極めて高い+b:高い	c:中位+d:低い	98	54	44	-0.26	0.4	0.187	-0.60	1.82	66
	518_要求レベル_セキュリティ	a:極めて高い+b:高い	c:中位+d:低い	99	39	60	-0.36	0.0	0.178	-0.85	2.28	73
	519_法的規制有無	a:業務レベルの規制あり+b:一般法レベルの規制あり	c:規制なし	92	38	54	-0.39	0.0	0.175	-0.93	2.44	74
開発担当者	601_PMスキル	a:b:c:(レベル4以上)(*2)	d:レベル3(*2)	96	77	19	0.48	0.0	0.178	1.13	2.99	79
	1010_テスト体制	a:b:c:(スキル、員数のいずれかが十分)	d:スキル、員数ともに不足	100	85	15	0.40	0.2	0.187	0.92	2.50	74

(\*1) 2つの群の平均値の差に相当。 (\*2) ITスキル標準(バージョン1)の職種「プロジェクトマネジメント」に基づく

## 4.1 新規開発プロジェクト

- 業種別では、製造業がほかの業種より工数が少なく、金融・保険業は工数が多い。
- アーキテクチャでは差がみられない（スタンドアロンとメインフレームは偏り率が大きく分析対象外とした）。
- プロジェクト全般では、112\_ 役割分担\_ 責任所在及び 113\_ 達成目標\_ 優先度\_ 明確度合がそれぞれ非常に明確な場合は工数が少ない。115\_ プロジェクト環境\_ 騒音では騒音のある方が工数は少ない。5241\_ 品質保証体制\_ 基本設計では「プロジェクトメンバが実施」する方が「品質保証の専門スタッフが実施」するよりも工数が少ない。
- ツールを利用する方が利用しない場合よりも工数が少ないものは、407\_ ドキュメント作成ツールだけであり、404\_ プロジェクト管理ツール及び 408\_ デバッグ\_ テストツールでは利用している群の方の工数が多い。
- ユーザ側では、509\_ ユーザ担当者が受け入れ試験に関与している方の工数が多い。
- 要求レベルでは、512\_ 要求レベル\_ 信頼性や 518\_ 要求レベル\_ セキュリティが高い方が、また 519\_ 法的規制が有る方がいずれも工数が多い。
- 開発担当者に関しては、602\_ 要員スキル\_ 業務分野経験の高い方が工数が多い。
- 最も工数削減に有効と思われるものは、407\_ ドキュメント作成ツールの利用である。次いで 113\_ 達成目標\_ 優先度\_ 明確度合を非常に明確にすることである。
- 最も工数を増加させる要因は、518\_ 要求レベル\_ セキュリティが高いことである。

## 4.2 改修・保守プロジェクト

- 業種では差がみられない。
- アーキテクチャでは、スタンドアロンはほかのアーキテクチャより工数が少なく、インターネット・イントラネットはほかのアーキテクチャより工数が多い。
- プロジェクト全般では、5241\_ 品質保証体制\_ 基本設計では「プロジェクトメンバが実施」する方が「品質保証の専門スタッフが実施」するよりも工数が少ない。
- ツールの利用では、403\_ 類似プロジェクトがある方が、また 407\_ ドキュメント作成ツールを利用する方が工数は少ない。404\_ プロジェクト管理ツールを利用する方が利用しないよりも工数が多い。
- ユーザ側では、501\_ 要求仕様が明確である方があいまいであるよりも工数が少ない。

- 要求レベルでは、512\_ 要求レベル\_ 信頼性や 518\_ 要求レベル\_ セキュリティが高い方が、また 519\_ 法的規制がある方がいずれも工数が多い。
- 開発担当者では、601\_ PM スキルが高い群（IT スキル標準のレベル 4 以上）が低い群（レベル 3）よりも工数が少なく、また 1010\_ テスト体制で（スキル・員数共に不十分）であると工数が多い。ただし、いずれも下位レベルの内容が d レベルだけ、すなわち最も低いレベルだけである。
- 工数削減に最も有効なものは、開発担当者に関するものを除くと 407\_ ドキュメント作成ツールの利用である。
- 工数を増加させる主な要因は、5241\_ 品質保証体制\_ 基本設計を品質保証の専門スタッフが実施すること、404\_ プロジェクト管理ツールの利用、518\_ 要求レベル\_ セキュリティ、519\_ 法的規制有無である。

## 5 考察

### 5.1 新規開発と改修・保守における影響要因の違い

新規開発と改修・保守の影響要因を比較した結果を表 5 に示す。特徴的なことを次に示す。

- 新規開発では業種によって差がみられ、改修・保守ではアーキテクチャによって差がみられる。
- プロジェクト全般では、112\_ 役割分担\_ 責任所在を明確にすること、113\_ 達成目標\_ 優先度を明確にすることは新規開発では効果があることが伺える。5241\_ 品質保証体制\_ 基本設計では、新規開発、改修・保守いずれも品質保証の専門スタッフが実施すると工数が多くなるが、品質向上の観点からはやむを得ないと考えられる。
- ツールの利用では、404\_ プロジェクト管理ツール利用はいずれも工数増に、407\_ ドキュメント作成ツール利用はいずれも工数減に働いている。403\_ 類似プロジェクトの存在は改修・保守では有効である。
- ユーザ側では、新規開発、改修・保守とも影響要因は 1 件ずつと少ない。
- 要求レベルでは、新規開発、改修・保守とも同じ傾向を示している。
- 開発担当者では、新規開発では 602\_ 開発担当者\_ 業務分野経験のレベルが高い方の工数が多く、改修・保守では 601\_ PM スキルが高い方の工数が少なく、1010\_ テスト体制のスキルレベル・員数共に不足している群の工数が多い。

### 5.2 経験則に合う結果と合わない結果

表 3 と表 4 に示された分析結果は、工数増に作用する

表5 新規開発と改修・保守における影響要因の違い

分類	変数	内容		効果量 $\delta$	
		上位レベル	下位レベル	新規開発	改修・保守
業種	201_業種1	製造業	左記以外	0.81	-
		金融・保険業	左記以外	-0.82	-
アーキテクチャ	308_アーキテクチャ1	a: スタンドアロン	左記以外	-	1.21
		e: イントラネット/インターネット	左記以外	-	-0.55
プロジェクト全般	112_役割分担_責任所在	a: 非常に明確	b: 概ね明確 + c: やや不明確 + d: 不明確	0.63	-
	113_達成目標_優先度_明確度合	a: 非常に明確	b: 概ね明確 + c: やや不明確 + d: 不明確	0.76	-
	115_プロジェクト環境_騒音	a+b: (小さい) (*1)	c+d: (大きい) (*1)	-0.79	-
ツール利用	5241_品質保証体制_基本設計	a: プロジェクトメンバが実施	b: 品質保証の専門スタッフが実施	0.68	0.97
	403_類似プロジェクト_有無	a: 有り	b: 無し	-	0.72
	404_プロジェクト管理ツール利用	a: 有り	b: 無し	-0.51	-0.91
	407_ドキュメント作成ツール利用	a: 有り	b: 無し	1.10	0.98
ユーザ側	408_デバッグ_テストツール利用	a: 有り	b: 無し	-0.60	-
	501_要求仕様_明確さ	a: 非常に明確 + b: かなり明確	c: ややあいまい + d: 非常にあいまい	-	0.58
要求レベル	509_ユーザ担当者_受け入れ試験関与	a: 十分に関与 + b: 概ね関与	c: 関与が不十分 + d: 全く関与していない	-0.74	-
	512_要求レベル_信頼性	a: 極めて高い + b: 高い	c: 中位 + d: 低い	-0.58	-0.60
	518_要求レベル_セキュリティ	a: 極めて高い + b: 高い	c: 中位 + d: 低い	-1.11	-0.85
開発担当者	519_法的規制有無	a: 業務レベルの規制あり + b: 一般法レベルの規制あり	c: 規制なし	-0.62	-0.93
	601_PMスキル	a: + b: + c: (レベル4以上) (*2)	d: レベル3 (*2)	-	1.13
	602_要員スキル_業務経験	a: 全員が十分な経験	b + c + d (左記以外) (*1)	-0.57	-
	1010_テスト体制	a: + b: + c: (スキル、員数のいずれかが十分)	d: スキル、員数ともに不足	-	0.92

(\*1) 具体的なレベルの内容は付録を参照のこと。 (\*2) ITスキル標準 (バージョン1) の職種「プロジェクトマネジメント」に基づく。

か工数減に作用するかという観点からみて、①納得のいくもの、②意外ではあるが結果を受け入れることができるもの、③筆者の経験則と合わないものに分けることができる。

新規開発の115\_プロジェクト環境\_騒音、408\_デバッグ\_テストツール利用、509\_ユーザ担当者\_受け入れ試験関与が②に属し、新規開発の404\_プロジェクト管理ツール利用、602\_要員スキル\_業務経験、改修・保守の、404\_プロジェクト管理ツール利用が③に属す。それ以外はすべて①に属すと考えられる。

②に属するものについては次のような解釈ができる。

- 115\_プロジェクト環境\_騒音：騒音が大きいと工数が少なくなる（生産性が高くなる）のではなく、コミュニケーションが活発に行われるなどチームが活動的な方が生産性は高くなると解釈することができる。
- 408\_デバッグ\_テストツール利用：テスト・デバッグツールの利用は、工数という点からはマイナスであるという結果が得られたが、信頼性の向上に役立っている可能性がある。
- 509\_ユーザ担当者\_受け入れ試験関与：ユーザ担当者が受け入れ試験に関与することは工数にマイナスであるが、信頼性向上には寄与している可能性が高い。

