

SEC journal

46

巻頭言

築田 稔 一般社団法人 組込みシステム技術協会
会長

所長対談

新たな時代を担う組込みシステムの 技術者に求められるものとは

井上 克郎 ESIP 組込み適塾 塾長 大阪大学 大学院 教授

論文

プロセス改善技術者育成コースの設計と実装

久野 倫義 三菱電機株式会社 設計システム技術センター／中島 毅 芝浦工業大学 工学部 情報工学科

芝田 晃 三菱電機株式会社 インフォメーションシステム事業推進本部 インフォメーションシステム業務部

近藤 聖久 三菱電機株式会社 設計システム技術センター／小笠原 公一 三菱電機株式会社 生産システム本部 生産技術部

特集

組込みソフトウェア開発

組込みソフトウェア開発をめぐる新潮流

山下 博之 SECシステムグループリーダー

モデルベース開発とコード解析を用いた組込みソフトウェアの開発

熟練者の知識流通を支える開発環境

小林 展英 株式会社デンソークリエイティブ／平鍋 健児 株式会社チェンジビジョン

組込みとOSSとアーキテクチャ

竹岡 尚三 株式会社アックス 代表取締役

組込み技術者育成へ初等中等教育からの改善

大原 茂之 株式会社オブテック 代表取締役会長 一般社団法人スキルマネジメント協会 理事長 東海大学名誉教授

報告

Embedded Technology West 2016 (ETWest2016) 出展報告

連載

情報システムの障害状況2016年前半データ

松田 晃一 IPA顧問／八嶋 俊介 SECシステムグループ 主任

Column

IoTイノベーションの期待

1

巻頭言

IoTの核心技術を担う組込みシステム技術

築田 稔 一般社団法人 組込みシステム技術協会 会長

2

所長対談

新たな時代を担う組込みシステムの技術者に求められるものとは

井上 克郎 ESIP 組込み適塾 塾長 大阪大学 大学院 教授

8

論文

プロセス改善技術者育成コースの設計と実装

久野 倫義 三菱電機株式会社 設計システム技術センター／中島 毅 芝浦工業大学 工学部 情報工学科

芝田 晃 三菱電機株式会社 インフォメーションシステム事業推進本部 インフォメーションシステム業務部

近藤 聖久 三菱電機株式会社 設計システム技術センター／小笠原 公一 三菱電機株式会社 生産システム本部 生産技術部

16

特集：組込みソフトウェア開発

組込みソフトウェア開発をめぐる新潮流

山下 博之 SECシステムグループリーダー

モデルベース開発とコード解析を用いた組込みソフトウェアの開発

熟練者の知識流通を支える開発環境

小林 展英 株式会社デンソークリエイティブ／平鍋 健児 株式会社チェンジビジョン

組込みとOSSとアーキテクチャ

竹岡 尚三 株式会社アックス 代表取締役

組込み技術者育成へ初等中等教育からの改善

大原 茂之 株式会社オブテック 代表取締役会長 一般社団法人スキルマネジメント協会 理事長 東海大学名誉教授

40

報告

Embedded Technology West 2016 (ETWest2016) 出展報告

荒川 明夫 SECプロモーショングループ 主任

43

連載

情報システムの障害状況2016年前半データ

松田 晃一 IPA顧問／八嶋 俊介 SECシステムグループ 主任

50

Column

IoTイノベーションの期待

鶴保 征城 IPA顧問、学校法人・専門学校HAL 東京 校長

51

書籍紹介

52

編集後記

SEC journal 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

IoTの核心技術を担う組込みシステム技術

篠田 稔

一般社団法人 組込みシステム技術協会
会長



我が国経済は、海外に起因する不安定要素はあるものの、政府の経済政策や日銀の金融緩和などを背景に輸出関連企業を中心に収益の改善が図られ、緩やかな景気回復が続いている。

他方、日本の産業構造は、デジタルイノベーション、オープンイノベーション、更にはAIの実用化も相まって、新たな時代を迎えつつある。その中において、組込みシステム技術の延長上に展開されるIoTの概念はこれらを支える共通基盤であると言えよう。

ネットにつながるIoT時代の特徴は、技術もさることながら、モノづくり、サービスにかかわらず、非ITベンダーをも含め、多くのステークホルダーが関与することである。従って、これらをいかにまとめ上げるかという協業推進力がビジネス成功の鍵の一つとなるであろう。

かつて組込みシステム業界は、サポーターインダストリーという言葉で象徴されていたように川下のモノづくりを中流、上流で支える企業の集合体として捉えられていた。もちろんどのような時代においてもモノづくりを支えることは重要である。しかしながら、IoT時代においては、サービスと川下製品やセンサを結び付けるコーディネーションが成功の鍵であり、そこに豊富な実現技術を持つ組込みシステム業界に新たな役割が期待されている。

サービス提供に当たっての重要課題は広義の信頼性である。個々の製品の信頼性確保はもちろんのこと、ネットにつながることによって生じる様々な、場合によっては新しい品質問題、セキュリティ問題などIoT時代の課題を解決してゆかねばならない。ここに当協会もIPA殿との連携がますます重要となるものと確信し、近年ソフトウェア高信頼化センター（SEC）殿と当協会の下記委員会で共同活動を進め成果を上げている。

1. 安全性向上委員会安全仕様化WG
2. 技術高度化委員会IoT技術委員会

この度の「組込みソフトウェア開発特集号」発刊は、上述の一連の方向性に則りIoTの基盤である組込みソフトウェア開発の重要性を再認識させる時宜を得たものである。

本年、設立30周年の節目を迎える当協会は、組込みシステム技術の高度化事業を中心に据え、関連するIoT技術の深耕、更にサービス事業などの他団体との連携により我が国の産業振興に寄与する所存である。

新たな時代を担う組込みシステムの技術者に求められるものとは

ESIP 組込み適塾 塾長
大阪大学 大学院 教授

井上 克郎

SEC所長

松本 隆明

IoT、AI、ビッグデータの進展による第4次産業革命の時代の中核を担うべき組込みシステム技術者の将来像について考える。組込みシステム産業振興機構 (ESIP) が推進する「組込み適塾」は、関西を拠点として先進的な組込みシステム技術者の育成を目的として幅広い活動を行っている。その塾長であり、大阪大学 大学院の教授でもある井上氏に、人材育成と組込み技術の未来について考えを伺った。

組込み適塾で中堅クラスの技術者育成を

松本 近ごろ、IoTやAI、ビッグデータなどの技術が大幅に進展し、システムの開発は大きく変化しつつあります。ハードウェアとソフトウェアの垣根がなくなり、ひとかたまりにして考えていかななくてはいけない時代となってきました。今後は、ハードウェアとソフトウェアをつなぐハブとも言える

組込み技術がますます重要な役割を果たすと思います。ESIPの組込み適塾塾長として業界の技術者育成をリードしている井上先生に、組込み技術者の将来像についてお伺いしたいと思います。まず、組込み適塾の取り組みについて紹介していただけますか。

井上 組込み適塾は、2008年に第1期がスタートし、今年で9期を迎えます。もともと関西の地場産業には、家電、電子機器、自動車などがあります。中でも、中小企業に活性化してもらうために力を入れるべきところを考え、組込み技術に焦点が当てられました。関西経済連合会が中心となって組込みシステムESIPを設立し、教育事業を進めよう、という運びになったのです。

教育プログラムを考え

ていく上で、不足しているのは中堅クラスの技術者ではないかという意見が多くありました。会社の将来を引っ張っていける人材を育成する、そんな教育システムが不足していたのです。大企業なら十分な教育制度があるのかもしれませんが、関西の地場産業を担う中小企業には荷が重い。そこで、寄り集まって教育システムを作りましょう、と立ち上がったのです。

産業界からのニーズが出発点で、大学がそれをお手伝いします。大学で行われる教育とは趣が異なり、会社の中ですぐに必要な人材を育成するためのカリキュラムです。

企業と連携する強みは、“実際の現場で必要とされる人材”を真剣に考えられる点です。それを繰り返し検証していき、毎年カリキュラムを改善します。改良を重ね続け、今年は9回目を開講しています。コース数は当初に比べて3倍になり、最先端のIoTやセキュリティの分野も用意しています。わざわざテコ入れをしなくても、自律的に発展しており、関西の地場産業を発展させるという狙いに貢献できているだろうと自負しています。

松本 産業界のニーズに基づいている、ということは、産業界からの要望でカリキュラムなどを設定しているのでしょうか？

井上 部会で大まかな方針を決めて、ワーキンググループで細かく検討していきます。ワーキンググループでは前年の実績や受講生・講師アンケートを集計、綿密にチェックをして、今後新たに必要になりそうなテーマや先端技術などを見極めます。そこに、講師の評価を加味して、カリキュラムを決めています。

松本 中堅クラスの育成が主な目的になるのでしょうか。

井上 申し上げた通り、中堅のトップクラスで活躍できる人材を輩出するのが最初のスタートでした。途中から、ハンダ付けから教えるような初級講座もスタートしています。「実装エンジニアリングコース」が初級者向けで、次に「アーキテクチャ設計コース」があり、組込み適塾の中核を成しています。その上にある「アドバンストコース」はかなり上級の講座です。



井上 克郎 (いのうえ かつろう)

1979年 大阪大学基礎工学部情報工学科卒。
1984年 同大学院博士課程修了。同年同大・基礎工学部・助手。1984～1986年 ハワイ大マノア校情報工学科・助教授。1991年大阪大学基礎工学部・助教授。1995年同学部・教授。2002年 大阪大学 大学院情報科学研究科 教授。博士(工学)。ソフトウェア工学の研究に従事。

今年の組込み適塾の特徴としては、IoT技術やビジネスを教える講座が増えています。今や組込みシステムはボードを設計してセンサやアクチュエーターを付けるだけでなく、IoTでつながる先の機器など、システム全体を考える必要があります。そのために欠かせないネットワークやセキュリティはもちろん、ビジネスデザインなどを教える講座を開設しています。

松本 「ビジネスデザイン」というと、幅が広いのではないのでしょうか。その場合、組込み技術者だけでなく、ITシステムのデザインや全体のアーキテクチャを考えられるような人材も対象にするのですか。

井上 対象者を別に設定するだけでなく、組込み技術からスタートした人がビジネスデザインも扱えるようになって欲しいと考えています。そこに特化するの難しいかもしれませんが、技術者が「組込みのことだけしか考えていない」という状態は避けていただきたいのです。

松本 そうですね。ビジネスのセンスを持った上で開発していくことが、これからは必要になるでしょう。

スタート直後のリーマンショックを乗り越え、遠隔地での開催まで発展

松本 今期で9回目とのことですが、それだけ長い期間継続されてきて、ご苦労もあったのではないですか。

井上 組込み適塾は2008年からスタートしましたが、同年にいきなりのリーマンショックが起り、経済の悪化から受講生は計画通りには集まらず、会員の企業様をお願いして参画いただくなど、波乱含みのスタートでした。

また、毎回改良を重ねていくことは非常に大きなエネルギーが要ります。誤解を恐れずに言うと、大学は改良を重視していないようなところがありますが、企業は大変に熱心です。「良いものを作らなあかん」と、新しい講座を企画したり、新しい先生をお願いしたり、スケジュールを改良するなど、事務局は本当に大変でなかなか落ち着くことはありません。

最近では遠隔地でも受講できる制度を整え、東日本大震災の復興支援も兼ねて東北で実施しました。テレビ会議で実施するので、場所を借りたり、機器のセットアップなどもなかなか手間がかかります。うまくいくか不安もたくさんありましたが、結果は上々でした。東北で受講した人達もしっかりと勉強をしてくださり、遠隔地の不安をある程度払拭できたと思っています。全受講生の中から選ばれる優秀学生に、遠隔受講者が選ばれた年もありました。

松本 一度修了した方がまた別のコースを受講するというケースもありますか？

井上 そこは私たちも期待しているところではあります。初級を受けた翌年以降に中級、上級を受講するというケースです。少しずつそういう人も増えてきていましたが、まだそれほど大きな流れにはなっていません。

卒業生同士の横のつながりも大切だと考えており、同窓会などの機会も積極的に提供するようにしています。ただ、受講生が自立的に同窓会をやり出すまでには至っていません。

松本 修了した人は即戦力でしょうから、企業はなかなか

業務以外の活動に参加させられないという実状があるのでしょうか。

井上 適塾の卒業生が今度は企業の中で講師になっているという話をよく聞きます。組込み適塾で使用したテキストと同じものを使い、同じ内容をレクチャーするようです。

IoTの台頭で、組込み技術者にも包括的な知識が必要

松本 今、IoTやセキュリティの講座もあるとお伺いしました。組込み技術を取り巻く環境は大きく変わりつつあると思います。データを中心に考える必要があると思いますが、求められるスキルや、その方向性は変わりつつありますか？

井上 部会では、みなさん熱心に議論なさっています。取り巻く環境はすさまじいスピードで変化しています。単に「マイコンの話をしれば済む」という時代は終わりました。IoTで様々なものにつながるわけですから、先を見て知識を吸収していかなければ一流の技術者にはなれないでしょう。ネットワーク、セキュリティ、ビジネス、AI、ビッグデータ……挙げればキリがないですが、本当に様々なことを押さえておく必要があるのです。

ただし、朗報があるとすれば、部品化が進んでいるということです。ソフトウェアなどもオープンソース化が進み、パッケージを利用すれば細かい開発をしなくてもよくなりました。ある意味で、細かい技術なら切り捨てられる、という場面も出てきています。それらを加味して、広い視野で全体のバランスを見据えることが求められるでしょう。ある部分はしっかり技術を身に付け、パッケージで済むところは利用する。適宜判断してうまく切り分けられる技術者になって欲しいと思っています。

松本 ハードウェアやアーキテクチャなどもある程度分かなければなりませんね。そう考えると、組込み技術者のスキルレベルは、今最も高い次元で求められていると言えるのかもしれませんが、エンタープライズ系であれば、ハードウェアやネットワークの話はある程度仮想化できており、アプリケーションレベルの話がほとんどです。ところが組込み技術者はそこまですべて自分たちで開発していかなければいけないわけですね。

井上 そうですね。マイ



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・エンジニアリング・センター(SEC)所長。博士(工学)。

コンの1ビットにかかわる細かな部分から、ネットワークにつながったビジネスモデルのことまで広範囲にわたるため、学ぶのは大変だろうと思います。すべてを網羅するのは確かに大変ではありますが、組込み適塾で用意している様々なカリキュラムを自分の好みで受講していただければ、近づけるのではないかと考えています。そのために、必要なものを適宜選べる環境を用意しています。

組込み適塾のカリキュラムは、受講生が分かりやすいようにコース・科目を設定しています。「こちらのコースに沿って学べばIoTに強くなれる」「こちらのコースならビジネスに……」という設計です。

松本 講座ごとに、受講生が多い、少ないといった人気の違いはありますか？

井上 ベーシックなものの人気が高いようです。アーキテクチャ設計コースの中の「ベース科目」では、ソフトウェア開発のV字モデルに沿って、開発工程をすべて網羅できます。体系的に学べるため受講生が多く、人気があります。中でも、設計コースと実装コースの共通講座であるリバーズモデリングは特に人気が高いようで、差分開発を対象にしたもので、最近の開発によく利用されています。現場のニーズに即したものはやはり人気です。

松本 基本的な内容を学べる講座の人気が高いのでしょうか？

井上 その傾向はあると思います。上級者向けのアドバンストコースで、「流行りだから受講生が多いのではないか」と思った講座にそれほど人が集まらない……ということもありました。人気を先読みすることは難しく、蓋を開けてから驚くことがままあります。