③に属するものについては、なぜ工数増につながるのかは明確ではなく、今後詳細に分析していく必要がある。例えば、404\_プロジェクト管理ツールの利用が工数を

増加させるのではなく、工数の大きいプロジェクトでプロジェクト管理ツールを利用してということも考えられるが、規模で補正しているため、この解釈だけでは説明できない。

### 5.3 結果の解釈

#### (1) 工数比率

今回の分析方法で求めた係数の値は、上位レベルと下位レベルそれぞれの群の工数の平均値の差に相当する。工数の値は対数変換されているので、平均値の差は元のスケールでは2つの群それぞれの中央値の比の値となる。表3と表4の「工数比率」はその値を示している。ただし、比較しやすいようにすべて工数の多い方の群を少ない方の群で割っている。例えば、表3の407\_ドキュメント作成ツール利用では、利用する場合としない場合で工数比率が2.36倍違う。これは利用しないプロジェクト群の工数の中央値が利用したプロジェクト群の工数の中央値の2.36倍であることを示している。このように中央値の比率（対数変換後の平均値の差）からみると、その差は非常に大きくみえる。

#### (2) 実現確率

ここで効果量 $\delta$ の意味を考えてみる。 $\delta$ の値は平均値の差が標準偏差の何倍であることを示すものである。ただし、今回は簡単のために母集団のふたつの群の標準偏差は等しいという仮定を設けている。 $\delta$ が小さいということは、平均値の差に比べて各群の分散が大きいということ、すなわち対象とする変数による影響に比べて他の要因の影響が大きいことを示している。標準偏差が

大きいとふたつの群の工数の分布は広く重なり合う。重なり合う面積は  $\delta$  が 1 で 31%、2 で 16%、3 で 17%、4 で 2.3% である。

ここで上位レベル群と下位レベル群からそれぞれランダムにひとつずつ同一 FP 規模のプロジェクト a と b を選び出した場合に、a の工数が b の工数よりも大きくなる確率を考える。ただし、ここでは a は工数が多い群、b は少ない群から選ぶものとする。これを本稿では「実現確率」と名付けてその値を表 3 と表 4 に示している。 $\delta$  が大きければ非常に高い確率で  $a > b$  となるが、 $\delta$  が小さい場合はその確率は低くなる。この計算は、平均が  $\mu_1$  と  $\mu_2$ 、分散がいずれも  $\sigma^2$  と仮定した 2 つの正規分布からそれぞれ選んだ 2 つの確率変数 X と Y の差の分布が、平均  $\mu_1 - \mu_2$ 、分散  $2\sigma^2$  の正規分布に従うということから計算できる。

例えば、新規開発の 407\_ドキュメント作成ツールの利用の実現確率は 78% すなわちほぼ 9 回に 2 回 ( $\approx 0.22 = (1 - 0.78)$ ) はほかの要因の影響によってツールを利用しないプロジェクトの方の工数が少ないことがある、ということになる。このように、それぞれの群の平均値の差だけでなく、標準偏差 (分散) も考慮することによって実際の影響度をより深く理解することができるようになると考えられる。

参考のために、407\_ドキュメント作成ツールの利用の有無別の FP 規模と工数の散布図と回帰直線を図 1 に示す。グラフに示された 2 本の回帰直線は次の重回帰式

$$Y = 1.09X_1 + 0.37X_2 + 0.38 \quad (2)$$

において  $X_2 = 0$  (利用あり) の場合と  $X_2 = 1$  (利用なし) の場合のものである。ただし、 $X_1$  は log FP、 $X_2$  は 407\_ドキュメント作成ツールの利用の有無、Y は log 工数 (人時) である。図 1 の目盛は常用対数で表示しているので、例えば log FP が 2.0 の目盛は元の FP 値では 100FP であることを表す。明らかに利用の有無で工数に差がみられ

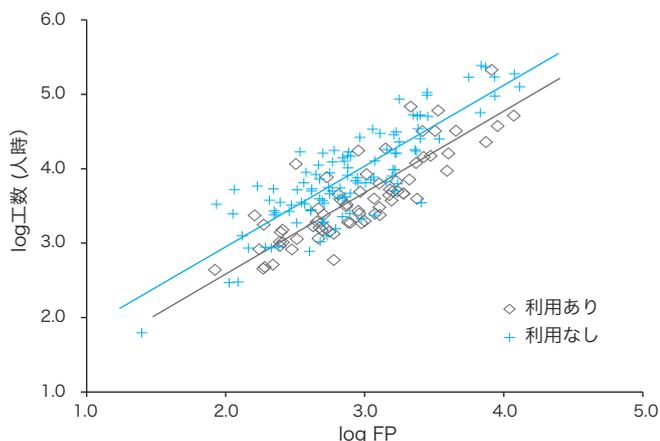


図 1 ドキュメント作成ツール利用の散布図と回帰直線

るが、重なり合う部分も少なくない。

表 3 と表 4 からわかるように、新規開発、改修・保守共に、実現確率は 65 ~ 80% の間にある。影響要因として選定した 2 つの群からそれぞれランダムにひとつずつ同一 FP 規模のプロジェクトを選び出した場合、工数が少ないはずの群からのプロジェクトでも 5 回に 1 回、悪くすると 3 回に 1 回はほかのプロジェクトよりも工数が多くなる。

## 5.4 業種別分析

表 3 に示したように、業種によって同一規模でも開発工数が異なることから業種で層別して同様の条件で分析を試みた。新規開発の製造業以外はデータ件数が不足して結果を得ることができなかった。新規開発の製造業の分析結果を表 6 に示す。

表 6 からわかる特徴的な事項を次に示す。

- 120\_計画の評価 (コスト) が工数削減に有効である。なお、121\_計画の評価 (品質) と 122\_計画の評価 (工期) も同様の結果であったが、これら 3 つの変数には高い相関があることから意味的に最も妥当な 120\_計画の評価 (コスト) のみを表 6 に示した。
- 406\_設計支援ツールの利用や 411\_コードジェネレータの利用が工数削減に有効である。
- 112\_役割分担\_責任所在、113\_達成目標\_優先度\_明確度合、407\_ドキュメント作成ツール利用は、表 3 で示した結果と同様にいずれも工数削減に有効である。いずれも係数、効果量共に表 3 に示した層別前の結果よりも大きくなり、実現確率は 112\_役割分担\_責任所在では 84%、407\_ドキュメント作成ツールでは 91% になった。

## 6 まとめ

質的要因が工数に与える影響度を調べた。工数に対して最も大きな影響を与える規模の影響を取り除くために、各質的要因と FP 規模を 2 つの説明変数とする重回帰分析を行い、係数の P 値と大きさ、効果量  $\delta$  の 3 つの選択基準のもとで、工数に大きな影響を与える変数を選択した。

今回の分析は各質的要因単独に対して行ったので、質的要因間の相関は無視したものとなっている。そのため複数の変数を組み合わせて重回帰分析を行った場合は異なった結果になっているものと思われる。とはいえ、変数単独の効果量でもプロセス改善やプロジェクト計画時の見積もりに対するひとつの参考情報になりうる。

今回の分析結果で次のことが明らかになった。

- 新規開発、改修・開発のいずれの開発種別でも、工数削減 (生産性向上) に大きく寄与するのは、ドキュ

表6 影響要因として選択された質的変数（新規開発：製造業）

分類	変数名	内容		N			質的変数		誤差分散 Ve	効果量 $\delta$	工数比率	実現確率 (%)
		上位レベル	下位レベル	総数	上位レベル	下位レベル	係数 (*1)	P 値 (%)				
アーキテクチャ	308_アーキテクチャ1	c:2階層クライアント/サーバ	左記以外	159	24	135	0.18	1.4	0.108	0.55	1.52	65
プロジェクト全般	112_役割分担_責任所在	a:非常に明確	b:概ね明確+ c:やや不明確+ d:不明確	89	39	50	0.42	0.0	0.090	1.40	2.62	84
	113_達成目標_優先度_明確度合	a:非常に明確	b:概ね明確+ c:やや不明確+ d:不明確	70	33	37	0.34	0.0	0.097	1.09	2.18	78
	120_計画の評価(コスト)	a:コスト算定の根拠が明確で実行可能性を検討済み	b:コスト算定の根拠が不明確、又は実行可能性を未検討+ c:計画なし	108	74	34	0.24	0.1	0.107	0.72	1.73	70
ツール利用	406_設計支援ツール利用	a:有り	b:無し	52	39	13	0.28	0.4	0.085	0.97	1.91	75
	407_ドキュメント作成ツール利用	a:有り	b:無し	54	32	22	0.44	0.0	0.056	1.86	2.75	91
	411_コードジェネレータ利用	a:有り	b:無し	55	23	32	0.30	0.1	0.083	1.04	2.00	77
ユーザ側	502_ユーザ担当者_要求仕様関与	a:十分に関与	b:概ね関与+ c:関与が不十分+ d:未関与	62	23	39	0.28	0.2	0.104	0.87	1.90	73

(\*1) 2つの群の平均値の差に相当

メント作成ツールの利用である。

- いずれの開発種別でも、要求レベルの高さが工数の増加（生産性の低下）を招く。
- 品質保証体制\_基本設計をプロジェクトで行わずに品質保証部門に委ねると工数が増加する。ただし、品質向上の観点からはやむを得ないと考えられる。
- 新規開発で工数削減に効果のあるものは、113\_達成目標\_優先度\_明確度合と112\_役割分担\_責任所在である。
- 新規開発で工数の増加に影響を与えるものには、デバッグ\_テストツールの利用、ユーザ担当者\_受け入れ試験関与があるが、品質向上の観点からは必要なものであると考えられる。
- 改修・保守で工数削減に効果のあるものは、類似プロジェクトの利用、要求仕様の明確化、PM スキルレベルが低くないこと、テスト体制においてスキルレベル及び員数のいずれも不十分でないことがあげられる。
- 製造業の新規開発で層別した結果は、新規開発全体で分析した結果と矛盾はなく、新たに120\_計画の評価(コスト)が十分な場合、406\_設計支援ツールを利用する場合、411\_コードジェネレータを利用する場合に工数削減が期待できる。

今後は、経験則と一致しない結果の詳細分析や交互作用の影響なども考慮した、より精度の高い分析を進めていく予定である。

## 謝辞

本研究は東海大学とIPA/SECが共同で実施したものである。松本所長、山下リーダー並びにSEC研究員の方々のご協力に深く感謝致します。

## 付録

変数の各レベルの内容

### (1) 115\_プロジェクト環境\_騒音

- a: 騒音による影響は全くなく、電話による作業中断も最低限
- b: 騒音はほとんど気にならない。電話による作業中断は時々ある
- c: 時としてかなりの騒音があり、電話も作業が時々中断する
- d: 騒音がひどく、必要な集中力が維持できない。電話による作業中断も一時間毎以上の頻度である

### (2) 602\_要員スキル\_業務経験

- a: 全員が十分な経験
- b: 半数が十分な経験、残り半数はいくらかの経験
- c: 半数がいくらかの経験、残り半数は経験なし
- d: 全員が経験なし

### 【参考文献】

- [Jørgensen 2007] Jørgensen, M, and Shepperd, M.: A Systematic Review of Software Development Cost Estimation Studies, IEEE Trans. Software Engineering, Vol.33, No.1, pp.33-53 (2007) .
- [Boehm 1981] Boehm, B. W.: Software Engineering Economics, p. 767, Prentice-Hall, Inc. (1981) .
- [IPA 2014] 独立行政法人情報処理推進機構 (IPA) 技術本部 ソフトウェア信頼化センター (SEC) 監修: ソフトウェア開発データ白書 2014-2015, 2014年.
- [Furuyama 2014] Furuyama, T.: Analysis of Factors that Affect Productivity of Enterprise Software Projects, IT Confidence Conference 2014 (2014) .
- [Cohen 1977] Cohen, J.: Statistical Power Analysis for the Behavioral Sciences, Rev. ed., p.474, Academic Press (1977) .

## 情報システムの事故データ

# 情報システムの障害状況 2015年後半データ

IPA 顧問

松田 晃一

SEC システムグループ 主任

八嶋 俊介

2015年7月から12月までに報道された情報システムの障害状況を報告する。この間に報道された情報システムの障害は合計24件で月平均4.0件となった。平均的な値に対しやや多い値である。とくに今期はマイナンバー制度の開始に伴いその関係の障害が多く発生している。また、長期間認識されずに運用されてきた不具合が当期になって発覚した事故が発生している。

## 1. はじめに

本稿では、2015年7月から12月までの2015年後半の半年間に報道された情報システムの障害状況をとりまとめて報告する。まず、次章で今期の概況について述べ、続く3章では、今期の特徴的な事例であるマイナンバー関連の事故についてとりまとめて報告し、4章において長期間認識されずに運用されてきたエラーが発覚した事故例について、5章では設計時の常識的事項の考慮漏れによる障害事例について述べる。

## 2. 2015年後半の概況

2015年7月から12月までの半年間で報道された情報システムの障害は合計24件となった(表1)。後半の障害発生件数を月平均にすると4.0件となるが、通年では合計45件、月平均3.75件となる(図1)。

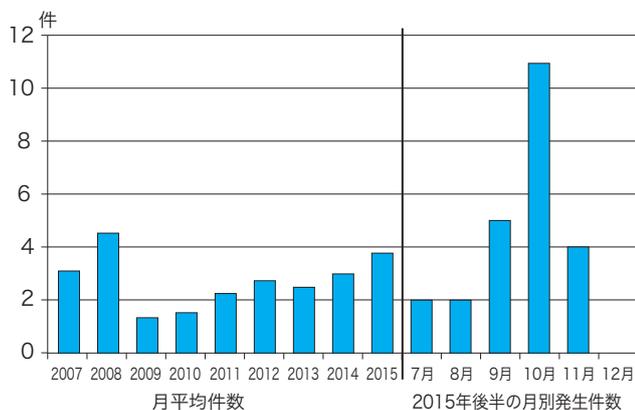


図1 情報システムの障害発生件数の推移

今期はマイナンバー関連の事故の発生が数多く報道されていることが特徴的である。今期から本格的な運用が開始され今後重要な社会インフラとして運用されるシステムであるため、関連の事故の概要について次章で取り上げる。

今期の事例の中には、以前から内在していた不具合が見逃されたまま長期間運用されてきたものが偶然発見された事故が5件発生(事例1524、1525、1527、1530、1541)している。この種の事故は今年前半にも4件発生しており、以前にも同種の問題を取り上げた[松田2015]が、あらためて4章で詳細を示す。

## 3. マイナンバー関連事故

2015年7月から12月に発生した、マイナンバー関連の障害は合計8件であるが、その原因別に3件にまとめた。概要は事例1533、1534、1535に示す通りである。その発生日時を見ると、すべての事例が昨年10月に発生しているが、この月はマイナンバーの通知カードが配布され始め、いよいよ本格的に制度が動き始めた月でもある。開始わずか1カ月で、多くの障害が発生した。

事例1533は、住民票を取得する際、マイナンバーの記載を希望していないにもかかわらず、システムの設定ミスにより誤ってマイナンバーが記載されてしまった事例である。影響は、茨城県取手市では、市内の2カ所にあった自動交付機から発行された、合計69世帯100人に及び、同県美浦村では28人に対し、希望していないにもかかわらずマイナンバーが記載された住民票が発行された。

事例1534も、住民票に誤ってマイナンバーが記載されてしまった例であるが、こちらはシステムの設定ミスでは

なく、ヒューマンエラーによるものである。多くの自治体で、「住民票コード」の記載を希望した住民票に対して、窓口の職員が誤って「個人番号（マイナンバー）」を記載した住民票を発行してしまった。今後、このような記載項目の誤りが発生しないよう、職員への周知及び、システムのチェックを強化する対策が検討されているとのことである。

以上のように、申請者が希望していないにもかかわらず、住民票にマイナンバーが誤って記載されるという障害が目立った。マイナンバーは、基本的には生涯不変の番号として発行されるが、例外として、第三者に番号が漏れたことが確認されたときは、本人の希望により番号の再発行が可能となっている。今回紹介した事例のように、住民票に本人が意図しないマイナンバーが記載され、それに気づかず第三者に渡してしまった場合も、番号の漏洩として扱われる。

今回の取りまとめでは、番号が配布され始めて1カ月を待たず、早速マイナンバーの再発行に至ったケースが数多く見受けられた。システムの設定を誤りなく実施することはもちろん、極めて重要な個人情報を取り扱われることを考慮し、利用者の立場に立ち、操作ミスがないようユーザビリティを向上させることも、システム開発者に課せられたミッションであろう。

## 4. 長期間の不具合放置

事例 1530 は、奨学金の返還を管理するシステムにおいて、返還が滞りなく行われているにもかかわらず、誤った情報を個人信用情報機関に登録していたことが、対象者からの問合せを受けて発覚した事例である。個人信用情報機関に対し情報を提供する連携システムに不具合があったことが原因である。少なくとも5年以上前からこの不具合を抱えたまま運用されており、この間632人の信用情報照会があり、内26人については誤った信用情報を回答し、ローン契約などに影響があった可能性があると報じられている。与信に疑問を持った人からの問合せを受け、点検した結果不具合を発見したとのことである。このような問合せが無ければ、引き続き誤った信用情報の登録が続き、誤った情報に基づいた与信により本人も気が付かない内に不利益を被っていた可能性があり、その影響は深刻である。

事例 1541 は緊急速報メールを送信するシステムが、導入時からメールを送れない設定のまま運用されていたところ、約1年半後の防災訓練において初めて不備が発覚した事例である。テストをしていればミスは容易に発見できたものであるが、それが行われていなかった。緊急時に重要

表1 2015年後半の情報システム障害データ（報道に基づきSECが整理）

No.	システム名	発生日時（上段） 回復日時（下段）				影響	現象と原因	直接原因	情報源
		年	月	日	時				
1522	au キャリアメール	2015	7	12	18時26分	全国の約796万回線でメールの送受信ができなくなった。送受信ができない状態は13日午後1時に解消したが、一部のユーザで送受信が遅延したり、一部の過去メールの閲覧ができない状態が続いていた。	通信監視施設の一室で機器が燃える小規模な火災が起き、火災報知機が作動、空調が停止した。この影響で室温が上昇し、メールサーバが自動停止したためと見られる。火災の原因は明らかにされていない。	火災によるハード障害	<ul style="list-style-type: none"> <li>・朝日新聞電子版（2015.7.13）</li> <li>・日本経済新聞夕刊（2015.7.13）</li> <li>・au プレスリリース（2015.7.13）</li> <li>・au プレスリリース（2015.8.3）</li> </ul>
		2015	8	3					
1523	外為どっとコム 外貨ネクストネオ	2015	7	13	13時20分	システム障害のため、全取引が一時停止した。同社の口座数は5月末で39万3584口座、預かり資産残高は1,096億7,100万円。  ※取引停止中にはユーロ圏首脳会議がギリシャ支援の再開に条件付きで合意し、為替相場が大きく変動していたが、同社の利用者は取引することができなかった。	注文処理等を実行するサーバ全5台のうち1台において、ネットワーク機器との通信経路上の一部ハードウェアに不良が発生した。本来は待機系に自動的に切り替わるはずが、ファームウェアの不具合により切り替えが正常に動作しなかった。また、当該サーバ1台をシステム構成から切り離したものの、未処理チェック機能が想定以上のデータ量を処理しようとし、結果的に処理能力を超えたため、長時間のサービス停止となった。	ハードウェア障害	<ul style="list-style-type: none"> <li>・NHK ニュース（2015.7.13）</li> <li>・日経コンピュータ電子版（2015.7.13）</li> <li>・朝日新聞朝刊（2015.7.14）</li> <li>・日本経済新聞朝刊（2015.7.14）</li> <li>・外為どっとコムシステムレポート（2015.8.13）</li> </ul>
		2015	7	13	21時50分				
1524	佐賀市納税サービス	2015	5			佐賀市の担当者が、ベンダの提案資料を誤解し、別契約が必要だったモバイルレジがそのまま利用できると勘違いした。クレジットカードの利用については事前にスマートフォンからの納税サービス（モバイルレジ）が利用できなかった。	要件定義ミステスト漏れ	<ul style="list-style-type: none"> <li>・日経コンピュータ（2015.8.6号）</li> </ul> ※障害発生日時は2015年5月であるが、影響が判明した日時に基づき掲載。	