松本 基本の技術を身に付けていないと、アーキテクチャやビジネスデザインはなかなかできないのでしょうかね。

井上 ベーシックな知識を身に付けてから、高度な範囲に首を突っ込むようにしていただきたいとは思っています。基本的な部分が身に付いていないうちに上級レベルだけをかじっても、学ぶものは少ないでしょう。ソフトウェアだけでも、ハードウェアだけでも足りないのです。システムとして考え、発想できるように育てていただきたいと考えています。本来は2～3年かけて受講していただければ理想的なのですが、残念ながら優秀な人をそれほど外に出してはおけないというのが中小企業の実状でもあります。

松本 コースとしては、1年で修了するのですか？

井上 例年、6月頃からスタートし、10月末に終わるというスケジュールです。期間はだいたい、4～5カ月となります。以前は1～1.5カ月間毎日みっちりやっていましたが、その間実務がストップしていました。現在は科目単位で間隔を取り、実務とバランスが取れるようにしました。

分野をまたがった交流ができる

松本 実践的なスキルはなかなか大学では教育できないため企業内教育もやらなければなりません、企業内ではその業務に依存することになり、体系的な教育ができないという課題もあります。体系的な教育を受ける機会はずも少ないですね。

井上 大企業は社内で体系立てて教育ができるのかもしれ

ませんが、ESIPに参画していただいている関西の中小企業ではなかなかできません。

松本 そんな中で、システムズエンジニアリングやビジネスデザインなどの能力は、どのように身に付いていくのでしょうか？ 座学だけではなかなか難しいと思います。

井上 最終的には、OJTが必要となります。ただし、OJTだけに頼るなら私共の組込み適塾の活動にあまり意味がなくなってしまう。適塾では基本的に、現場ビジネスに必要なケーススタディをお教えするという立場。もちろん実体験はできないので、その一手手前の演習にたくさん取り組んでいただきます。

考え方としては、100%は難しくても、80%くらいの経験ができるということです。受講された方には自信を持っていただいて良いと思います。

松本 現場では派生開発や差分開発が多くなり、全体のアーキテクチャから設計する機会はなかなかありません。正にそういった技術をどう身に付けるのか、企業として問題視しているという声をよく聞きます。

井上 そういう方にこそ、様々なケースを勉強していただきたいです。フルに開発した経験がなくても、様々な知識を持つことは大切です。例えば、アーキテクチャを設計したことがなくても、様々なオープンソースなどを使ってシステムが構築されているという全体像が理解できます。実際の現場ではないので製品に直接つながることはありませんが、環境は整っていますから、勉強する気になればいくらでもできるということです。

また、仲間がいることも大きな魅力ではないでしょうか。社内や講座の受講生などと一緒に切磋琢磨して勉強していき、擬似的な体験ができる良い機会としてもらえればと思っています。

松本 「切磋琢磨の場がある」ということは重要だと思います。企業ではそのような機会は減ってきています。お互いに技術を磨きたくてもそうした場がなかなかないという人にはありがたいでしょうね。

井上 受講生同士で刺激し合える環境があります。もし会社で孤軍奮闘していたとしても、外の世界で同じような人がいることが分かります。刺激になり、もっと勉強しようという気持ちになれるようです。横のつながりというのは大事なのだと感じさせられます。

松本 様々な企業と交流できることは貴重ですね。組込み技術の場合は、車載器や電化製品など、特定の分野に特化してしまう傾向があると思います。その結果、技術や知識の範囲が狭くなってしまいます。横のつながりができ、様々な分野の人と情報交換することで、「こちらの分野で組込みはこのような仕組みなんだ」と分かることは大きな意味があるのでしょうか。

井上 そうですね。本当に様々な企業の方が来てくださいます。新幹線、はかり、自動車など……。そのような人たちが一緒に勉強するのは、お互いの刺激になります。

入社直後からベテラン技術者まで、キャリアパスを明解にして幅広く対応

松本 受講生には、入社何年目くらいの方が多いのでしよ

うか？

井上 入社1年目の人もいるし、20年選手もいます。メインの層は5～10年くらいでしょうか。初級コースは1～5年目、中級コースは5～10年、一番上のアドバンスコースは7～20年くらいが多いと思います。5年以上のベテランでも、自動車なら自動車しか見てきていません。ほかの受講生と交流してほかの業界のことも分かり、充実感を得ていただけていると感じます。

ベテランの受講生が相手だと、講師もなかなか大変です。経験が豊富ですから、いいかげんなことを言うてしまうとやり込められてしまいます。

松本 生徒のほうがむしろ詳しいということもあるかもしれませんね。実経験で身に付けた技術がありますから。

井上 そうなんです。スタート当初は中級程度の人たちを上引っ張り上げるのをメインの狙いとしていたのですが、今は初級コースの人たちを真ん中くらいに引き上げたいという狙いもあります。ドメインごとにテスト関係、マネジメント関係などの講座も入れており、内容はIPA提唱の組込みスキル標準「ETSS」に準拠しています。

松本 キャリアパスが明解になっているほうが、受講生もやりやすいと思います。自分でどういうスキルを身に付ければ良いのか分かりますから。

井上 企業の方は、キャリアパスを大事にしているのです。そこからのスタートでした。大学はこういった発想ではないのです。

松本 企業の中でも、CDP (Career Development Program) をしっかりと描いているケースは多くありません。作ることの重要性は認識されていますが、どうしても現場重視で個別にならざるを得ないためなかなか難しいようです。

井上 現場の人たちから見ると「きれいごと」と思われがちですが、各社の「きれいごと」を集めているので、組込み適塾の考えとして説明しやすい。良い道具になっていると思います。

松本 講義は大阪中心に行われているのでしょうか。

井上 大阪だけでなく、去年は名古屋でも実施しました。今年は神奈川でも開催します。色々な場所で遠隔地展開していく予定です。

松本 組込みには地域性があるような気がしますね。「名古屋は自動車」など……。大阪は家電が多いですか？

井上 関西は、家電も多数ありますが、受講生の従事している業務を見る限り自動車を扱う企業も増えてきました。昨今は自動走行などの開発が盛んになっているので、これは全国的な流れなのかもしれません。

松本 今は自動車が日本の産業を引っ張っていると言えるのかもしれません。しかし、自動車も、セキュリティやIoTのことを考えなくてはいけない時代です。ハッキングで外部からコントロールされてしまったという事例もありました。従来、組込みシステムは単体で動くため、セキュリティの心配はありませんでした。ところが色々なものがつながるようになり、エンタープライズ系独自だった技術も網羅していかななくてはなりません。

井上 様々なつながりが出てきて、便利になった代わりに危険にさらすことにもなっていますね。エンジニアとしても幅広い知識が必要です。特にセキュリティの場合には、た

くさんの知識を身に付ける必要があります。ひとことに「セキュリティ」と言っても、数学やプログラミング、ハードウェアなど広範囲で要求されます。一つの分野を勉強すれば良いわけではなく、様々なパターンを身に付けてもらいたいと思っています。

松本 IoTのブームで、ネットワーク系技術の重要度が増しているとお考えになりますか？

井上 そうですね。ネットワーク系技術は組込みシステムで特に重要視され始めています。ユーザとどのようにインタラクションするのか、という知識が重要です。究極的にはそれがビジネスモデルに行き着くのだと思います。すべて詳しく知るわけではなくても、エンジニア、ネットワーク、その先のユーザに関する知識をある程度は見聞きできるようにしておいていただきたいですね。あくまでも、組込み技術者としてコアの部分はボードとセンサでありながら、その先も分かるというレベルが理想的です。

松本 IoTになって様々なものがつながり、更にユーザまでつなげられるようになってくると、利用者品質を確保することも必要になってきます。いかにユーザが使いやすいようにするか、という視点が重要になりますが、こういう利用者品質の重要性をどうお考えになりますか？

井上 ボードの中で発想を固定すると、使いにくいものしかできません。技術者自身が、自ら色々なものを使ってたくさん経験を積み、ボードの開発に活かしていただきたいと思いますね。

組込み産業を発展させるキーワードは「つなぐ」「見つける」「生み出す」

松本 新聞報道によればESIPでは「情報家電」「ロボット家電」「ヘルスケア」「環境エネルギー」の4分野で、IoTを中心に関連企業の支援に乗り出すとのことですが、支援の具体的な内容を教えてください。

井上 新聞で紹介されたのは、ESIPの今年度からの中期ビジョンについての掲載です。ESIPは、これまでメーカを中心に情報家電、ヘルスケア、ロボット、環境エネルギー分野の組込み産業を発展させるために取り組んできました。今年度からは、記事中に紹介されていたようにそれらにIoTを取り入れることで、新創出やオープンイノベーションを加速させようと考えています。既にESIPには、IoTを取り扱っている企業やこれから取り組もうとしている企業が、地域・ドメインを超え続々と入会・活動に参加してきています。

松本 記事にはIoTをテーマにした人材育成を行うことも書かれていました。具体的には、そのドメインに特化したスキルを身に付けるカリキュラムなどを増やすのでしょうか？

井上 教育事業としては、体系的や普遍性を重視する方向です。ESIP内の組込み適塾は、もともと普遍性や汎用性の高い教育を目指していたので、そこにIoTに対応できる人材を育てようということになっています。IoTを「つなぐ」「見つける」「生み出す」という三層に分けています。IoT組込み技術は、積極的に取り組んでいかななくてはならないという部会などの意見を踏まえ、その分野を拡充していくことで新しい分野に対応できる人材を育てていきます。特定の分野やドメインに閉じるというような意図ではなく、一応ジェネ

ラルな方向性です。

松本 「つなぐ」「見つける」「生み出す」は良いキーワードですね。全体の体系がつかみやすいです。「見つける」というキーワードには、ビッグデータやAIの活用も含まれるのでしょうか。

井上 そうです。そのために、Hadoopを利用するような授業も今年から入っています。データマイニング系が重要だという意見が多く、スタートしました。

松本 組込み技術者とデータマイニングには、どのような接点があるのですか？

井上 組込みからのデータをクラウドに送り、そのデータを処理します。データマイニングを知らないといけません。そのための基礎勉強という位置付けです。最近ボードの性能が優秀になり、その中でもちょっとしたデータマイニングやAIの計算などができるようになってきています。例えば、ボード内で画像認識程度はできるのです。データマイニングを学んでも、ボードの開発を主戦場にすることは可能です。ただし、今はクラウドを通してバックボーンでデータを扱える時代ですから、両方の知識があることが非常に重要だと考えています。

再利用が欠かせないプログラミングは管理がキーに

松本 話題は変わりますが、先生が研究されている内容について教えてください。IPA/SECは、ソフトウェア・エンジニアリングへの取り組みを中心としてスタートした組織です。先生はソフトウェアエンジニアリングの中でもプログラム解析を中心に研究されていますが、そこを専門にしている理由は何でしょうか？

井上 私のバックグラウンドや研究者としての歴史に依存しています。子どもの頃はラジオを自分で作ったりして、高校時代にはプログラム電卓に憧れていました。簡単なプログラムが書けるHPの製品です。実際に触ったことはありませんでしたが、雑誌の記事で興味を持っていたのです。その後、その思いを胸に情報工学科に入学します。コンパイラやアセンブラを勉強しており、卒論のテーマはコンパイラの解析を選びました。そこで、プログラムを解析して、余分なところをそぎ落とすプログラム解析に目覚めました。研究としては地味なもので、縁の下で支えるような技術と言えます。ただ、開発環境全体で見るとコンパイラの最適化は重要です。例えば、これまで実行に数時間かかっていたものが、数十分で済むという効率化に貢献します。プログラムの分析は重要だと強く感じました。就職先は大阪大学で、ソフトウェア・エンジニアリングが専門。そのようなバックグラウンドがあったので、プログラムを改良して世の中の役に立つことができないう研究を始めました。インターネットのように爆発的に広がりはありませんでしたが、ニーズは大きい分野です。

松本 プログラムは動くだけで十分とは言えません。中身を解析してプログラムの動きを押さえておかないと、メンテナンスに苦労することになります。

井上 プログラムが「良い」のか「悪い」のかを評価する際にも、まずデータをきちんと取る必要があります。「バグばか

り出て困る」と言うだけでは足りないので、データから「どれだけ悪いか」を証明しなくてはなりません。証明できれば、改良するためにお金を出す必要性が分かるようになります。

松本 具体的に、プログラム解析の分野で力を入れている領域はありますか？

井上 プログラムは、同じものが何度も再利用されています。再利用の度合いが高いプログラムを定量的に評価して、レコメンデーションしたり、次のステップに役立てる研究をしています。

再利用が重要になっていますが、さほど関心を持たれていないという課題があります。もっと再利用を重視して過去の自社システムを利用したり、オープンソースを活用したり、更にそれを組み合わせたりするツールがあれば良いと思います。

松本 あるプログラムにバグや脆弱性があった場合、どこで再利用されているか分からないと大きな問題になりかねません。

井上 管理台帳のようなものが用意されていても、さほど信頼性が高くないというケースが多いのかもしれない。

松本 とくにオープンソースの場合には安易に利用してきちんと管理していないケースも多いと聞きます。

井上 現代では、使わざるを得ないでしょう。組込みはもちろん、エンタープライズ系でも当然のように利用します。最初は便利なのですが、メンテナンスが大変。ベースの部分が変更となったら、対応しなくてはなりません。また、ライセンスの扱いにも危険が伴います。条文にあった正しい使い方をしなくてはならず、「公開する」ことを条件とされていたら公開しなくてはライセンス違反になります。いずれにしても、今の環境ではオープンソースを使わないと開発ができないと考えて良いでしょう。

松本 1から開発するわけにはいきませんからね。様々なソフトウェア部品を組み合わせることで簡単にアプリケーションを組み上げられてしまう「超高速開発」というやり方も注目されています。ウェブアプリはその傾向が強いですね。ただしたくさんのものを組み合わせると、中に何が入っているのか作った人も分からなくなるという課題があります。

井上 ソフトウェアは使い回されるという前提で「ここに使われている」「この部分はあそこから来ている」とすぐに分かるようなシステムがあると良いですね。

松本 国レベルでできたほうが良いでしょうか。

井上 フランスで、Software Heritageというプロジェクトがスタートしています。これまでに様々なソフトウェアが開発され、作っては消えていきました。それらを「全部集める」と宣言したグループです。確かに面白いと思います。今はとりあえずLinuxのバージョンを集めているようです。

松本 ソフトウェアは資産でもあります。資産管理をしていかないとどこかで破綻する可能性もあるでしょう。せっかく優れた資産があっても、誰にも使われず、知られていなかったらもったいないですね。

井上 企業内でも、作りっぱなしのソフトウェアはたくさんあります。また、今一生懸命作っているものが過去に誰かが作ったものと同じ、ということもあります。しっかり管理できれば、そのような非効率なことが避けられます。社外の資産も参照できればとても便利です。フランス人にはそ

ういうロマンがあるのかもしれませんが。2~3社に話してみましたが、あまり興味を持ってもらえませんでした。壮大過ぎる話に聞こえたのかもしれませんが。

松本 確かに、いきなりは難しいかも知れませんが、徐々に積み上げていけば、資産として使えるようになっていくのではないのでしょうか。

井上 Software Heritageにはロマンを感じますが、テクニカルにどんな実装をするのかは知りたいところです。とりあえず放り込むだけでは有効に利用できません。検索はできるかもしれませんが、整理したり分類したりするのは一筋縄ではいかないでしょう。