No.	システム名	発生日時 (上段) 回復日時 (下段)				影響	現象と原因	直接原因	情報源
		年	月	日	時				
1525	スパコン 網HPCI	2014	8			スパコンを共同利用するために結んだネットワークのデータ保存拠点(東工大)にて障害があり、利用者の保存データ約15万件が破損、うち約1,000件が消失した。	東工大のデータ保存拠点で2014年8月～10月、データの書き込みを行う部分に障害が起き、保存データ約15万件が破損した。うち1,072件はコピーも破損し、復旧ができなかった。障害はその後システムを更新する過程で修復されたとみられ、2015年3月に利用者が指摘するまでトラブルに気がつかなかった。2014年7月に落雷による停電があり、その後の復旧作業で障害が起きた可能性がある。再発防止策として、同様の障害が起きたとしても早期に発見できるようにシステムを整備した。	落雷による停電	<ul style="list-style-type: none"> <li>日本経済新聞朝刊 (2015.8.24)</li> </ul> ※障害発生日時は2014年8月であるが、影響が判明した日時に基づき掲載。
1526	NTT データ CAFIS システム	2015	9	5	16時06分	全国の小売店や飲食店などの店舗と、クレジットカード会社、金融機関を結ぶシステムに障害が起き、クレジットカードの決済が遅れたり、できなくなったりした。このシステム利用者は全国の小売店など約2,000社、国内のクレジットカード会社約120社、金融機関約1,600拠点。	障害の直接原因は、FEP(フロントエンドプロセッサ)サーバーと呼ばれる、大量の処理を振り分けるサーバーがダウンしたことによるもの。ハードウェアが不安定になり、それをコントロールするミドルウェアも制御できず、ダウンした。FEPサーバーは複数台あるが、そのうちの1台がダウンした。SEが状況を把握、判断した上で待機系のサーバーに切り替えて復旧させた。サーバーが不安定になった原因や、ミドルウェアが制御できなかった理由、切り替えに時間がかかった経緯などは不明。	ハードウェア障害	<ul style="list-style-type: none"> <li>NTT データプレスリリース (2015.9.5,6)</li> <li>日本経済新聞朝刊 (2015.9.6,7)</li> <li>朝日新聞朝刊 (2015.9.6,7)</li> <li>日経コンピュータ電子版 (2015.9.5,6)</li> <li>日経コンピュータ (2015.10.15)</li> </ul>
1527	厚生 労働省 メタボ 健診シ ステム	2009	4			メタボ健診で保健指導を受けた人とそうでない人の間で医療費に差が出るかを調べるため、2009年4月から導入されていたが、入力データの8割は活用されていなかった。	原因は、医療機関が入力する書式の不一致。たとえば、健診データが全角、レセプトが半角だった場合、システムで暗号化されたIDが異なると同一人物と認定されず、データを突き合わせられなかった。システム設計段階では模擬データを使った検証しかしていなかった。12年には突き合わせられるデータが少ないことを把握していたが、原因調査を行っていなかった。今後、システム改修を予定している。	設計ミス	<ul style="list-style-type: none"> <li>朝日新聞朝刊 (2015.9.5)</li> <li>日本経済新聞朝刊 (2015.9.20)</li> </ul> ※障害発生日時は2009年4月であるが、影響が判明した日時に基づき掲載。
1528	住信 SBI ネット 銀行	2015	9	7	22時11分	web サイトへのアクセス、ログイン、各種取引、ATMの利用ができなくなった。	原因調査中。	不明	<ul style="list-style-type: none"> <li>住信 SBI ネット銀行プレスリリース (2015.9.8)</li> <li>日本経済新聞朝刊 (2015.9.8)</li> </ul>
1529	気象庁 地震観測 システム	2015	9	16		東海地震や東南海地震の観測システムに障害が発生し、静岡県沖に設置した地震計など13観測点のデータが受け取れなくなった。周辺で地震が起きた場合、緊急地震速報の発表が最大14秒遅れる可能性があった。	電源の動作不良が起きていたことが原因。	ハードウェア障害	<ul style="list-style-type: none"> <li>日本経済新聞朝刊 (2015.9.17,18)</li> </ul>
1530	日本学生 支援機構 個人信用 (個人) システム	2010	4			規定の金額を返還していたにもかかわらず、返還額が規定に満たないとし、誤った信用信息を個人信用信息機関に通知していた。対象となったのは632人で、実際に金融機関等から照会があり、ローンの契約申し込みなどに影響が生じた可能性があるのは26人。	半年賦分の入金が毎月必要とみなし、月賦分のみで良い月も一部しか入金がないという誤った情報を登録した。原因は個人信用信息機関との連携システムの不具合(プログラムミス)。システムが稼働してから5年以上、不具合は発覚しなかったが、返還者からの問い合わせを受け、システムを点検してこの不具合を発見した。このような不具合は、納品時の受け入れテストで発見されることが多いが、支払い方法が多岐にわたっており、レアケースを中心にテストを行ったため、基本的な部分のテスト項目が漏れてしまった。	ソフトウェア障害	<ul style="list-style-type: none"> <li>朝日新聞朝刊 (2015.9.18)</li> <li>日本経済新聞朝刊 (2015.9.18)</li> <li>日本学生支援機構プレスリリース (2015.9.17)</li> <li>日経コンピュータ (2015.10.29)</li> </ul> ※障害発生日時は2010年4月であるが、影響が判明した日時に基づき掲載。

No.	システム名	発生日時 (上段) 回復日時 (下段)				影響	現象と原因	直接原因	情報源
		年	月	日	時				
1531	奈良県 土砂災害・ 防災情報 システム	2013	8			対応している web ブラウザが IE のみで、さらに IE7 以降のバージョンでは、互換表示機能を使わないと表示ができなかった。	県は不具合を解消する必要性を認識していたものの、検討のスピードが遅かった。(IE と互換表示機能の利用を呼び掛けるに留まっていた) 2015 年度予算にシステム改修費を計上したものの、契約方法の検討に手間取ったため、4月になっても着手ができなかった。(6月に新聞にて問題を報道されるまで着手しなかった)	保守の 対応 遅れ	・日経コンピュータ (2015.10.1)  ※障害発生日時は 2013 年 8 月であるが、影響が判明した日時に基づき掲載。
		2015	9	30					
1532	奈良県 土砂災害・ 防災情報 システム	2015	7	16		当該期間の約 1 カ月、システムを利用できない状態が続いた。	処理が遅延し、4～5 時間前の情報が表示される状態に陥ったため、システムの稼働を止めた。 原因はハード故障。冷却ファンとハードディスクを交換すると、正常に稼働するようになった。ハード故障は経年劣化によるもの。	ハード ウェア 障害	・日経コンピュータ (2015.10.1)
		2015	8	21					
1533	住民票 システム	2015	10			個人番号 (マイナンバー) が記載された住民票を誤って交付した。 ・茨城県取手市…69 世帯 100 人 (10 月 5 日から 9 日に市内 2 箇所にある自動交付機から発行) ・茨城県美浦村 (10 月 5 日) …28 人	個人番号 (マイナンバー) を誤って記載した住民票を交付した。 原因は、ベンダーが自動交付機発行システムの設定を誤ったため。	システム 設定 の誤り	・取手市ホームページ (2015.10.113) ・日経コンピュータ電子版 (2015.10.15) ・美浦村ホームページ (2015.10.13)
1534	住民票 システム	2015	10			住民から求められていないのに、誤って個人番号 (マイナンバー) を記載した住民票を交付した。 ・札幌市厚別区 (10 月 6 日) …2 人 ・福島市 (10 月 7 日) …1 人 ・石川県白山市 (10 月 5 日) …3 人 ・目黒区…1 人 ・岐阜県恵那市…1 人	原因は、窓口の職員の誤り。住民票コードを記載した住民票の申請に対して、個人番号 (マイナンバー) を記載した住民票を交付したケースが多い。 住民の意思を確認したうえで個人番号 (マイナンバー) を変更する対応を行った。	窓口の 操作 ミス	・NHK ニュース (2015.10.15) ・読売新聞電子版 (2015.10.22) ・読売新聞電子版 (2015.10.21) ・朝日新聞電子版 (2015.10.24) ・岐阜新聞電子版 (2015.10.24)
1535	住民票 システム	2015	10	14		個人番号 (マイナンバー) を記載した住民票を申請したが、番号欄が空白の住民票が発行された。 ・船橋市…32 人	原因は、印鑑登録関係のデータと証明書自動交付機のデータ連携がうまくいかなかったため。他市担当の SE からの連絡でチェックしたところ、検出された。	データ 連携 ミス	・日経コンピュータ電子版 (2015.10.16)
		2015	10	14					
1536	中部国際 空港 空港内 システム	2015	10	6	9 時 25 分	発着便を案内する表示盤の管理システム、国際線のチェックインシステム、手荷物取扱いシステム、店舗 POS レジのシステム、駐車場システム、空港警備システムが停止した。 国際線の出発便 10 便に最大で 1 時間の遅れが発生したほか、搭乗手続きや手荷物の搬送ができなかった。 店舗と駐車場では、クレジットカードのオンライン処理ができなかった。影響を受けたのは 1,500 人以上。	原因は、空港内のシステムが接続するネットワーク機器の障害。 8:53 にネットワーク機器を設置するサーバールームの空調機器が故障により停止したことで、機器に異常が発生し、ネットワーク障害が発生した。空港は 9:19 に空調機器を復旧し、システムを順次復旧させた。 管制システムとは別系統で、航空機離着陸の安全には影響がなかった。	ハード ウェア 障害	・ITpro (2015.10.6) ・読売新聞朝刊 (2015.10.7)
		2015	10	6	12 時 43 分				
1537	ピーチ 航空 予約シ ステム	2015	10	8	1 時 30 分	約 900 人の顧客について、クレジットカードや独自のポイントによる決済が処理されたにもかかわらず、予約が成立していなかった。返金措置を取ったが、顧客は再度予約を取り直す必要があった。	原因は調査中。	不明	・ピーチ航空プレスリリース (2015.10.8) ・ITpro (2015.10.9) ・産経新聞夕刊 (大阪) (2015.10.9)
		2015	10	8	5 時 30 分				

No.	システム名	発生日時 (上段) 回復日時 (下段)				影響	現象と原因	直接原因	情報源
		年	月	日	時				
1538	コンビニレジ 銀行 ATM 端末	2015	10	8	夕方	ローソンやファミリーマートなど一部コンビニのレジや銀行の ATM 端末で、入金ができなくなった。影響人数は数千人。	原因は調査中。	不明	・朝日新聞朝刊 (2015.10.9)
		2015	10	8	夜				
1539	中部国際 空港 空港内 システム (2回目)	2015	10	30	5時20分	ネットワーク障害が起き、オンラインでの国際線のチェックインなどが一時できなくなった。フライト情報システムの不具合で発着便を示す電光掲示板が更新できなくなり、手荷物取扱い、店舗レジ、駐車場のシステムにも影響が出た。運航への影響はない。	今月上旬にも同様のトラブルがあった。原因は調査中。	不明	・朝日新聞電子版 (2015.10.30) ・中日新聞夕刊 (2015.10.30)
		2015	10	30	7時20分				
1540	NTT 東日本 ひかり 電話	2015	10	8		ひかり電話ルータ/ホームゲートウェイに不具合があり、かけた電話番号と異なる番号に着信するケースがあった。(10回に3回程度) プレスリリース時点での顧客からの申告は16件。	不具合は、当該機種に接続している電話機の設定が「ダイヤルパルス 20pps」の場合に発生する。パルス信号を正しく処理できず、おおむね 10 回のダイヤルにつき 3 回程度の割合で異なる番号につながっていた。 暫定対処としては、設定を「ダイヤルパルス 10pps」「プッシュトーン」に変更すれば、不具合は発生しなくなる。 本格対処としては、11月10日をめどに、更新ソフトウェアを提供する予定。	ソフトウェア 障害	・NTT 東日本プレスリリース (2015.11.2) ・ITpro (2015.11.2)
1541	福島市 緊急速報 メール	2014	3		システム導入時から約1年半にわたって、緊急速報メールを送れない状態だった。 24日の市総合防災訓練までメールを使用する機会がなかったため、発覚しなかった。	職員が市内の携帯電話やスマートフォンにメールを送信しようとシステムを操作したが、携帯電話3社との通信が始まったものの、メールを送れないままシステムが途中で終了した。 システムを納入したベンダの担当者の初期設定にミスがあったほか、市がテスト送信をせず、気付かなかったことが原因。テスト送信をしなかった理由について、市は「緊急速報メールは重大な局面で使用するため市民を対象にした訓練にためらいがあった」とコメント。	設定 ミス テスト 漏れ	・福島民友ニュース (2015.10.28) ・福島民友ニュース (2015.10.31)	
		2015	10	30					
1542	ローソン おさいふ ポンタ	2015	11	3	既存のポイントカードであるポンタカードから、おさいふポンタへのポイント移行ができなかった。 チャージや決済は問題なく利用できた。	不明	不明	・日経コンピュータ電子版 (2015.11.6)	
		2015	11	4					18:00頃
1543	東京消防 庁 司令管制 システム	2015	11	10	11時30分	6分間、東京23区からの119番通報が不通となった。この間に7件の通報があったが、折り返し確認するなどした結果、大きな影響はなかった。	受付指令制御装置でソフトウェアの更新作業を実施しており、更新準備のための操作をしたところ、通報が受けられない状態に陥った。 ソフトウェア更新は月に1度実施しているが、これまでに不具合が発生したことはなかった。 東京消防庁では、今年3月と4月にも通報が受けられなくなるシステム障害が発生している。再発防止策を実施していたが、ソフトウェア更新作業時の障害対策については、検討が不十分だった可能性があるとしている。	不明	・日本経済新聞朝刊 (2015.11.11) ・日経コンピュータ電子版 (2015.11.12)
		2015	11	10	11時36分				
1544	三重銀行 ATM、ネット 取引	2015	11	24	11時30分	断続的に障害が起き、取引できない状態が続いた。	システムを管理するサーバが故障したため。 一度、13:30頃に完全復旧したと公表したが、16:15頃に再び障害が発生した。	ハード ウェア 障害	・上毛新聞朝刊 (2015.11.25) ・朝日新聞朝刊 (名古屋版) (2015.11.25) ・日経コンピュータ (2016.1.21)
		2015	11	24	19時15分				

No.	システム名	発生日時 (上段) 回復日時 (下段)				影響	現象と原因	直接原因	情報源
		年	月	日	時				
1545	JR 東日本 山手線新 型車両	2015	11	30		大崎駅では、車両のドアと駅に設置したホームドアが開かなくなった。目黒駅では、所定位置から 0.5 メートルオーバーランした。大塚駅では、所定位置の 1.5 メートル手前で停止した。	新型車両で初めて搭載した次世代車両制御システム『INTEROS (インテロス)』のソフトウェア不具合がトラブルの原因とみられる。 テストが不十分だった可能性があり、試運転期間中に電車で乗車率 40% に相当する重りを乗せたテストを行ったが、不具合は出なかった。だが 100% を超えるような加重については、実車ではなくコンピュータ上でシミュレーションを実施しただけだった。	ソフトウェア障害	・ITPro (2015.12.01) ・日本経済新聞朝刊 (2015.12.03)

な役割を果たすべきシステムが使えない状態で運用されていたことは肝を冷やす思いであるが、これまでに緊急事態が生じなかったことがせめてもの幸いである。

事例 1525 はスパコンの共同利用システムにおいて、保存データが破壊され一部は修復できず失われたにもかかわらず、数カ月後に利用者から指摘されるまで障害が認識されなかった事例である。事例 1524 及び次項で取り上げる事例 1527 も、影響はそれほど深刻ではないが、同種の事故である。

以上の 5 件はいずれも、一見正常に運転されているにもかかわらず、実は重大な問題を抱えたまま運用されていた事例である。不具合によってシステムがダウンするなどの現象が発生すれば、誤りは直ちに検知されるが、この種の事故は開発段階での綿密なテスト計画の策定とその着実な実施という基本的な対策でしか回避の方法はない。

## 5. 設計時の常識的事項の考慮漏れ

事例 1527 は、データの処理に当たって、例えば全角と半角といったデータの表現形式の違いがあっても、同じデータと見なして処理すべきであったにもかかわらず、それらを異なるデータとして取り扱っていたために、データの解析が意図通りに行われなかった事例である。

表記の揺れの存在は開発技術者にとっては常識であっても、利用者にとっては必ずしもそうではない。利用者の作った要件定義書などに表記の揺れの取扱い方法が明示されていないからといって勝手に判断せず、利用者に対し問題の所在を説明し、システムの目的に沿うためにはどのように取り扱うべきかを相談し合意した上で開発を進めることが必要である。

この事例と直接的には関係がないが、ある EC システム開発において SQL インジェクション対策を怠った結果、システムが攻撃を受けクレジットカード情報などの漏えいの被害を受けたことに対し開発業者の責任を問う民事訴訟裁判が行われた。開発業者は、システムの発注側からそのよう

な対策の実施の指示が明示されなかったことを理由に責任はないと主張したが、当時既に経産省や IPA から SQL インジェクション攻撃についての警告や対策のガイドラインが公表されていたにもかかわらず対策を怠った開発側の責任は大きいとする判決が確定した (東京地裁判決 平成 23 年 (ワ) 第 32060 号)。つまり、利用者から明示的に求められないからといって、開発業者にとって一般的となっている対策を怠り、プロの開発者としての責任を問われた例である。この観点から、先に紹介した事例 1527 は同種の問題を示唆している。開発技術者のプロとして求められる責任の大きさをあらためて感じる事例である。

## 6. むすび

2015 年後半 6 カ月間の情報システムの障害について、報道などをもとに整理し報告した。これらの事例の中から開発・運用にあたって参考にすべき多くの教訓を汲み取ることができる。これらを社会の共通の財産として共有し、少しでも事故を防ぎ、安心・安全な IT 社会に向けて地道な努力を続けていく必要がある。

SEC ではこのような活動の成果を教訓集として公表している。これまでに「情報処理システム高信頼化教訓集」[SEC1 2014] [SEC2 2014]、その改訂版として「情報処理システム高信頼化教訓集 (2014 年度版)」[SEC 2015] を公表した。さらに、今期から教訓がまとまるごとに逐次 WEB で公開することとしたので、参考にされたい。

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

### 【参考文献】

- [ 松田 2015 ] 松田晃一・八嶋俊介：情報システムの障害状況 2015 年前半データ、SEC journal No.42, Vol. 11, No2, pp.32-pp.37, Sep.2015
- [SEC1 2014] 情報処理推進機構 SEC: 情報処理システム高信頼化教訓集 (IT サービス編), 2014 年 5 月
- [SEC2 2014] 情報処理推進機構 SEC: 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 5 月
- [SEC 2015] 情報処理推進機構 SEC: 情報処理システム高信頼化教訓集 (2014 年度版) (IT サービス編), 2015 年 3 月

# 一般社団法人 組込みイノベーション協議会 活動紹介



一般社団法人 組込みイノベーション協議会 代表理事・理事長

鶴保 征城

## 1 はじめに

製品・システムの高機能化や多機能化が加速し、その実現手段として重要な役割を担うソフトウェアが急速に大規模化・複雑化していることは周知の事実であるが、これに加えて近年では、インターネットなどを介して様々な製品・サービスが相互に接続され、利用者がそれらを組み合わせて利用するようになりつつある。そのような状況下、単一の製品・サービスを利用していた頃では想定できなかったような不具合が生じたり、不具合の影響範囲が個々の製品・サービスを提供する事業者の域を超えて非常に広範囲になりつつあるのが現状である。