松本 企業はバックログで苦勞しているの、良い試みだと思えます。現在、COBOLが分かる技術者は少なくなっているの、COBOLで書かれたプログラムがどう動くかしているのか分かり難くなりつつあります。それでも、どこかで使われているんです。プログラムを管理するシステムが必要はなはず。作りっぱなしで動いているものをなんとかしなくてはいけない時期なのかもしれません。

井上 Software Heritageも、同様の発想なのかもしれません。大規模システムの初期メンバなどは現場から退いてしまつて、分かる人がいなくなっているのしょう。

産官学連携には“雑談の場”が功を奏す

松本 IPA/SECも産官学連携をテーマにして取り組みをしています。産業界と大学の連携で、お互いの距離を縮めていくには何が必要なのでしょうか。

井上 劇的な方法はすぐに思いつきませんが、お互いに敷居を低くしていく必要はあると思えます。過度な期待をせず、雑談できるような環境が大事です。大学の人も、産業界に出ていく必要があります。ソフトウェア工学を研究している人にとって、ソフトウェア自体にあまり意味はないはず。動いて、使われて初めて意味を成すのです。役立てている人の意見を聞き、ソフトウェアについてあらためて考える必要があるでしょう。机上だけで論ずるより、現場に出ていき使っている人たちのフィードバックをもとに研究していく必要があると思えます。企業の人には「大学はとっつきにくいかもしれないが、雑談から始めて欲しい」と話しています。雑談の最中に「ここに問題がある」と言われれば、「既に研究している」「それはちょっと難しい」と気軽に返せます。お互いの敷居を低くして、気楽に話せる場を持ちたいと考えています。そういう意味で、ESIPは役に立っているのではないかと考えています。

松本 直接話す機会がなかなかないのかもしれないですね。

井上 大学の人もなかなか忙しく「論文を書け」と言われています。企業の人と話をしても、すぐに論文に結び付くとは思えないのかもしれない。ただ、長期レンジで見ると、しっかりとニーズに向かった研究につながります。

松本 ニーズに基づいた研究であれば、現場ですぐに実践につながられます。

井上 卒業生は実際そのように活躍しています。

松本 “雑談の場”を作らないといけないのですね。

井上 積極的に大学の人も出ていかななくてはならないし、企業の人にも入ってきてもらいたいと思えます。解決策を

即座に期待するより、お互いに1時間の雑談を楽しむことが良い結果に結び付くような気がします。いろんな人と話していただくことが大事ではないでしょうか。

松本 IPA/SECもソフトウェア工学に取り組んでいます、私共に対する要望や期待などがあれば教えてください。

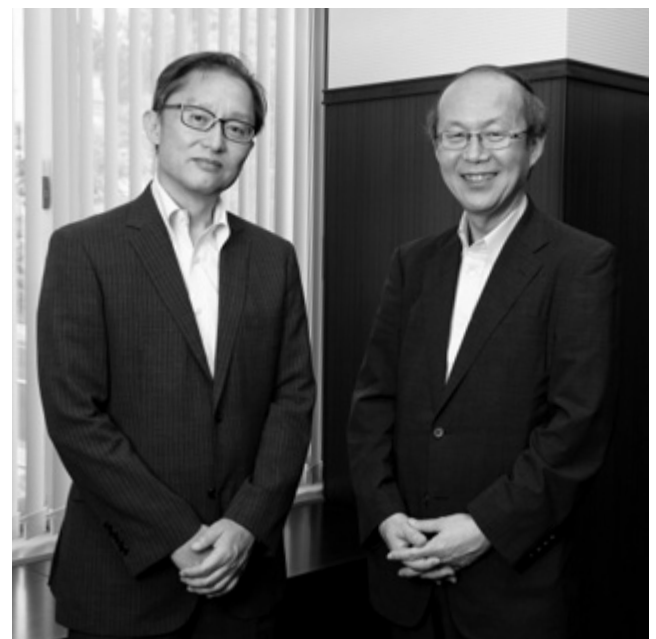
井上 ソフトウェアの大元締めとして様々な施策を実施していらっしゃいます。これからも中心的な活動をしていただきたいです。今、ソフトウェアに風が吹いていると思っています。安倍首相が「小学校からプログラミング教育」と言っていたり、文科省でも情報技術が中核として重要だと言っていたりします。IPA/SECはその中心として、日本のソフトウェア業界を引っ張っていただきたい。日本のソフトウェアが世界に出ていくような教育を進めたり、国際交流の場を設けたりしていただけると嬉しいです。

ソフトウェア産業は輸出できないと言われていました。今後は、日本から外に出ていくような工夫が必要だと思っています。日本からのオープンソースで世界に貢献しているものが少ないと言われていたため、IPA/SECの支援で共同開発したり、コントリビューターに支援したり、日本人が世界のコミュニティに入っていきための後押しをしていただきたいですね。

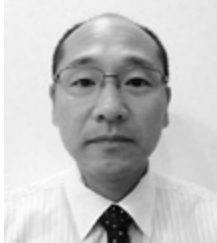
松本 国際化の話は確かに必要ですね。大学や企業だけでは難しいのかもしれない。

井上 企業は戦略的にオープンソースに投資しています。アメリカの企業などはそうです。日本の企業もメリットを感じられれば進めていけると思えます。官のほうで人材育成の一環として、オープンソースの共同開発に投資しても良いのではないのでしょうか。すぐにメリットが見えないかもしれませんが、人材教育の一環なら進めやすいと思えます。

松本 国際的に通用する人材を育てるのは非常に重要ですね。日本産のプログラミング言語「Ruby」を国際的にしようIPAも力を入れた時期がありましたが、これに続く国際化の支援活動をぜひ考えていきたいところです。今日は貴重なお話をありがとうございました。



プロセス改善技術者育成コースの設計と実装

久野 倫義^{※1}中島 毅^{※2}芝田 晃^{※3}近藤 聖久^{※1}小笠原 公一^{※4}

ソフトウェア開発組織の改善の核となるプロセス改善技術者の重要性が指摘されている。しかし、同技術者の育成コースは、スキル要求が多岐にわたるため、スキル達成目標を定めるににくい。さらに、一事業所ではコースの指導者も育成対象者も少数となるため、集合研修形態での育成コースの実現が難しい。三菱電機グループはこうした課題を解決するプロセス改善技術者育成コースを開発した。同コースは、現場の改善技術者が出会う困難を解決するためのスキル習得に目標を絞ることで、少ない開講期間で実用的に実現できる。また、企業グループ全体を母体として育成対象者と指導者を選出することで、集合形式のコースで効率良く育成できる。本論文では、本コースの設計・実装について述べ、受講後のアンケート調査などにより、全受講者の67.2%が改善推進者として成長を実感し、受講が成長に貢献したという回答が89.8%であることで、本コースが有効であると評価した。

Design and Implementation of Training Course for Software Process Improvement Engineers

Noriyoshi Kuno^{※1} Tsuyoshi Nakajima^{※2} Akira Shibata^{※3}
Kiyohisa Kondo^{※1} Koichi Ogasawara^{※4}

It is widely recognized that software process improvement (SPI) engineers need to be trained for software engineering groups to perform quality SPI activities. However, such engineers are required to have a wide range of skills, and therefore it is difficult to properly determine the scope and goal of training courses. In addition, classroom training courses are difficult to implement because it is unlikely that there will be sufficient numbers of trainees and trainers in individual business offices. To solve such problems, Mitsubishi Electric Corporation and its group companies have developed a training course for SPI engineers that provides practical contents in a short period by focusing on skills for solving real problems that SPI engineers often encounter in the field. Also, the course can be efficiently implemented in a collective form by extending its base for selecting trainers and trainees to the group companies. This paper describes the design and implementation of the course, and proves its effectiveness with a questionnaire survey answered after taking the course, from which results show that 67.7% of the trainees find themselves much more skilled than before., and 89.8% believe it due to taking this course.

※1 三菱電機株式会社 設計システム技術センター

※2 芝浦工業大学 工学部 情報工学科

※3 三菱電機株式会社 インフォメーションシステム事業推進本部 インフォメーションシステム業務部

※4 三菱電機株式会社 生産システム本部 生産技術部

1 はじめに

近年、多くのソフトウェア組織がプロセス改善に取り組んでおり^{1) 2)}、その改善の核となる改善推進グループ(SEPG: Software Engineering Process Group)の重要性が指摘されている³⁾。しかし、SEPGを構成するプロセス改善技術者(以下SEPG要員と呼ぶ)は、一般に、高い問題解決力と、ある程度の開発経験とパーソナリティを必要とするため、その確保が難しい。そのため、質の高いSEPG要員を効率よく育成することはプロセス改善を進める上で重要な課題となっている。

SEPGの業務は重要かつ多岐にわたる。Humphrey⁴⁾は、SEPGの役割を以下の6つと定義している。①プロセスの標準を確立する、②プロセスのデータベースを維持する、③技術導入の中心として支援する、④鍵となるプロセス教育を提供する、⑤プロジェクトのコンサルテーションを提供する、⑥定期的なアセスメントと状況報告をする。

前述の①～⑥を実施できるSEPG要員は、プロセスにかかわる技術だけでなく、開発技術や管理技術、パーソナル技術などが広範囲に必要とされる。そのため、SEPG要員の育成コースを作る場合、SEPG要員に求められる幅広いスキルの範囲に対して、どう達成目標を定めるかが重要となる。

さらに、事業所内でSEPG要員の育成コースを作ろうと考えても、ソフトウェア開発者の3～5%の育成が必要と言われるSEPG要員数⁴⁾では、各事業所の育成対象者が少なく、また指導する講師も確保できないという運営上の課題もある。

こうしたSEPG要員育成の課題を解決するために、三菱電機グループはプロセス改善技術者育成コース(以下SEPG要員育成コースと呼ぶ)を企画・設計した。同コースは、現場のSEPG要員が出会う困難を解決できることに絞ってスキル達成目標を設定する。これによって、座学と演習を、短い開講期間により深い内容で実現できるようになる。また、同コースは、企業グループ全体を母体として育成対象者の募集と指導者の選出を行うことによって、効率良くコース運営を行うことができる。

同コースは2011年度から7回実施し約90名のSEPG要員を育成した。本論文では、コース開発の背景、設計上の課題と解決策、その実施結果の評価について述べる。

2 SEPG要員育成コースの検討過程

2.1 SEPG要員育成が必要となった背景

三菱電機グループは、2004年度より全社的な改善の枠組みを作り、ソフトウェア改善活動を実施してきている。三菱電機の各事業所とソフトウェア開発を請け負う各関係会社(以下単的に「事業所」と呼ぶ)が、年度ベースで事業所全体のソフトウェア開発の実態把握と改善施策の立案を行う。そして各事業所の代表者の合議体が、グループ全体のソフトウェア施策を合意実施していく体制を

作っている。

2007年度、各事業所の自律的改善を進めるため、プロセス改善推進体制を強化することが課題とされ、事業所の改善推進者とコーポレートのSEPGが集まりソフトウェアプロセス改善に関する課題解決を行う場としてEP-WG(Engineering Process Working Group)が設立された。このWGの中でISO/IEC 15504 Part 5⁵⁾のプロセス記述の解釈や、改善情報の交換を進め、WGに参加するSEPG要員は事業所のコア人材として育っていった。

次に、自律的なプロセス改善を促進するために、事業所のSEPG要員をソフトウェア技術者に対して一定比率にする目標を立て合意した。この目標は2年でほぼ達成された。しかし、SEPG要員は、一定の開発経験を持つことを条件に集められたものの、本人の業務に対する動機付けや、改善を進める上でのスキルが不足していたので、せっかく確保した要員の活用が十分できていないことが次の問題となった。そのため、現有及び将来のSEPG要員の質向上を次なる課題として設定した。

2009年度EP-WGにおいて、SEPG要員の現状分析と、それに基づく上記課題の解決策として、SEPG要員育成コースを提案し、同コースに対する要求事項の検討を行った。

2.2 SEPG要員の現状と育成ポイント

SEPG要員に対する教育は、ISO/IEC 15504 Part 5やCMMI⁶⁾で示されるプロセスの理解など、抽象的な知識を詰め込む形式に陥りやすい。そのため、教育講座の受講で獲得した知識を直ぐに開発現場の問題解決に利用することが困難という課題があった。そのため、SEPG要員育成コースの目標は、SEPG要員が現場で直面する問題を解決できることにおいた。

EP-WGにおいて、SEPG要員が現場で直面する問題を抽出し、以下の4つの課題に集約した。

課題1：役割の不明確さ

事業所の中で、SEPGの役割と責務が明確に定義されていないことが多い。その場合、真の改善を推進していく役割と能力が見過ごされがちとなる。また、個々のSEPG要員も、限られたリソースの中で活動範囲と優先順位を個別に設定して行動するようになり、組織全体の取り組みとして整合性を欠いた改善活動となってしまふ。

課題2：経営者の理解、現場の賛同の欠如

ハードウェアの生産ラインでは、ライン効率や歩留まり率など改善指標と経営数値の関係が比較的明確なため組織的な合意の下に改善が進みやすい。しかし、ソフトウェア開発では、価値・規模に相当する生産量の定義に決定打がなく、改善効果の定量化を軽視する傾向がある。それが経営者や現場の理解を得られない一因となっている。ライン管理者以下も同じで、改善を導入する際の初期コストと運用コストを超える改善効果を示すことができなければ、改善活動に賛同を得ることは難しい。

課題3：プロセス理解のない問題分析と解決

プロセス改善を図る際、CMMIやISO/IEC 15504などの、

プロセスモデルという実績ある道具がある。しかし、多くのSEPG要員はこれらの道具を知っていても使いこなせていない。そして、自職場の問題に対して、現状のプロセスの分析をしないか、分析しても始めから答えありきで、自分の得意分野と興味に偏った技術的手段で解決を図ろうとしがちである。こうした解決策は、概して問題に対して的外れなものであるか、適用上の考慮が欠けたものとなるため、失敗の確率が高い。

課題4：PDCAのない改善活動

プロセス改善は、対象となるプロジェクトに適用してこそ改善効果や課題が明確になる。改善活動は、対象プロジェクト中心に準備と適用を計画し、適切な時期にその進捗と効果を確認し、必要なら再計画しながら進むべきであり、そうでない場合、対象プロジェクトの意図とスケジュールに合わず、改善活動が無理や無駄を生み中途半端になりがちである。また、SEPG要員は、通常複数の業務を並行して遂行しているので、計画をしっかり立てておかないと、忙しさに紛れてより困難かつ重要な改善活動に手が回らない場合が多くなる。

2.3 SEPG要員育成コースに対する要求事項

EP-WGが設定したSEPG要員育成コースの育成目標は、「ソフトウェア開発現場の現状を正しく認識でき、組織の事業目標を実現する効果的で実行可能な改善計画書を書き、説明できる」こととした。これを行うためには、前節で述べた課題を解決する能力がSEPG要員に求められる。4つの課題に対応して、受講生の達成レベルに対する要求事項を以下のように設定した。

- ① SEPGの役割・意義を知り動機付けを持つ【課題1】
- ② プロセスモデルをベースに改善することの重要性を理解できる【課題3】
- ③ 開発プロセスガイドを使いプロセス点検ができる【課題3】
- ④ 改善のための問題解決手順を実践し、実行可能な改善計画を作成できる【課題4】
- ⑤ プロセス改善を経営の問題と考え、経済的合理性を訴えることができる【課題2】

ここで、開発プロセスガイドは、ISO/IEC 15504の各プロセスを、三菱電機グループの組織がどう実装すべきかをガイドしたもの、改善のための問題解決手順は、プロセスモデルベースの改善を実施するための手順である。これらはEP-WGで開発済みのコンテンツである。

上記の達成レベルに加えて、育成対象者を目的に沿うものとするため事業所の推薦者として、年間各事業所1名以上育成できること、業務への影響を少なくするためコース日数は5日以内、連続した日数3日以内であることなどを要求事項とした。表1にEP-WG提言としてまとめた育成コースに対する要求事項を示す。

表1 育成コースに対する要求事項

Table 1 Requirements for education course for SPI engineers

項目	要求事項
対象者	SEPG実務者、及び組織的改善をプロジェクトサイドから推進すべきリーダークラス 各事業所の推薦者に限定
育成数	各事業所1名/年以上
達成レベル	ソフトウェア開発現場の現状を正しく認識でき、組織の事業目標を実現する効果的で実行可能な改善計画書を書き、説明できる。以下が要求事項: ① SEPGの役割・意義を知り動機付けを持つ ② プロセスモデルをベースに改善することの重要性を理解できる ③ 開発プロセスガイドを使いプロセス点検ができる ④ 改善のための問題解決手順を実践し、実行可能な改善計画を作成できる ⑤ プロセス改善を経営の問題と考え、経済的合理性を訴えることができる
日数	5日以内(連続は3日以内)

3 コースの設計

3.1 コース構成

EP-WGの要求事項を受け、本社生産技術部は2011年度SEPG育成コースを設計・実施することに合意した。図1に設計・実施したコーススケジュールを示す。受講日数は要求事項を満足するように、全体で4日間、連続は3日以内となっている。

図1において、単元1～5は座学、WS1-3はワークショップの形式である。3.2で座学、3.3でワークショップの構成を示す。

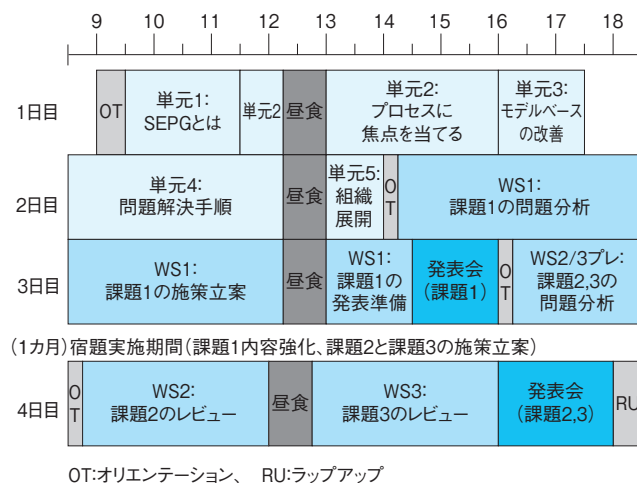


図1 SEPG育成コースのスケジュール

Figure 1 Time table of education course for SPI engineers

3.2 座学の構成

座学は、SEPG業務に対する正しい認識と基礎的な知識の習得、及びワークショップの導入の位置づけを持っている。表2に、各座学の達成目標と表1の要求事項への

対応、及び各単元の講義項目と内容を示す。

座学は、グループ演習を組み合わせ、知識の定着を図るとともに、事業所間の状況や課題を情報共有する機会としている。例えば、表2の1-4のG演習1では、SEPG活動チェックリストを用いて、自らの業務を振り返る演習を実施している。ここで、SEPG活動チェックリストとは、IPA/SECの改善のプロセス³⁾をベースに、EP-WGが三菱電

機グループの事業所におけるSEPGの業務範囲として定めたものをチェックリストにした表である(図2)。

表2の1-4において、受講者に事業所全体の活動と自らの活動内容をこの表にマッピングする事前課題を与え、座学による解説の後、その題材を素に、自部門のSEPGの活動内容の過不足を議論させる演習を実施する。図2の例で×は自部門として弱みとして認識された部分である。

表2 座学の単元と達成目標、各単元の項目に対する内容

Table 2 Classroom lecture units and their goals/contents.

単元/講義項目	達成目標(→表1の要求事項への対応)/内容
1 SEPGとは	SEPGの役割・意義を知り改善活動の流れを理解する(①)
1 SEPGがなぜ必要か	<ul style="list-style-type: none"> 事業課題から展開された組織QCD(品質/コスト/工程)の向上が目標 標準プロセス・インフラ確立・維持・向上→組織能力の向上
2 プロセス改善のPDCA	SEPGはどんな流れの中で活動するのか <ul style="list-style-type: none"> 年度計画/プロジェクトベースのPDCAサイクル SQA (Software Quality Assurance)との関係
3 プロセス改善のプロセス	<ul style="list-style-type: none"> SEPG活動内容と具体的活動項目(図2)
4 改善推進体制と改善の進め方	<ul style="list-style-type: none"> SEPG体制のタイプ別特徴と留意点 IPA/SEC プロセス改善虎の巻Q&A紹介³⁾ [G演習1]自職場SEPG機能の過不足をチェック
2 プロセスに焦点をあてる	プロセスの重要性とプロセスの構成要素を理解する(②)
1 プロセスに注目する理由	<ul style="list-style-type: none"> 仕事の段取り(プロジェクト、個人)と制御(進捗、品質) 標準化→コミュニケーションの向上、ノウハウの蓄積・再利用、教育の容易化、見える化、改善・評価が進む
2 プロセスを制御する	<ul style="list-style-type: none"> プロセスアプローチ:高品質の製品は良いプロセスから プロセス品質を制御することでプロダクト品質を達成する
3 プロセスモデルとその意義	<ul style="list-style-type: none"> モデルを活用して組織(あるべき)プロセスを定義する 世界の企業間で共通の仕事のやり取りができる 国際標準の動向(ISO/IEC 15288, 12207, 15504, CMMI)
4 V字モデルの意義	<ul style="list-style-type: none"> 設計と検証の対応関係をとる 品質要求を含む要求の定義と、その追跡性の確保が重要
5 ライフサイクルモデル	<ul style="list-style-type: none"> ウォーターフォール、インクリメンタル、エボリューションモデルの意味、強み弱み ライフサイクルの組み立て
6 プロセスの要素	<ul style="list-style-type: none"> プロセスチャートを使ったプロセスの構成要素の説明 [G演習2][契約見積もり]のプロセスチャートを作成
3 モデルベースの改善	モデルベースの改善の流れを理解する(②)
1 モデルをものさしにする	<ul style="list-style-type: none"> プロセスモデルとの差異で、改善機会を抽出。改善機会の優先度は、事業課題の達成への貢献で判断
2 モデル体系	<ul style="list-style-type: none"> ISO/IEC 15504のモデル体系
3 QMS整備との関係	<ul style="list-style-type: none"> モデルベースの改善はQMS (Quality Management System)整備と矛盾する活動ではない
4 改善のアプローチ	<ul style="list-style-type: none"> 改善活動でプロセスモデルを活用する
4 問題解決手順	ワークショップ実施に必要な技術・技法として、問題解決手順の流れと考え方を理解する(③, ④, ⑤)
1 事業目標、改善スコープ・目標の設定	<ul style="list-style-type: none"> 事業目標から改善目標へのブレークダウン 目標状態(To)と現状(From)とのギャップの認識 データからの傾向分析
2 プロセス点検	<ul style="list-style-type: none"> 開発プロセスガイドに沿った主要プロセスのプロセス成果と基本プラクティス、レベル2, 3の共通プラクティスの理解 プロセス点検の実施方法
3 対策の立案	<ul style="list-style-type: none"> 取り組み課題の施策化 投資対効果に基づく優先順位付け、実施施策の選択 計画(担当、スケジュール、監視・制御)、現場の巻き込み方
4 改善の定量化	<ul style="list-style-type: none"> 改善を進めるための指標の重要性 改善の活動、効果のメトリクスの考え方 指標の設定・評価方法
5 組織展開	改善の組織展開に向けた具体的なノウハウを共有する(→①, ④)
1 改善の定着	<ul style="list-style-type: none"> 組織展開の重要性の意識づけ 具体的展開のノウハウなど(教育、体制、評価方法など) [G演習3]プロセス改善推進上の課題と取り組みの共有

改善サイクル	活動内容	具体的活動項目	SEPG実施部門			コメント
			自部門	他部門		
			事業所SEPG	部門SEPG	PRJリーダー	
②計画	活動施策の具体化	実行計画の策定	○	○		
	具体的改善項目の特定・選定	ニーズの把握	×			ニーズの把握を明示的に行っていない
		シーズの把握(適用新技術の識別・評価)	○			
		個別施策化・見積り評価	○	○		
	改善施策の計画	WBS、費用、体制、スケジュールなどの計画	○	○	○	
	①評価・導入施策	プロセス/ツール/技法評価・導入計画の策定	プロセス	ツール技法	対象	プロセスに偏っており、技術的な改革に関与できていない
	②定着施策	プロセス/ツール/技法定着計画の策定	プロセス	ツール技法	対象	
	③プロセス診断施策	プロセス診断の計画	○	支援	対象	OK
	④教育施策	組織要員能力(含以前の教育効果)の把握	×			現状把握の活動がない
		教育実施計画の策定	×	○	支援	教育は部門任せ
	⑤外注施策	外注施策の策定	○	支援	支援	OK
	改善施策の評価方法の決定	施策評価方法確立	×			評価が弱い
施策キックオフ(関係者への説明・周知)	改善施策の説明会(個別に実施)	○	○	参加	OK	

図2 SEPG活動チェックリスト(計画段階部分) Figure 2 Checklist for SEPG's activities (its planning part)

3.3 ワークショップの構成

(1) モデルベースのプロセス改善と問題解決手順

モデルベースの改善アプローチの有効性は一般によく理解されており、SEPG育成の取り組み事例^{7) 8)}でもCMMIなどのプロセスモデル習得やアセッサ育成の内容をプログラムに組み込んでいることが多い。

しかし、CMMIやISO/IEC 15504などは頭で理解してもそのままでは記述の抽象度が高く応用が難しく、逆にアセッサ育成コースはアセスメントの実施に重点をおくため、抽出した弱みや改善機会を、有効で実施可能な施策に落とし込むための部分が弱い。そのため、これらの講座だけではSEPG業務を遂行するための即戦力養成の目的に十分ではない。

そこで、以下に示すような問題解決の手順を定義し、その手順の中でプロセスモデルベースの改善を、②の(ア)(イ)及び③の(ア)として位置づけることで、適用を容易にする工夫を入れた。

① 問題分析：問題を設定する

- (ア)改善テーマを記述し、ビジネスドライバを設定する
- (イ)問題と考える現状QCD状況を把握する
- (ウ)狙いとすべき目標QCDを設定する
- (エ)目標と現状のギャップを問題として認識する

② 問題分析：課題を抽出する

- (ア)データに基づき問題となるプロセスとその悪い状態を洗い出す
- (イ)問題プロセスを、開発プロセスガイドに沿って簡易診断することにより、問題の原因を深掘りし、課題として抽出する
- (ウ)重要な課題に狙いを定める

③ 施策立案

- (ア)課題を、プロセスの枠組み(組織・プロジェクトPDCA、標準化、インフラ、教育、評価・検証)に沿って施策化する
- (イ)改善対象プロジェクトへの適用・評価、年度の改善サイクルを考慮して、改善のステージを設定し、施策をスケジュールリングする
- (ウ)改善の活動目標と効果目標を設定する
- (エ)改善活動のコスト対効果を計算する

④ 改善をステークホルダに説明する

(2) 3つのワークショップとその実装

受講者が上記の問題解決手順を確実に習得できるようにするため、WS1-3の3つのワークショップを設定した。図3にワークショップにおけるチームの構成と作業の流れを示す。

ワークショップにおいて以下の人員構成をとる。

- 異なる事業所の受講者3人を1チームにする。
- 指導者として、プロセス診断の専門家(リードアセッサクラス)を各チームに1名割り当て、張り付きで指導・助言を行う。

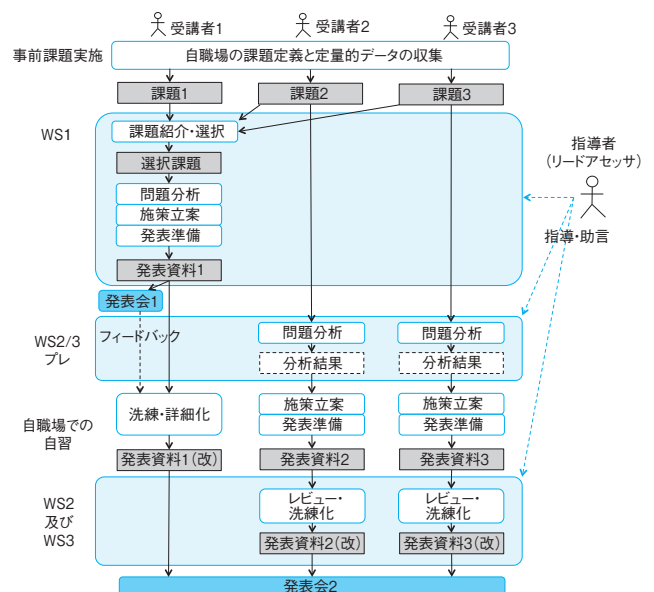


図3 ワークショップにおけるチーム構成と作業の流れ Figure 3 Team organization and activity flow in Workshop

チームごとに、問題解決手順を適用して課題を解くサイクルを、3人の自職場の課題に対してそれぞれ実施する(WS1-3)。図3に示す流れを以下に説明する。

- 事前課題実施：受講者は、自職場の課題定義と定量的データの収集を行ってくる。
- WS1：課題紹介を行い、比較的好くまとまった課題か

つ定量的データのあるものを選ぶ。その課題を受講者3人と指導者により問題分析、施策立案、発表準備を行う。これにより、問題分析のやり方や施策を立てる手順やノウハウを共有する。指導者は、特にプロセス点検を重点的に指導する。

- **発表会1**：課題1に対する改善施策について受講者1が発表する。発表は、組織関係者へのプレゼンを想定して行う。発表に対して指導陣が質問・コメントする。
- **WS2/3プレ**：発表会の後、ほかの受講者の課題2,3について、指導者が主導してプロセス点検を中心に問題分析を行い、問題解決の方針を立てる。
- **自職場での自習**：受講者は、約1カ月間自職場で課題に取り組み、発表資料の形式にまとめる。受講者1は課題1の改善施策について洗練・詳細化し、受講者2,3はWS2/3の指導結果に基づきおのおのの課題について改善施策を立案する。
- **WS2及びWS3**：受講者が事前に実施してきた課題2,3に関する改善施策を、指導者とほかの受講者でレビューし洗練化する。
- **発表会2**：課題1の発表資料(改)及び課題2,3の発表資料について発表し、指導陣が質問・コメントする。

(3) 狙いとする効果

ワークショップを(2)のように構成した狙いは、以下の3点である。【】内に3点の要求事項への対応も示す。

- 自職場の課題を解くことにより、各事業所が取り組むべき改善活動を計画する(あるいは現在やっている改善活動を見直す)機会とする【③④】
- 他事業所の課題を共同して解くことで、問題解決手順を繰り返し実施することができ、同手順の習得と定着を促進する【④】
- 組織関係者に改善施策を説明する想定で、受講者に発表させ指導することで、改善施策の必要性と経済性を論理立てて説明する方法を習得する【⑤】