この問題に対応するためには、複数の産業分野を横断した取り組みが必須となるが、現在の我が国では、個々の産業が縦割りでの能力と効率を発揮する構造になっており、分野横断で生じる課題の解決や、分野横断的に重要

な技術を高度化させる事による産業競争力の強化といった取り組みが不十分な状態である。

一般社団法人組込みイノベーション協議会（以下、「EI協議会」と略す。）は、2014年12月に設立・発足し、初代の代表理事・理事長を鶴保征城（IPA/SEC 元所長・現顧問）が拝命した。

EI協議会では、先に発足した組込み IoT イノベーション議員連盟（以下、「組込み IoT 議連」と略す。）と連携することで、「政・官・学・産一体の取り組み」と「組込みソフトウェア産業の競争力強化」により「組込みソフトウェアによる産業の競争力強化」を図ることを目標としている（図1）。

そして、この「組込みソフトウェアによる産業力強化」をもたらすことで、「産業イノベーション」、ひいては、「社会イノベーション」に資することを狙いとして活動を開始した。

## 2 これまでの主な活動

組込みシステム／ソフトウェア産業の実態と現状の理解を深めるために、2014年4月の組込み IoT 議連の発足時より EI協議会と組込み IoT 議連は、精力的に意見交換や視察会・勉強会などの活動を行ってきた（表1）。今後も不定期ではあるが広範にテーマを設定し継続する予定である。

## 3 実態把握のための調査

組込みソフトウェア産業は、川上産業であり中小企業を中心である。また、最終製品の分野（サービス、輸送

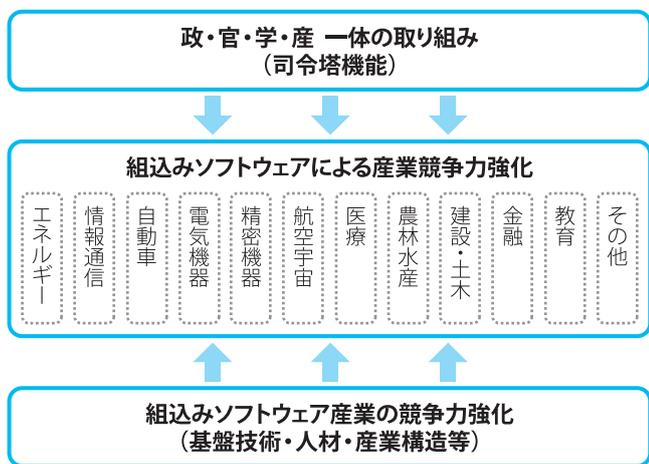


図1 産業競争力の強化

機器など)で見た場合、組込みソフトウェア産業はこれらの分野を横断する川上側の産業であり、その実態の把握は難しい。

過去には実態把握が行われた時期もあった。2003～2010年度の8年間に渡って経済産業省が主体となって実施した「組込みソフトウェア産業実態調査」、その後を受けて2011～2012年度の2年間にIPA/SECが主体となって実施した「ソフトウェア産業の実態把握に関する調査」では、組込みソフトウェア産業の実態を定量的に明らかにした。しかしその後調査は行われておらず、2013年度以降は客観的なデータに基づいた検討が十分に行えていない。一部の企業では資金調達に支障が生じるような事態まで発生している。

また、組込みシステム/ソフトウェアが重要な位置づけを担うCPS/IoT時代を迎えつつあるが、具体的な課題の把握を行うための基礎情報がない、または、不十分な状態にあり、政府等における産業政策や施策の立案にも影響が生じている。

EI協議会は実態把握を復活させるべく、2015年度に「組み込みシステム及び関連ソフトウェアに関する技術課題の検討」調査研究事業の一部を担当し、従前の調査の再開に向けた第一歩を踏み出した。

調査報告書は、2016年度にNEDOより公開される予定である。また、2016年度以降の調査主体・方法などについては未定であるが、所管省庁などと継続的に協議を行い、継続実施に向けて取り組む所存である。

## 4 活動の枠組み

前節で述べた実態把握調査は、データ収集のために先行着手したが、現在は全体取り組みの設計と始動を実施している。

組込みIoT議連が示した「組込みシステムの新たな展開に向けた提言」では、司令塔機能の実現、技術力の強化、人材の確保、産業の高度化、の主要4項目で構成されている。EI協議会ではIoT議連の4項目に対応する4つの委員会「司令塔(SC:Steering Committee)」、「技術(TC:Technical Committee)」、「人材(HC:Human resources Committee)」、「産業(IC:Industrial Committee)」を設置し(図2)活動内容の詳細化を行っている。

前節で述べた実態把握調査はSCの一部に位置付け、調査結果を戦略策定の材料とする。更にSCには「アカデミア連携(AL:Academic Liaison)」を設置し、TC、HC、ICを含むすべての活動に対して、アカデミアの知見を取り込めるように設計した。

表1 これまでの活動

視察会
<ul style="list-style-type: none"> <li>横浜スマートコミュニティ スマートセル・プロジェクト (2015年2月23日)</li> <li>組込みシステム開発技術展 (ESEC) (2015年5月13日)</li> </ul>
勉強会
<ul style="list-style-type: none"> <li>第1回 (2014年6月10日) 「日本版NISTの設立を目指して」 講師: 坂村 健 様 (東京大学大学院情報学環教授・工学博士)</li> <li>第2回 (2015年6月24日) 「組込みシステムによる産業イノベーションに向けて」 ～日本の製造業の危機感の共有～ 講師: EI協議会 平出理事、長谷川理事</li> <li>第3回 (2015年7月15日) 「組込みIoTシステムが創る世界に向けて」 講師: EI協議会 風見理事</li> <li>第4回 (2015年7月22日) 「先進国型イノベーションをソフトウェアが先導する時代の到来」 ～IoTやIoHの時代に日本のものづくりをどう方向性付けるか～ 講師: 小川 紘一 様 (東京大学 政策ビジョン研究センター シニア・リサーチャー)</li> <li>第5回 (2015年9月15日) 「IECにおける国際標準化活動」 講師: 野村 淳二 様 国際電気標準会議 (IEC) 会長、パナソニック株式会社 顧問</li> <li>第6回 (2015年9月24日) 「領域を横断する組込みイノベーションの実現に向けて」 講師: EI協議会 村山理事</li> </ul>

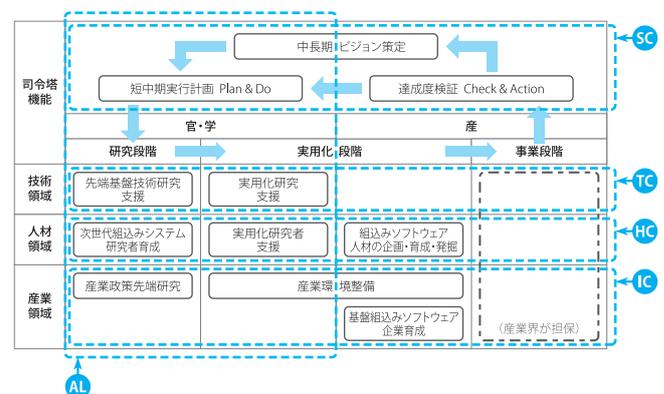


図2 今後の取り組みの方向性

2016年度はこれらの組織設計に基づき活動を具体化し、CPS/IoT時代の本格到来に向けて、組込みシステム/ソフトウェアを中心とした産業イノベーション、社会イノベーションに向けて活動を加速する予定である。

<所在地及び連絡先>

EI協議会の活動にご興味のある方(企業、団体、個人)はお気軽に下記までお問い合わせください。

東京都品川区西五反田 2-25-2 飯嶋ビル 5F  
 電話: 03-5434-7076  
 事務局担当: 伊藤 Email: katsuki.ito@esic.or.jp  
 URL: <http://www.esic.or.jp/>

# 組み込みシステムの新たな展開に向けた提言



組み込み IoT イノベーション議員連盟 会長  
衆議院議員

河村 建夫

## 1 議員連盟設立の目的と趣意

我が国における組み込みシステム関連産業の規模は GDP の 10.5% を占めており、この産業の成長失くしてアベノミクスにおける「第三の矢」、すなわち成長戦略の完成はありえない。そこで、我が国の「もの作り」産業の発展・グローバル化を推進するために、政産官学が一体となって組み込みシステムの重要性を認識し、もって日本の産業界の発展を図るべく、組み込み IoT イノベーション議員連盟を 2014 年 4 月に結成した。

組み込み IoT イノベーション議員連盟では、技術者の育成、ネットワーク化とセキュリティを確保した「つながりど安心」の実現、日本発の標準化作り、そして組み込み関連産業を支える中小企業の皆様の支援施策などを含め、国家の基幹産業の一つである組み込みシステム産業を振興し、産業イノベーションを促進することを趣意とする。

## 2 政策提言の背景

あらゆる製品・システムがネットワークでつながる本格的な IoT (Internet of Things) 時代を迎えつつある。これらの製品・システムには「頭脳」として電子部品(組み込みシステム)が搭載され、これらが組み込みソフトウェアによってコントロールされるようになる。組み込みソフトウェアは、ユーザの利便性・安全性を向上させると共に、製品・システムを製造する者の産業競争力を高める極めて重要なものである。よって、組み込みソフトウェア産業は下請産業であるという従来の認識を改め、広く社会や産業の発展を牽引する産業と位置付け直すべきである。

しかしながら、「ハードウェア中心の経済」から「ソフトウェアが競争力を左右する経済」への急速な移行に対応しなければならない中で、我が国の組み込みソフトウェア産業は、人材が足りない、短納期・低価格での受注が多い、小規模な企業が多いなど、課題を抱えている。加えて、今後の技術革新に伴い、組み込みシステムはますます高度化・複雑化し、様々な製品・システムがネットワークでつながるようになることから、組み込みソフトウェアの構成(アーキテクチャ)や開発工程(プロセス)の再構築をはじめ、業種横断的な活動が必要となる。更に、組み込みシステムにおいてもほかの産業と同じように垂直統合から水平分業への変革も予想され、競争力を維持していくためには、技術力の強化、人材の確保、標準化活動など戦略的な取り組みが必須である。