4 コースの実施と評価

4.1 コース実施状況

SEPG要員育成コースは、年2回のペースで実施してきており、毎年30名を育成することが可能である。実際、本コースは、2011年度後半から年2回、2014年度末現在で7回実施し、事業所の技術者約90名の育成を完了した。

4.2 受講者アンケート調査結果とその分析

SEPG要員育成コース開始後2014年10月に第6回までの受講者に対して、アンケートで追跡調査を行い58名より回答を得た。以下にその分析結果を示す。

(1) 理解度と役立ち度

受講時点あるいはアンケート時点で、全受講者のうち、SEPG要員(事業所全体、製造部門全体、開発チーム内SEPG担当で兼務者も含む)の割合は70.7%であった。事業所は、おおむねコース開催の意図通りに受講者を派遣している。

受講者のコース各パートに対する理解度と役立ち度を、それぞれ図4と図5に示す。

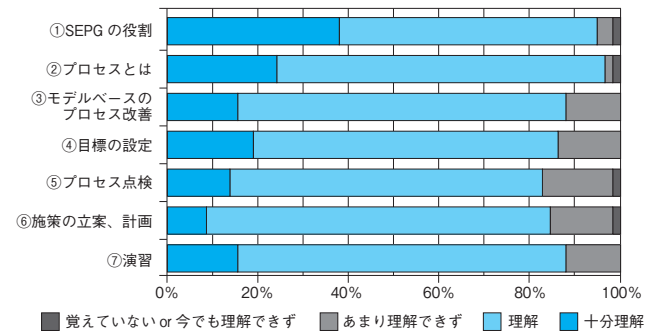


図4 コースの理解度
Figure 4 Degree of participants' understanding

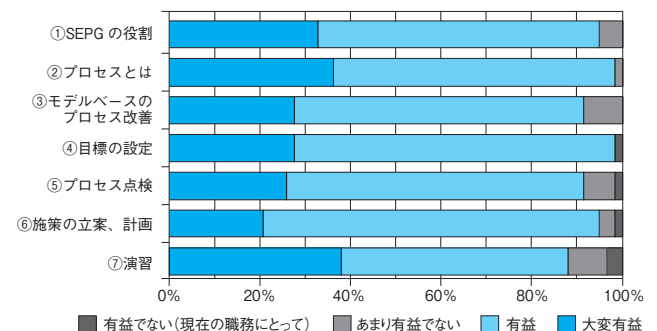


図5 コースの役立ち度
Figure 5 Participants' evaluation on value of the course

質問は4択であるが、理解度と役立ち度は共に、全単元で肯定的な反応が80%を超える結果となった。

(2) 成長の実感とコースの貢献度

図6に、受講者のSEPG要員としての成長の実感と、成長を実感していると回答した場合のコースの貢献の程度を示す。全受講者の67.2%が改善推進者としての成長を実感していると回答し、実感していると回答した人は、本コースの受講が「大変貢献した」「まあ貢献した」合わせて89.8%であった。受講生の実感としてのコースの有効性が分かる。

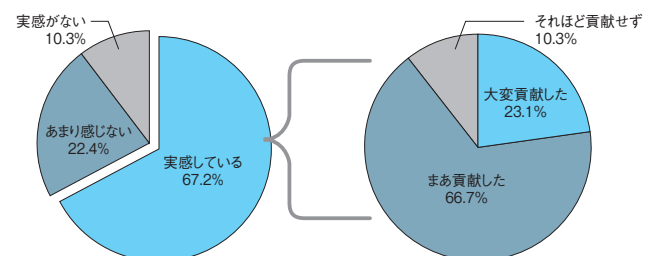


図6 受講者の成長の実感とコースの貢献度
Figure 6 Participants' self-evaluation on their career development and contribution of this course

(3) 経験差による層別分析

経験で層別したコース効果を評価する。図7に、SEPG要員経験で層別した受講者数(左軸)と、経験別の成長の自覚(「自覚あり」と回答した割合:右軸)、本コースにおいて自ら提案した改善施策の実施状況(「おおむね実施できた」と回答した割合:右軸)、を示す。

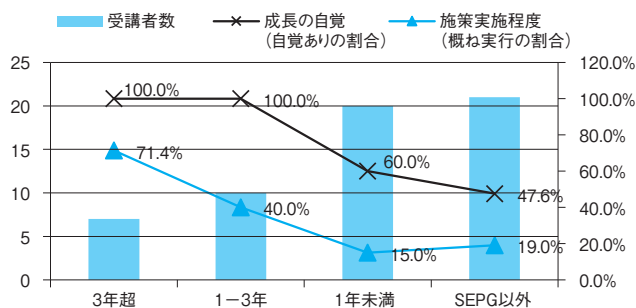


図7 SEPG要員経験の層別分析

Figure 7 Stratified analysis based on participants' experience

SEPG要員としての経験年数が、1年以上の人は17名、1年未満の人は20名受講している。1年以上の人では全員が成長を実感しているのに対して、1年未満の人は60%になっており効果が薄かったことが分かる。改善施策の実施についても、経験の長い人ほど実施できていることが分かる。1年未満のSEPG要員経験者にこうした傾向が現れるのは、改善施策を自ら主導できない立場にいて、本コースで得た知識と技法を利用する機会が少ないことが影響しているのではないかと考える。

(4) 自由質問に関する分析

コースに対して、何が役に立ったのかという自由質問に対して46件の回答を得ており、それを分析した。コースの要求事項①～⑤に対応した肯定的なコメントの言及数と、その代表的なコメントを表3に示す。各要求事項に対する満足が多数得られていることから、コースは要求事項を満たす所要の効果を上げていることが推測できる。

表3 要求事項に対応したコメント数と代表的なコメント
Table 3 Number of comments and typical comments corresponding to requirements for this course

要求事項	意見数	主なコメント
① SEPG役割	8	役割を体系的に知ることができた。自組織で何が足りないかがよく分かった。
② プロセスとは	11	プロセスのあるべき姿が分かった。
③ プロセス点検	8	プロセス点検で役に立っている。
④ 問題解決手順	14	改善施策を事業目標からブレークダウンして考えられるようになった。講義を聞いた後に演習で自ら改善の筋道を考えたことが最も効果的だった。
⑤ 事業目的展開	5	経営層の巻き込みと、目標と現実の定量的なギャップ認識を意識できるようになった。

4.3 上長に対するアンケート調査結果とその分析

受講者を送り出した事業所の派遣元上長に対してもアンケートを収集し分析を行った。質問は、講座が受講者の育成に有効であったかを問うもので、4つの選択肢 ①大変有効であった、②ある程度有効であった、③あまり有効ではなかった、④ほとんど有効ではなかった、から選ぶ形式である。

図8は、アンケート結果である。

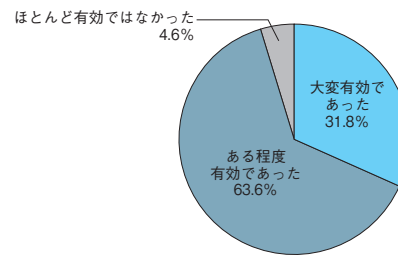


図8 受講者を派遣した上長による講座の評価

Figure 8 Managers' evaluation on value of the course

全派遣元上長の95.4%が本コースを「大変有効であった」「ある程度有効であった」と評価しており、客観的にもコースの有効性を評価できた。

4.4 コースの狙いに対する総合評価

コースは、表1に示す要件の目的であるSEPG要員の質の向上に関しては、4.2に示したように、受講者の67.2%が自身のSEPGとしての成長を実感し、そのうち89.8%がコースの貢献を認めていることから、一定の効果が得られていると評価している。

また4.3に示したように、事業所の派遣元上長の評価も95.4%が効果を認めており、実際に役立っていることが分かる。事業所代表者会議からも、同コースに対してSEPG育成に高い貢献があるとの評価を受けている。

受講者と指導者の確保についても、毎年始めに行うコース受講要望の調査で事業所のニーズが高い状態が続いているため年2回のペースの開催を維持している。コースの指導者は、ISO/IEC 15504やCMMIのリードアセッサらによる5名体制を維持することができている。

コースに対する表1の要求事項①～⑤に対応した各パートの理解度と役立ち度がともに80%を超えるとともに、表3に示したように、要求事項に直接言及する多数のコメントを得ることもできていることから、満足する結果を得たものと評価している。

4.5 コースの課題

(1) 経験の浅い要員の自職場における実践機会の確保

4.2 (3)に示した経験差による層別分析から、経験の少ない要員は、コースで得た知識と技法の利用する機会が少なく、そのため育成の効果が比較的低めに出ていると結論づけた。この問題の解決は重要であり、今後受講者の上司と連携し、コース実施後の改善施策提案の実施に関するフォローをきめ細かく行うことなどを検討している。

(2) ワークショップ時間の延長

WS2とWS3の受講者は自職場での自習が主体となる。自習内容の質が低い場合、半日のレビューだけでは良い改善施策にならないケースが多い。この問題は、例えばコース日程を1日増やしWS2とWS3に時間をかけることで改善できるが、指導者と受講者の負担が増えることになるため、投資対効果の議論が必要である。

(3) 指導者の計画的育成

コースを実施するためには5名以上のリードアセッサクラスの指導者が必要となる。そのため、指導者クラスの育成も計画的に行っていくことが必要である。

5 関連研究

5.1 SEPG要員に必要とされるスキル

Humphreyは、SEPG要員に対して期待することとして以下の4点を挙げている。①プロセスの変更と指導に対する熱意、②技術的・政治的に問題を理解し効果的な解決を実現する能力、③現場の技術者からの尊敬、④経営者の支持。熱意やパーソナルスキルを重視している。

小川ら⁹⁾は、IPA/SECのプロセス改善推進者の体系的な育成支援のための取り組みを紹介している。高度プロセス改善推進者やアセッサに至るキャリアパスを提供、アセッサトレーニングとしてロールプレイを取り入れた実証実験を行っている。彼らは、プロセス改善推進者に必要な能力を①基盤となる人間力、②モデルの活用・応用力、③業務に関する新技術の習得力であると結論づけている。小川らも、プロセスモデルの活用技術以外はパーソナルスキルを重要視したものになっている。

ETSS (Embedded Technology Skill Standards)¹⁰⁾はSEPG要員を、開発プロセス改善スペシャリスト「開発プロセスとその実施状況をアセスメントし、改善の推進を担当する専門技術者」と定義し、技術要素、開発技術、管理技術、パーソナルスキルの各スキル要素に対する全般的な要求事項をまとめている。広くSEPGのキャリアを考える場合有用な基準を提供している。

福原⁷⁾は、SEPG要員に必要とされる能力を、①組織のあるべき姿を描く、②規格／モデルを活用し現状を把握する、③あるべき姿と現状のギャップを課題として特定する、④有効かつ現実的な改善策を提示する、の4つとしている。これは我々のコースに対する5つの要求事項のうち①～④にほぼ対応している。

5.2 SEPG要員の育成方法

小笠原ら¹⁾は、東芝における全社的なソフトウェアプロセス改善活動の実践において、SPIを定着させるための仕組みの一部としてSEPGの教育を位置づけ、プロセス改善の基礎知識としてCMMIとIDEAL¹¹⁾の習得を主な目的としたSEPGリーダートレーニングコースを提供している。同コースは、11日間のコースを4つのタームに分け、CMMI、チーム活動・診断・プロセス定義、PSP/TSP、PMBOKなど、既に確立したベストプラクティスや知識体系など、幅広い範囲の内容を、座学、演習、事例紹介により実施している。また、人材育成カルテを導入し、受講者に対して不足したスキル分野を明確にすることで、受講前にコース受講の動機付けを与え、受講後にコース受講の効果を評価する取り組みを行っている⁸⁾。我々のコースに比べ、標準的なモデルや技法を中心に幅広くカ

バーし、長い期間をかけ実施するものとなっている。

福原は、NECのプロセス改善のコンサルティングサービスの一環としてSEPG立ち上げトレーニングを実施している。個別に改善を指導する際、SEPG要員の立ち上げトレーニングはCMMIのモデル及びアセスメント座学を用い、モデルと現状の対応づけを使った課題抽出、改善策の検討を共同で行う。我々のコースと、スキル習得に自職場の問題の解決を指導する点は同じであるが、複数の組織向けの集合形式のコースとして設計されたものではなく、SEPG要員の育成面だけを捉えるとその育成効率は低い。

6 おわりに

SEPG要員の育成を企業グループ全体で効率的に行うために、三菱電機グループはSEPG要員育成コースを企画・設計し4年間にわたり実施し、事業所のSEPG要員育成に貢献してきた。

本コースは、企業グループのSEPG要員(候補)を対象に、4日間という短期間で効率よく育成目標を達成することを狙いに、集合講座の形式を採りつつ現場のSEPG要員が抱える自職場の課題を解決する能力の獲得に主眼に置いている。そのためのスキルの達成目標を5つの要求事項として設定して、それらを満足するように、設計している。

ワークショップは、リードアセッサクラスの講師の指導の下で、プロセス改善における問題解決手順を自職場の問題で繰り返し実施することで、リアルで実践的な能力の習得を行うことができる構成になっている。講座受講者に対するアンケート結果から、講座が高い質と育成効果を持っていることがわかった。

今後、育成効果の向上を行うために、受講者が受講後に改善施策を確実に実践できるようにフォローを行うことや、ワークショップを拡充することを検討していく必要がある。

参考文献

- 1) 小笠原秀人, 艸薙匠, 會澤実: 全社的なソフトウェアプロセス改善活動の実践結果とその振り返り, SQiPシンポジウム2011
- 2) 近藤聖久, 久野倫義, 中島毅: 品質改善のためのプロセス診断技術, 三菱電機技報Vol.87, No.4, pp.210-213, 2013年4月
- 3) IPA/SEC プロセス改善ナビゲーションガイド～虎の巻編～
<https://www.ipa.go.jp/sec/publish/trn08-009.html>
- 4) Watts S. Humphrey: Managing the Software Process, Addison-Wesley (1989)
- 5) ISO/IEC 15504:2012 Process Assessment -- Part 5 An exemplar software life cycle process assessment model.
- 6) CMU/SEI編: CMMI for Development, Version 1.3 (2010). CMMI is registered with the U.S. Patent and Trademark Office.
- 7) 福原綾介: ソフトウェアのプロセス改善モデルを有効活用するSEPGのスキルアップ, Leader's Report 2011/10
- 8) 二宮隆夫, 緒方勝, 阪本太志: SEPGリーダートレーニングコースへの「人材育成カルテ」の活用, SPI Japan 2011
- 9) 小川清, 河野文昭, 尾形俊彦, 穴田直也, 近藤聖久, 白杵誠: プロセス改善推進者向け人材育成の在り方, SPI Japan, 2012
- 10) IPA/SEC編: 組込みスキル標準ETSS2008 キャリア基準 V1.2
- 11) IDEALsm, PSPsm, TSPsm and SEPGsm are service marks of Carnegie Mellon University.

組み込みソフトウェア開発をめぐる新潮流

～IoT時代に求められる サービス・トランスフォーメーション～

SECシステムグループリーダー 山下 博之

社会生活や経済活動への情報通信技術 (ICT) の活用が進むに伴い、組み込みソフトウェアに対しても大規模・高機能化が求められるようになり、半導体技術の著しい進展がその実現を後押ししている。一方、組み込みソフトウェアの大規模・高機能化が進むと、その開発効率の向上が望まれるようになり、開発環境や開発技術などが発展してきている。これらは、更に組み込みソフトウェアの大規模・高機能化をドライブしている。IoT時代を迎え、組み込みシステムはエンタプライズ系システムと連携し、多くの分野でサービス指向がより強くなりつつある。この状況に対応するために、組み込みソフトウェアの開発スタイルの見直し、開発組織のスキルと体制の強化が求められる。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、組み込みソフトウェアの現状とIoT時代に向けた課題を概観する。

1 組み込みシステム開発の変遷^[1]

世界最初のコンピュータENIAC (Electronic Numerical Integrator and Computer)が開発された1946年から下ること25年、インテル社からマイクロコンピュータシステムMCS-4が1971年に発売された。このマイコンは、日本のビジコン社の依頼で開発され、プリンタ付き電卓に使われたが、ENIACに匹敵する四則演算能力を備えたと共に、外部機器制御をソフトウェアによって行うものであった。これは、組み込み機器の原形の一つと言えよう。

MCS-4の核となる4ビットマイクロプロセッサ4004が登場して以降、小型・低価格という特徴とプログラム制御の柔軟性から、様々な機器・システムにマイクロプロセッサが適用されていった。それに伴い、性能や機能を高めたマイクロプロセッサも次々と開発された。現在では、家電、自動車・車載機器、携帯電話/スマートフォン、テレビ、AV機器など、ほとんどの電気機器・システムにプロセッサが組み込まれ、それらの上では組み込みソフトウェアが動作している。近いうちに、最近注目されている人工知能も組み込みAIとして様々なシステムに搭載されることは、ほぼ間違いないであろう。

このような組み込みシステムの発展過程で、組み込みソフトウェアを取り巻く環境は、エンタプライズ系(情報処理システム向け)ソフトウェア開発の技術を取り込みながら進化してきた。例えば、当初はアセンブラによる開発であったが、リアルタイム制御を特徴の一つとする組み込みシステム向けのリアルタイムOSとして、1979年にOS-9が、1980

年にVxWorksがそれぞれ発表された。その後、1987年にITRON仕様が公開され、1998年には汎用PCなどに使われていたLinuxをベースとした高機能OSが組み込みシステムでも使われ始め、2007年にはAndroidが発表されている。組み込みソフトウェアの開発環境では、Embedded Javaの仕様が1997年にSun Microsystems社から発表されている。プログラミング言語関連では、1998年にC言語の標準コーディングガイドラインであるMISRA-C^{※1}が発表されている。また、車載情報端末向けのソフトウェアプラットフォームAuto PC 1.0が、1998年にマイクロソフトから提供開始された。更に、2000年には、組み込み用データベースSQLite Version 1.0がパブリックドメインソフトウェアとして公開されている。

このようなシーズとしての情報通信技術 (ICT) と、ICTを活用した製品やサービス(ニーズ)とは、相互に好循環の影響を及ぼし合いながら速いペースで進展してきた。例えば、iモード携帯電話の登場によりモバイル・インターネットが普及し、コンテンツ配信が急速に発展した。それにより一層高まったサービス・ニーズに応える形でモバイル端末がスマートフォンやタブレットに進化すると、その高い機能・性能を活用した現地決済やオンラインゲームなどの高度なサービスが次々と生まれている、といった具合である。別の例では、自動車に通信機能が搭載されるようになると、自動車に対する様々な情報提

【脚注】

※1 欧州の自動車関連ソフトウェア業界団体MISRA (The Motor Industry Software Reliability Association) が策定。

供サービスが生まれたが、これが自動車からの情報収集のニーズを喚起し、双方向の通信機能が具備されるようになった。その結果、気象や渋滞などの自動車から発信される様々な広域の環境データを集約して提供するというサービスに発展した。

前者のように、もともとITサービスの端末などとしてITサービスと共に使用されてきた組み込み機器・システムも、後者のように、当初は単独で使用されていたものがネットワークを介して接続されるようになってきた組み込み機器・システムも、もはや区別なく、ITサービスと連携して発展するようになってきた。IoT時代には、この傾向はますます強くなるものと思われる。

また、社会生活や経済活動へのICTの活用が進むに伴い、組み込みソフトウェアに対しても大規模・高機能化が求め

られるようになった。組み込みシステムの少リソース、高リアルタイム性という特徴は、組み込みソフトウェアの大規模・高機能化を阻害するものであるが、半導体技術の著しい進展が、その実現を後押しする形となった。一方、組み込みソフトウェアの大規模・高機能化が進むと、その開発効率の向上が望まれるようになった。それを実現・推進するのが、開発環境や開発技術などの発展である。これらは組み込みソフトウェアコード量の増大を招き得ることから、組み込みシステムの特徴は開発効率の向上をも阻害するものであるが、半導体技術の進展がそれを緩和している。

このように、開発効率の向上が更に組み込みソフトウェアの大規模・高機能化をドライブするという循環が生じている。(図1参照)

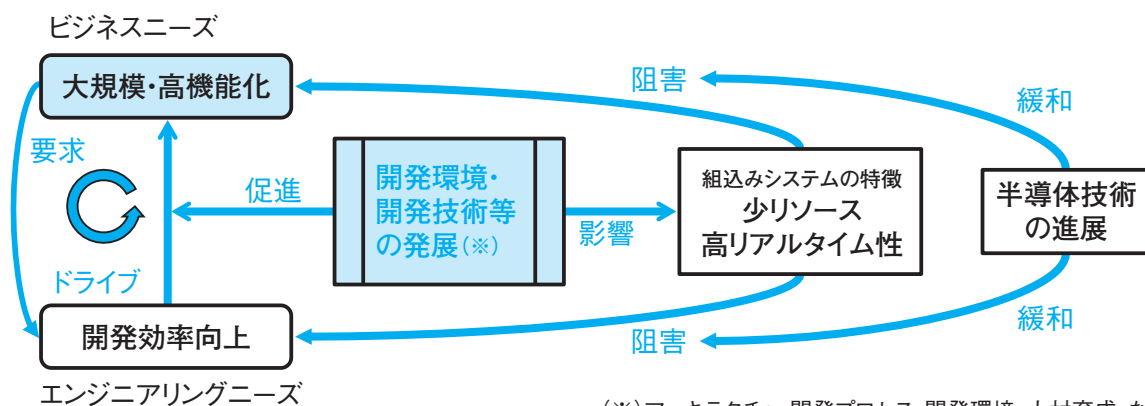


図1 組み込みソフトウェア開発の循環

ここでポイントとなる開発環境・開発技術などとしては、アーキテクチャ、開発プロセス、開発環境(ツールなど)、人材育成などが考えられる。本稿以降では、組み込みソフトウェアを中心とするこれらに関する最近の動向について、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。本稿では、その前段として、これらについての状況を概観し、簡単な考察を加える。

2 組み込みシステムのアーキテクチャ

組み込み機器・システム及びそれらの開発は、一般に、次のような特徴を持つ：

- 利用可能なコンピューティング資源が少ない
- リアルタイム性の要求が厳しい
- 外部環境とのインタラクションが多い
- 顧客の要求が明確ではない

これらの特徴について、大規模・高機能化の影響を簡単に分析する。

(1) 利用可能なコンピューティング資源が少ない

とくに、組み込み機器・システムに搭載可能なメモリサ

イズが大きな制約である。半導体技術の著しい進展により、この制約は緩和されるかに見えるが、組み込み機器・システムの大規模・高機能化に伴って組み込みソフトウェアが必要とする資源量も増大する。また、組み込みソフトウェア開発の効率向上などを目的とするコード生成ツールや機能パッケージなどの使用により、機能当たりのコードサイズが増大している。従って、相対的にはこの状況は大きくは変わらない。

(2) リアルタイム性の要求が厳しい

マイクロプロセッサやメモリの性能向上に伴い、リアルタイム性要求を満たすことが容易になるかに見えるが、プラットフォームやコード生成ツールなどの使用により、各処理の所要ステップ数は増大している。従って、相対的にはこの要求の厳しさは大きくは変わらない。

(3) 外部環境とのインタラクションが多い

組み込み機器・システムが前面となって、外部の様々なデータの収集が行われる。IoTでは、その種類や数が飛躍的に増大する。従って、組み込みソフトウェア開発への要求もますます増大する。

(4)顧客の要求が明確ではない

家電製品やスマートフォンのように、多くの組込み機器・システムは、顧客からの明示的な要求に基づいて開発されるわけではない。これらの場合、開発者サイドが経験と想像に基づいて機器・システムへの要求を設定し、未来予測をその要求に反映し、多くの関係者(関連部署)と協力しながら開発を進める。近年、通信技術などの進展により、市場からのフィードバックを迅速に行う仕組みを設けることができるようになり、短いサイクルで機能を積み上げ、評価しつつ、製品の価値を高めていくスタイルが普及しつつある。

一般的には上記のような特徴を有する組込みシステムであるが、システムの種別により採用するアーキテクチャは異なる。

例えば、スマートフォンでは、単一のプロセッサ上で多数のアプリケーションが動作する。これらのアプリケーションは、頻繁に更新される。このような機器は、パーソナルコンピュータやメインフレームなど、単一のコンピュータと類似した構成と言える。

他方、自動車では、エンジンやブレーキ、パワーウィンドウなどの制御対象対応に専用のプロセッサが存在する。プロセッサの合計は数十個に及び、最近では100個程度搭載した車もあるという。各プロセッサ上のアプリケーション、すなわち制御プログラムは、更新されることは多くない。なお、各プロセッサ間の連携については、当初は少なかったものの、高機能化に伴い急激に増加傾向

にある^{※2}。このような構成は、エンタプライズ系の機能分散システムあるいは広域のサービス連携システムと類似しているとも言える。レスポンスタイム制約の下で、(制御)データの配置と交換に工夫が必要となる。組込みシステムでは、とくに、リアルタイム性の制約が厳しいため、ハードウェアとソフトウェアとの協調が重要となる。

更に、継続的に機能拡張を行うシステムにおいては、アーキテクチャの問題は一層重要度が増す。また、大規模・高機能化に伴うシステム構造の階層化傾向が顕著になってきている。このような状況で、機能拡張を効率的に行いつつ、システムの品質を維持し続けるために、システムの特徴に応じた適切なアーキテクチャを採用する必要がある。

3 組込みシステムの開発プロセス

2013年のET-West講演でクリストファー・テイトは、日米における組込みデバイス(機器・システム)を比較しており^[2]、それを図2に示す。同図によれば、日本のデバイスは米国のデバイスに比べ、組込みソフトウェアの比率が小さい。また、米国のデバイスには、日本のデバイスにはない「インターネットを介した改善」の部分が含まれる。概念的ではあるが、米国では当時から既に組込みソフトウェアのオンライン更新の導入が進んでいたことが分かる。なお、組込みソフトウェア開発データ白書^[9]によれば、日本でも約半数がオンライン保守可と回答している。

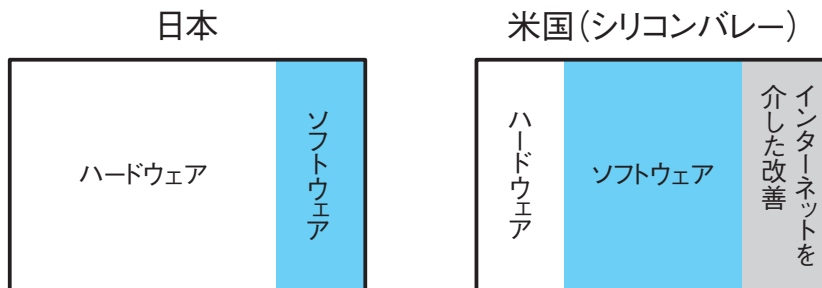


図2 日米における組込みデバイスの比較

クリストファー・テイトに倣ってIoT時代の組込みデバイスのモデルを描くと、図3の左上のようになろう。最近では、FPGA (Field Programmable Gate Array) などのPLD (Programmable Logic Device) の大規模化が進んでハードウェア論理の書換えが容易となっており、その状況を盛り込んだ。IoT時代には、組込みデバイスなどによって収集された各種データ(デバイスやその周辺環境の状態、デバイスの利用状況、など)がクラウドなどに集積され、ビッグデータ解析などに利用されると想定されている。デバイスのハードウェア論理やソフトウェアの更新の仕組みと合わせて、図3のモデルに含めた。

このようにして収集した組込みデバイスの利用者(エン

ドユーザ)の声を分析し、デバイスの機能拡張や修正を行うことができる。このようなエンドユーザからのフィードバックは、デバイスの開発開始時点では掴み得なかった「真の要求」と言える。

このようなデバイスの(定期的な)更新を適切に行うための有力なソフトウェア開発手段が、アジャイル開発である。

【脚注】

※2 例えば自動車の場合、エンジン制御が単独ではあり得ない状況であり、燃費、排ガス(環境)のトレードオフに対応するためにエアコン、シャーシー系との連携が必須になっているとのことである。

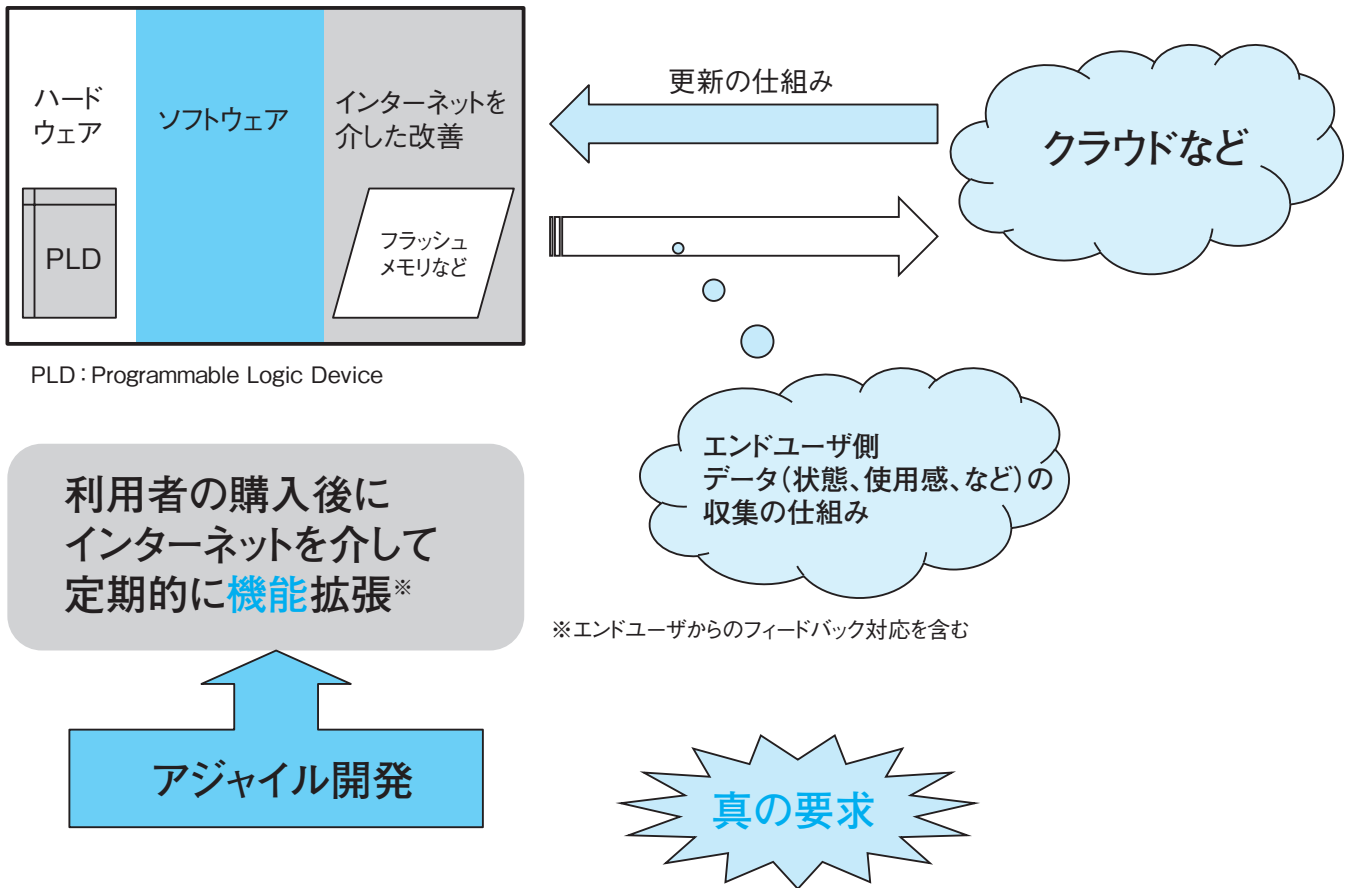


図3 組み込みデバイスとアジャイル開発

4 組み込みシステムの開発環境

ソフトウェアの開発プロセス(スタイル/手法)を効果的に運用するためには、適切なツールを活用することが重要である。そして、アプリケーション・ソフトウェアの要求管理、設計、実装、テスト、リリース管理などのライフサイクルにわたって一貫した管理が有効である。このような各フェーズ対応に適切なツール群を活用した統一的なプロセス管理は、ALM (Application Lifecycle Management)と言われる。