これらの背景から、組み込み IoT イノベーション議員連盟においては、我が国組み込みソフトウェア産業の競争力強化、持続的発展のために必要な事項として、以下の 4 点を提言する。

### 提言 1 司令塔機能の実現

様々な応用・産業分野に跨って搭載される組み込みソフトウェアという性質を考慮して、産学官が連携し、府省の縦割りを排して一体となって組み込みソフトウェアに関する政策を立案・推進できる体制を構築し、我が国が目指すべき姿を描き、その実現に向けた計画を策定し、必要なアクションを実施し、継続的にその達成度を検証する。

#### (1) 中長期ビジョンの策定

我が国の社会、科学技術、経済・産業などが目指すべき姿を描く 5～10 年のビジョン(中長期ビジョン)を

策定する。中長期ビジョンでは、地方創生、アジアとの連携なども考慮する。

## (2) 実行計画の策定

中長期ビジョンの実現を目指し、我が国の組込みソフトウェア産業の競争力を強化するための3ヶ年の計画(実行計画)を策定する。

実行計画では、これからの我が国の製造業にとって重要な業種横断的な基盤技術に重点を置き、研究開発から標準化まで、戦略的な取り組みを提案する。

実行計画には、技術、人材、産業の高度化などに関することを盛り込む。

## (3) 達成度の検証と改善

実行計画の達成度、施策の効果などを毎年度継続的に検証する。その結果、必要に応じ、実行計画や体制などを修正する。

また、組込みソフトウェア産業は幅広い分野で活動し、小規模な事業者も多く、産業実態の把握が難しいことから、我が国の組込みソフトウェア産業の実態を、継続的に調査する。更に、有識者及び関係者からのヒアリングなどを随時行い、組込みソフトウェア産業の課題の把握に努める。

## 提言 2 技術力の強化

我が国の製造業の競争力強化のため、組込みソフトウェアに関する先端技術などを戦略的に強化・提供可能なエコシステムを、産学官が連携し、構築する。

### (1) 先端技術への取り組み

次世代の組込みソフトウェア(プロダクト)技術や開発工程(プロセス)技術など、組込みソフトウェアに関する技術を広く俯瞰し、かつ、欧米の研究開発プロジェクトなども参考にしつつ、産学官連携の下、長期的に我が国製造業の競争力強化に資する組込みソフトウェアに関する先端技術を抽出し、官学を中心に研究開発を推進する。

### (2) 実用化研究への取り組み

産学官連携の下、広く我が国の製造業を支える組込みソフトウェアに関する基盤技術の研究開発プロジェクトに重点的に投資し、実用化を促進する。

また、今後IoT化がますます進展する分野と連携した実証実験(国家プロジェクト)を、IoT推進コンソーシアムなどとも連携し立ち上げる。実証試験においては、国民の安全・安心にかかわるような分野、業種横断的・分野融合的な分野などを重視する。

## 提言 3 人材の確保

組込みシステムを搭載する製品・システムは高度化・多様化し、我が国の組込みソフトウェア産業の人材の質・量及び多様性が、我が国の製造業の競争力に直結することとなる。しかしながら、我が国ではこのような人材が慢性的に不足していると言われている。優秀な人材を育成・確保する企業に対する支援策の検討を始め、以下のような対策を講じる。

### (1) 次世代の研究者育成

政府及び学界においては、組込みソフトウェアに関する先端技術の研究を重点領域の一つとする。また、この領域における教育体制・教育環境を充実させ、我が国の次世代の組込みソフトウェア産業を担う研究者を育成する。

### (2) 実用化研究の研究者育成

産業界は、組込みソフトウェアに関する基盤技術の実用化研究を研究者のキャリアパスの一つとすることを促進する。産学官連携し、この領域における教育体制・教育環境を充実させ、即戦力となる研究者を育成する。

### (3) 技術者のスキルの明確化と人材確保

産学官が連携し、我が国の組込みソフトウェア産業において中長期的に求められる技術者の職種・スキル(職能)及びその量的規模の明確化を図る。その上で、人材需給のミスマッチや人材不足を解消すべく、人材流動化、職能転換及び職能強化を進める。とくに、ビジネスモデルの変革によって不足すると予想されるアーキテクト、上級エンジニア、マネージャなどの職種の人材育成を重視する。

また、ITに興味を持っている若い人材が組込みソフトウェア産業において活躍する機会の拡大や、海外のIT人材の受け入れを進める。

## 提言 4 組込みソフトウェア産業の高度化

組込みシステムが搭載される製品・システムは、技術革新に加え、開発・供給者とユーザのグローバル化が進み、ビジネスモデルや産業構造が変わると予想される。このため、技術力の強化、人材の確保に加え、ビジネスモデルの変革を踏まえ、我が国の組込みソフトウェア産業をより強固な産業として成り立たせるための道筋を付ける。

### (1) 環境整備

世界中のあらゆる製品・システムがネットワークでつながり、組込みソフトウェアによってコントロールされ

るようになる中、情報漏洩や誤作動の防止などセキュリティや安全性・信頼性に関する技術、人工知能などの新たな技術などの導入が重要な課題である。また、組込みソフトウェアの開発・供給体制はグローバル化に対応する必要がある。

政府は、新たな技術を搭載した製品・システムを早期に普及させるため、必要に応じて規制緩和や基準の整備などの制度改革を進める。

更に、政府及び産業界は、組込みソフトウェア産業における下請取引の適正化に努める。

## (2) 国際標準化

産学官連携の下、技術革新やグローバルなビジネス環境の変化に合わせて、組込みソフトウェアに関する技術・製品などの標準化を推進し、製品開発の効率化などを図ると共に、我が国の技術・製品の安全性・信頼性などをアピールする。標準化に関しては、国際情勢を踏まえ、戦略的に取り組む。

## (3) 新たなアイデア、ビジネスモデルへの挑戦

現在、産業界は組込みソフトウェアの開発費用を人月ベースで算出しているケースが多いが、今後は、ソフトウェアの付加価値を適正に評価し、付加価値ベースでの対価回収モデルへと改革し、我が国の組込みソフトウェ

ア産業がより競争力のある魅力的な産業へと発展するよう努める。

更に、組込みソフトウェアに関する斬新なアイデアを持つ若い起業家やベンチャー企業などに対して、先輩起業家や投資家などが助言や資金的支援などを行う仕組みを産学官連携によって構築する。また、基盤技術を使った提案型、プラットフォーム型のビジネスに挑戦する組込みソフトウェア企業をとくに支援する。このため、基盤的なソフトウェアや新たなツールを扱うベンダの育成などを支援する。

# 3 おわりに

世界は「ハードウェア中心の経済」から、「ソフトウェアが競争力を左右する経済」へと急速にシフトしている。この「ソフトウェア経済」において、我が国は我が国の強みを最大限活用しながら競争力をますます高めていかなければならない。

このことを念頭に本提言の内容は、「日本再興戦略」などの国家戦略への反映を図るべく、組込みIoTイノベーション議員連盟所属の議員が一体となって継続的に活動を行っていく所存である。

### 組込みIoTイノベーション議員連盟所属議員一覧（2015年12月末現在）

#### 幹事

会長	河村 建夫	衆議院
幹事長	鶴保 庸介	参議院
事務局長	宮内 秀樹	衆議院（国土交通大臣政務官）
事務局次長	津島 淳	衆議院（国土交通大臣政務官、内閣府大臣政務官）
事務局次長	細田 健一	衆議院

#### 会員（五十音順）

##### ■ 衆議院

穴見 陽一、安藤 裕、池田 佳隆、今津 寛、今村 雅弘、うえの 賢一郎、大野 敬太郎、金子 万寿夫、神山 佐市、亀岡 偉民、菅家 一郎、北村 誠吾、小松 裕、佐々木 紀、左藤 章、鈴木 淳司（経済産業副大臣）、武部 新、土屋 正忠（総務副大臣）、富岡 勉（文部科学副大臣、内閣府副大臣（経済再生担当））、中谷 真一、中村 裕之、西村 康稔、福山 守、藤井 比早之、松本 文明（内閣府副大臣（国家公安委員会））、宮崎 政久、務台 俊介、八木 哲也

##### ■ 参議院

岩城 光英（法務大臣）、小坂 憲次、滝波 宏文、三宅 伸吾、森屋 宏（総務大臣政務官）

# 「人工知能の未来」

IPA 顧問、学校法人・専門学校 HAL 東京 校長

鶴保 征城

筆者が大学を卒業したのは1966年であるが、その前にゴードン・ムーアが自らの論文上で、「部品あたりのコストが最小になるような複雑さは、毎年およそ2倍の割合で増大する」と述べている。有名なムーアの法則であるが、その後「集積回路のコスト性能比は18カ月で倍になる」と公式化された。

当初は何十年も続くものとは信じられてなかったが、なんと50年も続いている。簡単な計算であるが、10年で100倍になり50年では100億倍になる。

一つの重要な発明がほかの発明と結びつき、次の重要な発明の登場までの期間を短縮し、イノベーションの速度を加速すると言われている。ムーアの法則が代表であるが、科学技術は直線的ではなく指数関数的に進歩する。

「京」は我が国が誇るスーパーコンピュータであるが、5年以上の期間と1000億円を超える経費を要している。一方、消費電力当たりの性能を競う世界ランキング「Green500」で2015年の1～3位を独占したPEZYコンピュータは、「京」に比べて遥かに短い期間と少ない経費で開発されている。更に、PEZYグループは2020年までに「京」の100倍のスーパーコンピュータを開発可能だとしている。