筆者が各開発現場の声を聞いたところでは、残念ながら我が国では、このようなツールが十分に活用されているとは言えないという印象を持っている。OECDの調査においても、我が国の労働生産性は先進国の中でも下位に位置する^[3]。労働集約性の高いソフトウェア開発の現状がこの調査結果と合致している。この状況は、一つには、多重下請けというソフトウェア産業の構造的な問題が起因していると思われる。すなわち、大手ITベンダの系列ごとに独自のツールを使用することになり、汎用的なツールやオープンソースソフトウェアの普及がなかなか進まないのである。

こうした状況は、モデルベース開発(MBD: Model Based Development)といった技術の普及が進まない理由の一つになっているのではないと思われる。MBDは、大規模・

複雑化するソフトウェアに対し、効率的で高品質な開発を可能とする有力な手法であるが、専用のツールが欠かせない。後述の、MBDの全面的採用に踏み切ったトヨタ自動車のような事例^[4]が、今後増えていくことを期待したい。

5 組み込みシステムの開発組織・人材

前述のモデルにおけるもう一つのポイントは、組み込みデバイスとエンタプライズ系システムとの連携である。これはIoTという語自体から当然と言えるが、組織・人材の観点で、より重要な意味を持つ。

技術者は、機器・システムだけを見ては不十分である。すなわち、イノベーションに結び付きにくい。機器・システムの機能拡張の仕組み(サーバー/バックヤード/クラウド側)の理解と、機能拡張項目選定のトリガ(利用者の声を捉える仕組み)の理解が必要となる。このことは、組織としては組み込み系とエンタプライズ系の協働、個々の人材としては両スキルの獲得ということに帰着する。

また、製品やサービスのアーキテクチャとは別に、組織構造としてのアーキテクチャも重要である。顧客のニーズに応えた製品やサービスを継続的に提供し続けるための組織とは、どのような機能を有し、それらがどのように連携するべきか? ちなみに、コンウェイの法則とい

うものがあり、「設計組織により生み出されるシステムは、その組織とそっくり同じ構造となる」のだそうである。

対象システムにどのようなアーキテクチャを採用するか、多様な開発スタイルのうちから、システムのどの部分にどの開発スタイルや手法/ツールを用いるかを決定するためには、高度なデザイン力が求められる^[5]。なお、組込みソフトウェア関連人材のスキルとしては、IPAは組込みスキル標準(ETSS Series)^[6]を作成し、現在は、一般社団法人スキルマネージメント協会(SMA)^[7]がその普及啓発を行っている。

上述のように、最近では、組込みシステムに関するスキルだけ保持していれば十分ということではなく、エンタプライズ系の素養もある程度は必要になってきている。これは、すべての人材に言えるというわけではないが、少なくとも組織としては保持すべきものである。従って、「組織スキル」ということになる。

6 IPA/SECの取り組み

IPA/SECでは、その設立以来、組込み系のソフトウェア・エンジニアリングに関する取り組みを行ってきた。その成果は、次の書籍などにとりまとめ、公開している。

- ESxR (Embedded System xxx Reference) シリーズ^[8]
 - 組込みソフトウェア開発向けコーディング作法ガイド(ESCR)
 - 組込みソフトウェア向け開発プロセスガイド(ESPR)
 - 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(ESMR)
 - 組込みソフトウェア開発向け品質作り込みガイド(ESQR)
 - 組込みソフトウェア向け設計ガイド(ESDR)
 - 組込みソフトウェア開発における品質向上の勧め[テスト編~事例集~](ESTR)
 - 組込みソフトウェア開発における品質向上の勧め[バグ管理手法編](ESBR)
- 組込みスキル標準(ETSS Series)^[6]
- 組込みソフトウェア開発データ白書^[9]

また、最近では、機器やシステムの障害事例を分析し、教訓としてとりまとめている^[10]。これは、経験を開発・運用プロセスの改善につなげるものである。

これらの成果、特にESxRシリーズについては、IoT時代に対応して改訂する必要があるかもしれない。本特集の記事がその促進材料となることを期待したい。

最近、組込み機器・システムは大規模・複雑化しており、ほかのシステムや人との連携も増大している。このような機器・システムにおいて、安全上のリスクを設計時に分析したり、トラブルを解析したりする技術に対しても、従来とは異なる考え方が求められるようになってきている。更に、セーフティとセキュリティの両方を考慮した効率的な設計を求められている。セーフティは、一般に、

いったん設計した後で要件が変わることはあまりない。他方、セキュリティについては、提供後も新たな脅威が発生することがあり、提供者には顧客のもとにある既存の製品・サービスにおいてもその脅威への対応が求められる。IPA/SECでは、このような傾向に対応すべく、次のような新しい技術の検討にも着手している。

- ハザード分析(STAMP/STPA)／障害診断技術(事後V&V)^[11]
- Safety&Security設計
 - システムの大規模化・複雑化に伴う影響の一つとして、トラブル解析の困難さの増大がある。IPA/SECでは、機器やシステム障害事例の分析に取り組んでいる^[10]が、次のような事例が発生している：
 - 通信回線制御ボード内のバッファメモリ(キュー)のオーバフローにより機能停止したが、そのバッファメモリはプログラムからは見えない内部バッファであり、動作中のキューの状態が全く分からない。
 - 機器間通信スイッチ内のメモリの読み書きデータ不正により機能停止したが、スイッチ内のメモリの動作状況について、外部からは全く分からない。

かつての通信機器には、プロセッサやメモリが搭載されていなかったため、監視対象となる状態は、処理中/アイドルの別やデータの有無程度であり、あまり監視の必要はなかった。その後、それらの機器は次第にインテリジェント化し、今では、プロセッサや相当容量のメモリを含むのが普通である。にもかかわらず、機器の監視が従来の考え方のままであるならば、何が起きているかを適確かつ迅速に把握することはできない。

これに関連して、興味深い事例がある。自動運転支援で有名なテスラ社製の電気自動車では、事故を起こした車のログの解析により、次のことが分かったという^[12]：

「該当車のログデータを分析したところ、この車両は正常に手動運転される状態にあり、事故が起きた際にも、また、その数分前にも、自動運転やクルーズコントロールの設定はされていなかったことが確認された。また、データは、約9.7km/hの速度で走行中に、突然、アクセルペダルが100%踏み込まれたことを示している。ドライバーのこの行動に伴い、車両は指示通りに駆動力を増加して加速した。」

また、事故などによりエアバッグが展開した場合、その事実などが事故直後に自動車メーカーに自動的に通知されるというセーフティ&セキュリティ機能が、米国では比較的広く浸透しているという。ただし、通信回線の容量によっては、事故の詳細データのすべてが転送されないこともあるらしい^[13]。

いずれにしろ、トラブル解析を容易にするために、適切なログの取得機能が必要である。また、故障の予兆を監視するための機能も有効であろう。これらの機能は、ハードウェアとソフトウェアとの協調により、効率的に実現される。

7 まとめ～IoT時代に向けて

本誌第44号において、IoT時代に求められるソフトウェア開発スタイルについての特集^[5]を掲載した。同特集の各記事は、必ずしも組み込みソフトウェア向けではないが、IoT時代の組み込みソフトウェア開発に対する多くの示唆を含んでいる。

組み込みソフトウェアの開発では、従来、多くの現場で派生開発^[14]が行われてきた。これは、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要求機能などを具備するソフトウェアを作るものである。IoT時代では、とくにコンシューマ向け機器においては、外部環境の変化に機敏に対応した商品/サービスの投入が求められることから、いわゆるアジャイル開発^[15]も必要となる。このアジャイル開発と派生開発とは対立するものではなく、両立するものであると考える。各現場の経験などの状況に応じ、派生開発をアジャイルに回す、あるいはアジャイル開発に派生開発の要素を採り入れる、などの工夫を行うことが有効であろう。更に、制御のパターンやルールがある程度整理され、パラメータ化できるような部分に対しては、超高速開発^[16]を採用することも考えられる。

ここで、新しい開発スタイルを全面的に採用した事例を紹介しておく。

トヨタ自動車は、2015年12月に発売した新型「プリウス」から、ハイブリッドシステムに用いる制御ソフトウェアの開発プロセスを全面的に刷新した^[4]。

新型プリウスでは、より高度な制御が求められたほか、制御に対する要求が多様になった。このため、制御ソフトウェアを全面的に刷新すると共に、開発プロセスにも全面的にモデルベース開発を取り入れた。

具体的には、複数の海外ベンダから自動コード生成ツールやソフトウェア検証ツール、構成管理ツールなどを導入し、開発プロセスの効率化を図っている。制御ソフトウェアそのものも、品質を確保しつつ新機能を迅速に追加できるようにするため、処理を集約したり、モード選択の組み合わせ数を減らしたりするなど、ソフトウェアの構造を簡略化した。これらにより、ソフトウェアの複雑度は従来に比べて半減したという。また、従来はソフトウェアの一部の開発をサプライヤに委託していたが、今回はすべて内製化したという。

この刷新に先立ち、トヨタ自動車社内では、数年前から、制御システム基盤開発を推進する部署がハイブリッドシステム開発の部署にMBDのメリットを説いて回っていたという。そうこうするうちに、3年前に、自動コード生成ツールの性能向上、すなわち、手書きの場合と同程度のコードを生成できることが明らかになったことなどを契機に、MBDの採用を決断したという。その前後で、2年間ほどをかけて、技術者へのMBDの教育を行ったという。

以上のように、組み込み機器・システムは、従来の制御

対象に特化し、いったん設計・提供された後はほぼそのままライフサイクルを全うするものから、顧客ニーズに応じて変更、拡張されるものへと変化してきている。いわば、組み込み機器・システムのサービス化が進んでいると言えよう。このような動きに対応するために、組み込み機器・システムの開発組織には、「サービス・トランスフォーメーション」が求められる。これを実現するために必要なことは、開発スタイルのみならず、アーキテクチャ、開発環境や組織・人材など、企業全体(エンタープライズ)のレベルで取り組む、「エンタープライズシステムズエンジニアリング」であろう。

【文献】

- [1] 拡大し続ける「組み込みシステム技術」の29年史, Tech総研, http://next.rikunabi.com/tech/docs/ct_s03600.jsp?p=001274
- [2] クリストファー・テイト: イノベーションを生み出す日本へ、再び ～ソフトウェアとハードウェアの対話が、日本に強さをもたらす～, ET-West 2013 ヒートアップセッションHU-5講演, 2013年6月14日, 大阪.
- [3] 平成27年版 労働経済の分析, 厚生労働省, 平成27年9月.
- [4] 阿部眞一: 第4世代プリウスとモデルベース開発の活用, dSPACE Japan User Conference 2016, 平成28年6月3日.
- [5] 山下博之, 室修治: 多様化が進むソフトウェア開発スタイル, SEC journal, Vol. 11, No. 4, IPA/SEC, 2016年3月1日.
- [6] 組み込みスキル標準 (ETSS Series), <http://www.ipa.go.jp/sec/softwareengineering/std/etss.html#200>
- [7] 一般社団法人スキルマネジメント協会, <http://www.skill.or.jp/index.php>
- [8] IPA/SECの組み込み系成果, <http://www.ipa.go.jp/sec/softwareengineering/std/emb.html>
- [9] 組み込みソフトウェア開発データ白書2015、及び、組み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], <http://www.ipa.go.jp/sec/reports/20151116.html>
- [10] 重要インフラ分野のシステム障害への対策, <http://www.ipa.go.jp/sec/system/index.html>
- [11] 「はじめてのSTAMP/STPA ～システム思考に基づく新しい安全性解析手法～」の公開, <http://www.ipa.go.jp/sec/reports/20160428.html>
- [12] テスラ, 「モデルXが勝手に加速した」というオーナーの主張を否定, AutoBlog日本版, 2016年6月9日, <http://jp.autoblog.com/2016/06/09/tesla-rejects-claim-unintended-acceleration-model-x/>
- [13] 野辺継男: テスラEVでまた事故, 喫緊の課題は「機能告知の徹底」, 日経テクノロジーオンライン, 2016/07/09, <http://techon.nikkeibp.co.jp/atcl/column/15/415543/070900042/>
- [14] 清水吉男: IoTの時代における派生開発の対応, SEC journal, Vol. 11, No. 4, IPA/SEC, 2016年3月1日.
- [15] 平鍋健児: 変化を味方につけるアジャイル開発, SEC journal, Vol. 11, No. 4, IPA/SEC, 2016年3月1日.
- [16] 関隆明: ユーザーイニシアティブを可能にする超高速開発, SEC journal, Vol. 11, No. 4, IPA/SEC, 2016年3月1日.