正にイノベーションがイノベーションを呼ぶ好例だと思う。収穫は<sup>ていげん</sup>逡減せず、<sup>ていぞう</sup>逡増あるいは加速するのである。

人間の経済活動は長い間、収穫逡減に苦しんできた。例えば、農業でいうと、最も豊かな土地から耕し始めて、そこから最大の収穫を得る。農場の経営を拡大しようと思えば、痩せた土地も耕作せざるを得なくなり、生産性は徐々に低下する。

収穫逡減はビジネスの基本法則のように思われてきたが、通信やソフトウェア業界は収穫逡増の原理で動いている。加入者増あるいは取引規模拡大が進めば進むほど、得るものが大きくなるのである。Google、Amazon、

Facebookに見られるように、ソフトウェアとインターネットをベースとしたビジネスも同じような特徴を持つ。

テクノロジーの進化が生み出す大きな流れはとどめようがない。2008年では1兆ページだったGoogleの検索WEBページは2013年の時点で30兆ページに拡大している。5年間で30倍に増えた計算になる。Googleコンピュータはこれら30兆のWEBページのすべてを人間よりも遥かに正確に記憶し処理している。

コンピュータ技術が今のペースで発達し続けると、ある地点で地球全人類の知能を超える究極のコンピュータが誕生すると言われている。そのコンピュータが更に自分よりも優秀なコンピュータを作り上げる、といった具合に連鎖的に進化し続ける可能性が指摘されている。これが技術的特異点（シンギュラリティ）と呼ばれる現象であり、2040年前後に到来するのではないかとされている。

シンギュラリティの仮説は現時点では広く認知されているとは言い難いが、IBMワトソン、コンピュータ将棋、東大入試ロボット、会話型ロボットなど人工知能の急速な進化は周知の事実である。

オックスフォード大学の研究によると、現時点で存在する仕事の50%近くが人工知能に置き換わるという。ロボットによる単純労働の置換は当然だが、医師、弁護士、税理士などプロフッショナルとされてきた仕事も無関係ではない。

人工知能が人類に対して良いものかどうか議論することは必要だ。たぶん、過渡期には色々な問題が起こると思うが、最近、SNSで中学生や高校生が、我々の気付かないような視点でシンギュラリティの議論をしていてすごく勇気づけられる。我々よりもっと若い世代が、新しい世界や価値観を違和感なく受容していくのではないだろうか。



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

ISBN: 978-4905318361  
A4 変形判・180 頁  
定価 1,852 円 (税抜)  
2015 年 11 月 18 日刊

## 組込みソフトウェア開発データ白書 2015

～定量データ活用で見て来た組込みソフトウェア開発の一面を初公開～

組込みソフトウェア開発においても、品質に対する説明責任やコスト競争力強化のために、定量データを活用したプロジェクトマネジメントや組織改善を推進する企業が年々増加しています。

IPA/SEC は、定量データ活用を組込みソフトウェア業界に広く推進させることを目的に、公的機関としての中立的立場を活かし、組込み分野に特化したプロジェクトデータを収集し、分析を実施しました。

「組込みソフトウェア開発データ白書 2015」は、2013 年 4 月～2015 年 3 月までの 2 年間に、組込み分野におけるソフトウェア開発企業 10 社から提供を受けた総計 174 件のプロジェクトデータを分析したものです。

本書では、収集データのプロファイルと、組込みソフトウェア開発プロジェクトの規模・工数・工期の関係や、生産性・信頼性などの分析結果を掲載しています。



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

ISBN: 978-4905318378  
A5 判・86 頁  
定価 463 円 (税抜)  
2015 年 11 月 18 日刊

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

～定量データ活用による組織の開発・管理力向上～

「定量データ活用は手間とコストがかかるだけでメリットが無い」という先入観を持たれている組込みソフトウェア開発企業の経営者層に向けて、経営者の視点で定量データ活用による利点を訴え、定量データ管理の導入を支援する啓発書です。

コストをかけなくても定量データ収集の仕組みが構築できることや、定量データを活用した具体的なプロジェクトマネジメントの例

や指標を活用した組織の改善方法なども紹介しています。「組込みソフトウェア開発データ白書 2015」と合わせてご利用ください。

## 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～



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

A5判・216頁  
定価 926円 (税抜)  
2015年5月29日刊

IT やそれ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、全体として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ます。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ます。

本ガイドブックは、このような背景のもとで、製品・サービスを提供する事業者が、広い視点で

品質要求を洗い出し、確実に設計・実装し、かつ定量的に評価するための指針を規定した国際規格「SQuaRE」<sup>※1</sup>について、上記の背景とそこで必要となる品質の考え方、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たものです。現場の技術者やリーダーだけでなく、会社経営者の方々にもご活用いただけます。

※1 SQuaRE : 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。

## つながる世界のセーフティ&セキュリティ設計入門

～IoT時代のシステム開発『見える化』～



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

ISBN: 978-4905318354  
A5判・90頁  
定価 556円 (税抜)  
2015年10月7日刊

スマートフォンで家電を制御するサービスや、クラウドを利用して健康器具のデータを管理するヘルスケアサービスなど、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がつながるサービスが出現すると見込まれています。このような「つながる世界」においては個々の製品の問題がシステム全体の問題となります。とくに、安全性(セーフティ)やセキュリティの問題が懸念されます。本書では、組込み系

のソフトウェア技術者や製品開発責任者(経営層)を対象に、セーフティ設計(設計の段階で安全を作りこむこと)・セキュリティ設計(設計の段階で脆弱性の低減や脅威への対策を考慮に入れること)とその設計品質の見える化(エビデンスに基づき、第三者でも容易に理解できる表記で論理的に説明すること)の導入を促すことを目的に、設計の重要性やリスク分析手法、設計手法、見える化手法、経営層の関与のあり方などを解説しています。

上記書籍は、IPA 直販でご購入いただけます。詳細は Web サイトにてご確認ください。SEC BOOKS : <http://www.ipa.go.jp/sec/publish/index.html>  
※一部の書籍は amazon でもご購入いただけます。また、お近くの書店でお取り寄せが可能な場合があります。詳しくは各書店にお問い合わせください。

## 編集後記

今号の特集は「IoT時代に求められる開発スタイル」としました。ビジネススピードが極端に速くなり、要件も決まっていなのに短期間に開発しなければいけない。しかし、従来のウォーターフォール型の開発では納期だけでなく、変化する要件に対応することも難しい。また、発注者と開発者といった役割分担も何度もリリースしユーザの反応をみて改善を繰り返すといった開発スタイルにはなじまない。そこで、現代の変化する要求に対応した開発スタイルにはどのようなものがあるか、適用領域や課題はどこにあるのか、今回集めてみたいと思って特集にしました。

今回特集を組むまで、非ウォーターフォール型の各種開発スタイルは、欧米からの手法の輸入であると思い込んでいたのですが、そのコンセプトには日本の製品開発の考えを取り入れているものも複数あり、このような手法の発生の過程を知ると今回紹介した開発手法たちに親しみを感じるようになりました。

(編集長)

## 編集部より

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

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

## SEC journal 編集委員会

編集委員長	遠藤秀則
編集委員 (50音順)	荒川明夫
	石橋正行
	日下保裕
	千脇誠司
	中尾昌善
	長谷川佳奈子
	三原幸博
	室修治
	山下博之
	和田恭



梅の花 (堀切菖蒲園) (撮影：M.Ooe)

SEC journal 第11巻第4号 (通算47号) 2016年3月1日発行

© 独立行政法人情報処理推進機構 2016

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

※本誌は「著作権法」によって、著作権等の権利が保護されている著作物です。  
※本誌に掲載されている会社名・製品名は、一般に各社の商標または登録商標です。

# SEC journal 論文募集

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

## 論文テーマ

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

## 論文分野

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

## 応募要項

締切り：1月・4月・7月・11月 各月末日

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

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

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

## SEC journal 論文賞

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

IoT時代に活躍する【組み込みシステムの腕利きエンジニア】を目指す！

## 国家試験 エンベデッドシステムスペシャリスト試験

### 高度な実践能力の証明に！

- ▶ 身近な場面を想定した出題を通して、最適な組み込みシステム実現のために必要となる高度な実践能力（レベル4）を問います。

**レベル4の定義**：専門分野において、自らのスキルの活用によって、独力で業務上の課題の発見と解決をリードするレベル。

#### 技術要素

プロセッサ、メモリ、バス、計測・制御、リアルタイムOS、プラットフォーム、電気・電子回路、ネットワーク、セキュリティ

#### 開発技術

- ・要求分析の実行とレビュー
- ・設計の実行とレビュー
- ・テストの実行とレビュー

#### 管理技術

- ・開発環境マネジメント
- ・知財マネジメント
- ・構成管理、変更管理

- ▶ 近年の試験では、「無線通信ネットワークを使用した安全運転支援システム」、「3次元複写機」、「通信機能をもつ電子血圧計を用いた健康管理システム」、「非接触型ICカードを使用した入退場ゲートシステム」などのテーマを出題しました。
- ▶ 自動車、家電、モバイル機器などに搭載する組み込みシステムや重要インフラの制御システムを、ハードウェアとソフトウェアを適切に組み合わせて構築し、求められる機能・性能・品質・セキュリティなどを実現できる組み込みエンジニアを目指す方に最適です。

### 試験概要

**【試験区分】** エンベデッドシステムスペシャリスト試験（情報処理技術者試験 高度試験の1区分として実施）

**【日 時】** 年1回の実施（毎年4月第3日曜日）

**【申込受付】** 毎年1月中旬から2月下旬（予定）までWEB・郵送で申込み受付

詳しくは、Webページをご覧ください。<http://www.jitec.ipa.go.jp/index.html>

試験概要の最新情報、過去問題、活用事例などをご紹介します。



# IPA Better Life with IT

SEC journal No.44  
第 11 卷第 4 号 (通卷 47 号)  
2016 年 3 月 1 日発行

© 独立行政法人情報処理推進機構

ISSN 1349-8622



古紙パルプ配合率70%再生紙を使用