モデルベース開発とコード解析を用いた組込みソフトウェアの開発

(先進的な設計・検証技術の適用事例報告書 2015年度版より編集部要約)

本事例は、アルプス電気株式会社(以降、アルプス電気)の自動車向け組込みソフトウェア開発における、「設計・検証コストの改善」事例である。

自動車の高機能化・電子制御化に伴い、一台の自動車に搭載されるECU(電子制御ユニット: Electronic Control Unit)の数は年々増加し、また要求される機能も複雑化している。ECUの増加に伴い、実現するソフトウェアにおいては、ソースコード量の増加やECU間のネットワーク制御が必要になり、ソフトウェアの品質が全体の性能・信頼性に大きく影響を与えると共に、設計・検証コストが増大する課題が表面化してきた。本事例では、これらの課題を解決するために「モデルベース開発」と「コード解析」を適用した独自の手法であるMDD(Model Driven Development)にTDD(Test Driven Development)の思想を取り入れ、モデルベース開発とテスト駆動開発を融合させプロセスの洗練化を図ったものを活用した。その結果、従来開発に比べて開発工程が減ることでコストが削減でき、視覚的に理解しやすいモデルを検証することで、検証効率と品質を向上することができた。また、コード解析では、市販コード解析ツールと自社開発ツールを組み合わせることで、人手で行っていた検証作業をツールに置き換え、スキルによらない検証が行え、検証コストの削減につなげることができた。

1 技術・手法を導入した理由や経緯

1.1 モデルベース開発

(MBD: Model Based Development)

年々加速度的に大きくなる車載ソフト規模の市場要求と、開発能力の関係を図1に示す。

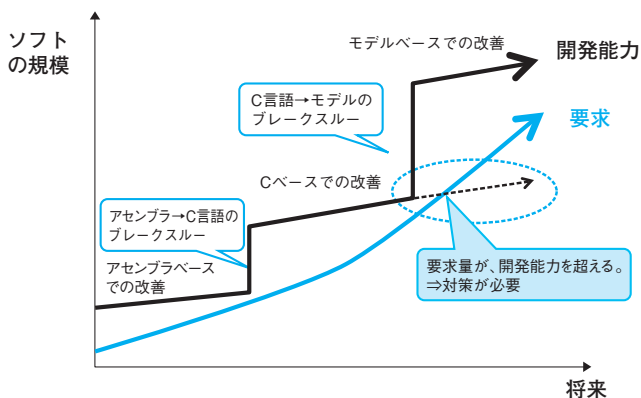


図1 モデルベースの開発の必要性

開発現場での地道な改善活動により開発能力は毎年少しずつ向上しているが、それだけでは近年高まるソフト

ウェア開発の要求量と要求スピードについていくことができない状況となってきている。かつて車載ソフトの開発現場ではアセンブラベースの開発からC言語ベースの開発に移行することで大きな開発能力向上のブレークスルーを経験した。しかし、現在ではもはやC言語ベースの改善の積み重ねでは市場要求についていくことは困難になりつつある。C言語ベース開発から移行する第2のブレークスルーとして、制御系を含む製品開発の手法として注目されている「モデルベース開発」を設計に適用することにした。

一般的にモデルベース開発の目的は、「複雑なシステムを効率的に開発すること」で、下記のような利点がある。

- ① 仕様書の明確化と再利用性の向上
- ② シミュレーションをしながら仕様を決定できる
- ③ 仕様書から自動でコードが生成されるために、バグが入りにくい
- ④ 制御対象と制御装置を同時に開発できる

モデルベース開発の手法はソフトウェアの開発規模の増大にも対応できる手法であり、モデルの作成ができるので、ソースコードが自動生成されるので、プログラミングのミスが防止でき、品質の向上が見込めると共に生産性の向上も期待できる開発手法と言える。

1.2 コード解析

設計においてはモデルベース開発を適用したが、検証については、下記の理由から各種ツールを利用した「コード解析」を実施することとした。

- ① 人による検証からツールによる検証への置き換え
- ② 顧客要求に応じた各プロジェクトのサポート
- ③ ソフトウェア品質の可視化

コード解析を実施することで、今後増え続けるであろう検証負荷を軽減し、開発効率及び開発品質を向上する。図2に示すように、コード解析の目標は、統合損失(品質損失+検証コスト)を最小にする点を求めると共に更にそれを小さくすることである。

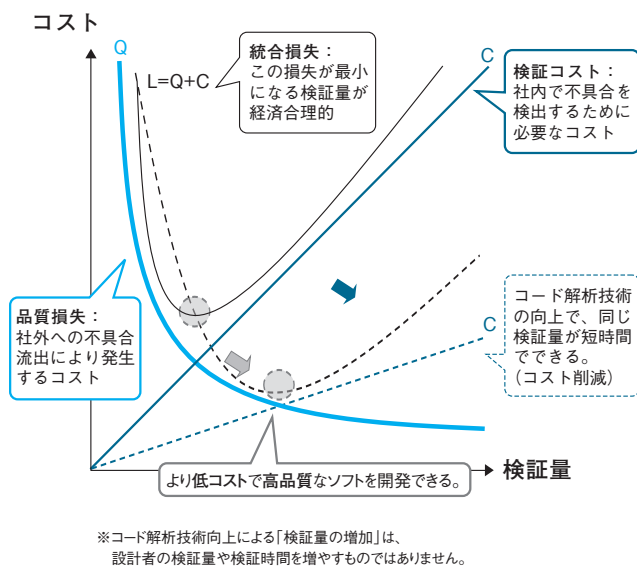


図2 コード解析の目標

2 スムーズな導入と活動定着のための事前準備や工夫

2.1 ワーキング活動と開発体制

MBD開発に興味のある有志を中心にワーキンググループ(WG)を構成し、2週に1回程度の頻度で定期的に勉強会や討論会(いわゆる小集団活動)でツールのカスタマイズなどを実施した。MBDのWG活動については既に10年以上継続している。WG活動の結果として、従来のハンドコーディングと同等のコードサイズを実現するMBD開発ができるようになった。

また、コード解析においても同様にコード解析WGを構築し手法の学習や習得に努めてきた。活動は2週に1時間程度の周期で実施してきた。

2.2 開発工程ごとの解析項目

解析種別により複数の解析ツールを用途に応じて使い分けている(表1)。また、図3に示すように、コード解析(専任者)が使用する解析ツールと開発工程、コード解析(設計者)が使用する解析ツールと開発工程を、それぞれ決めている。

表1 コード解析ツール一覧

解析種別	検証項目の例	ツール
ランタイムエラー	配列オーバーフロー、ゼロ割など	Polyspace®
MISRA-C	MISRA-C 2004対応	QAC®+M2CM
ソフトウェアメトリクス	サイクロマティック複雑度、実行行数	QAC®+自作
コーディングルール	社内コーディングルールへの適用	QAC®+自作
ソフトウェア構造	タスク間インターフェースなど	Imagix4D
コードクローン	コードクローン率	CCFinder

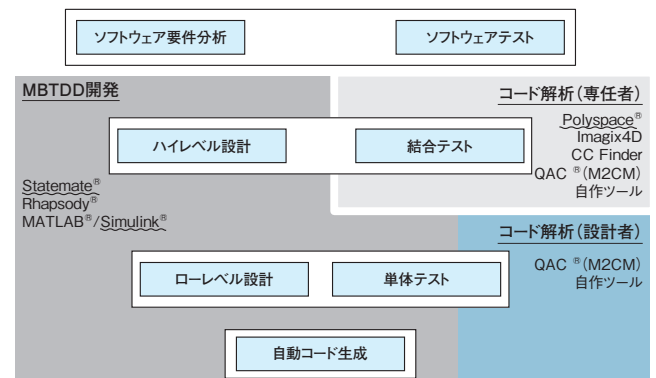


図3 開発工程ごとの専任体制

3 効果測定の方法とその結果

3.1 モデルベース開発

(1) 適用状況

現在、アルプス電気では適材適所でモデルベースの開発を実施している。適用対象事例を表2に示す。適材適所のモデルベース開発では、モデル仕様に基づいて開発プロセスを構築する手法である広義のMBD (Model Based Development)と、組込み制御システム開発の狭義のMBDがあり、車載用組込みソフトで狭義のMBDは全体の20%程度にしか適用できない。このため、残りの80%についてはMDD (Model Driven Development)で開発し効率化を目指している。

表2 モデルベースの開発状況

名称	適用対象項目	手法	使用ツール	
広義のMBD	狭義のMBD 制御系アルゴリズム開発	制御ブロックモデリング	MATLAB®/Simulink®	
	MDD	アプリケーション開発 (製品機能実装部)	構造化モデリング	Statemate®
		プラットフォーム開発 (ハードウェア制御部)	オブジェクト指向モデリング	Rhapsody®

(2) MBTDD開発

アルプス電気ではMDD (Model Driven Development)にTDD (Test Driven Development)の思想を取り入れ、モデルベース開発とテスト駆動開発を融合させてプロセスの洗練を行うMBTDD (Model Based Test Driven Development、アルプス電気の造語)開発を実施している。MBTDD開発では、MDDのモデルシミュレーションの代わりに、モデルから生成されるコードについて直接テストを実施する。ここにTDDの思想を取り入れテストとモデルの洗練を繰り返し実施する。コードではなくモデルを洗練することで本当の意味での設計の洗練になる。

図4に人手によるコーディングをしていた従来プロセスとMBTDDプロセスの比較を示す。

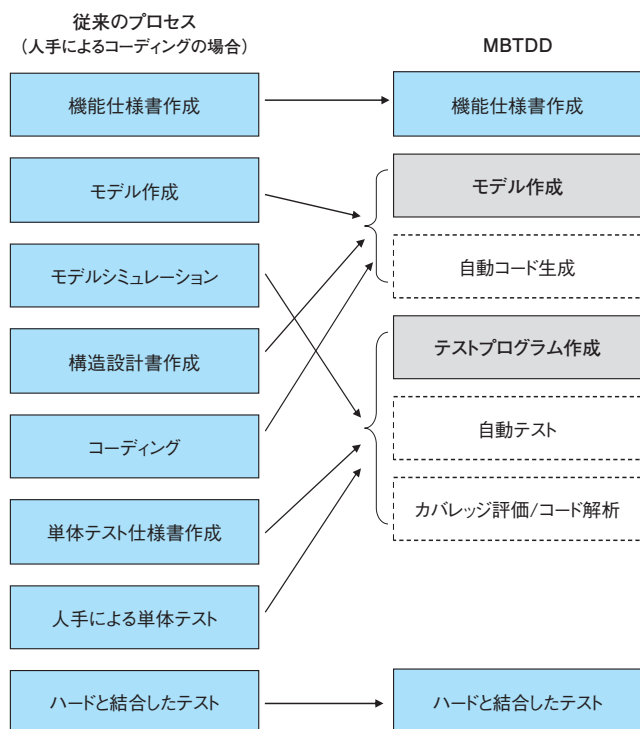


図4 従来のプロセスとMBTDDプロセスの比較

従来プロセスでもモデルは使用していたものの、その主な目的はシミュレーションによるモデルの動作検証であった。モデルシミュレーションで動作検証が済んだモデルをコーディングするために必要な情報を記述した構造設計書を作成し、それに基づいて人手でコーディングしていた。また単体テストも単体テスト仕様書を作成し、人手による単体テストを実施していた。

MBTDDでは、モデルの中に従来の構造設計の情報を埋め込み、最適なコードが生成されるようにカスタマイズしたツールのコード生成機能を用いてモデルからボタン一つでコードを生成している。また単体テスト仕様書をプログラム化し実施を自動化した。

これらの改善によりモデルシミュレーション工程をなくし、モデルができたボタン一つでコード生成し単体テストプログラムを走らせることで従来のモデルシミュレーション、コーディング、人手による単体テストの工程をなくすことができた。また単体テストの実施を自動化し何度でも簡単に実施できるようにしたことと合わせ、この工程で簡単にテストカバレッジ及び設計者自身によるコード解析もできるような環境を整備した。結果としてモデル作成工程とテスト工程を行ったり来たりしながらモデルの洗練と単体テストの洗練を行うようになった。

(3) MATLAB®/Simulink®の活用事例

MATLAB®/Simulink®をカーナビディスプレイの開閉制御に活用した例である(図5)。実機での動作検証前にMATLAB®/Simulink®でシミュレーションし、システム成立性の確認を行った。

また、モデルとコード解析を使った作業(図8)を繰り返すことで、ロバスト性が高く、高品質のソフト開発を行うことができた。

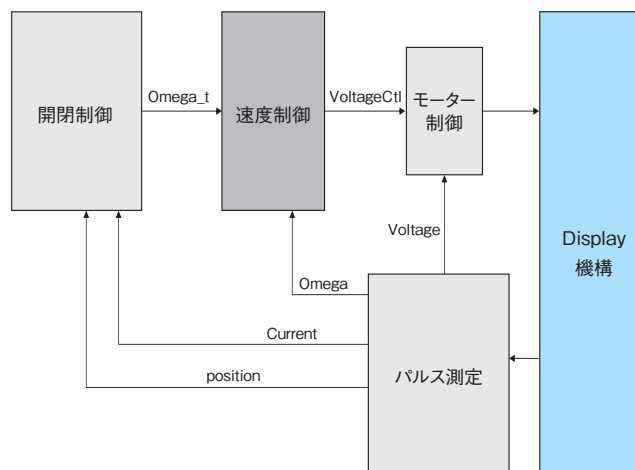


図5 MATLAB®/Simulink®の活用事例

3.2 コード解析

3.2.1 コード解析ツールの選定

アルプス電気では複数の市販コード解析ツールと自社開発ツールを組み合わせ使用しており、より精度の高いコード解析を達成するため、市場のコード解析ツールのベンチマーキングを実施しツール選定を行っている。以下にその評価方法と結果を示す。

(1) 評価方法

解析ツールの選定では以下の5つのステップで評価方法を定義している。

- ① 社内ソフトウェアでよく実装してしまう不具合を選別
⇒過去の不具合について洗い出しを行い、類型化を図る
- ② 不具合がない3種類のソースコードを用意
⇒3種類のソースコードを使い解析ツールを評価
 - Statemate[®]ツールによる自動生成コード
 - Rhapsody[®]ツールによる自動生成コード
 - MATLAB[®]/Simulink[®]による自動生成コード
- ③ 各ソースコードに、選別した不具合を埋め込む
⇒②のソースコードに意図的に不具合を挿入
- ④ 社内外の解析ツールで不具合埋め込みコードを解析
- ⑤ 解析結果を分析し、結果を比較

(2) 解析ツールの評価基準

理想的なコード解析ツールとは、①すべての不具合を検出すること(=不具合検出率が高い)、及び、②正しいコードを誤って不具合としない(=有効警告率が高い)ことである。評価基準を表3に示す。

表3 評価基準

評価項目	計算式	基準
不具合検出率	検出された埋め込み不具合の数/埋め込み数	高いほうが良い
有効警告率	埋め込み不具合に対する警告数/総警告数	高いほうが良い

(3) 評価結果

図6に市販のコード解析ツール(ツールA～F)と自社開発ツール(ALPS)の評価結果を示す。自社開発ツールは、とくに重要な不具合検出率が一番高く、有効警告率も中間に位置しており、現在のツール選定の妥当性が確認できた。また、同時に、本評価結果を元に改善点も明確になり、改善することができている。

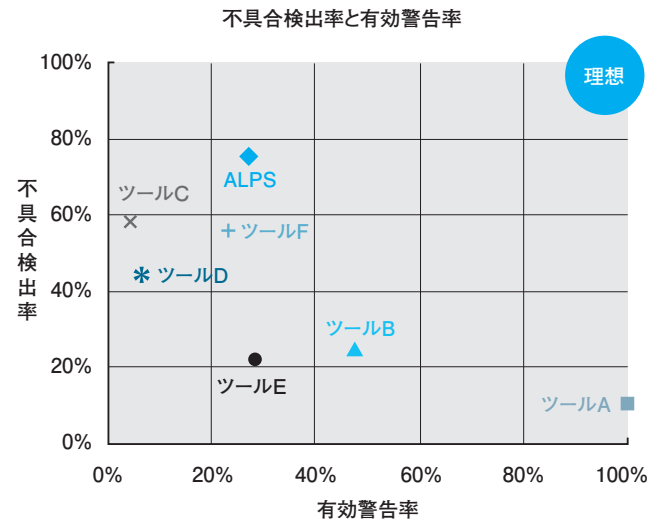


図6 解析ツールの評価結果

3.2.2 ソフトウェア品質の可視化

(1) ソフトウェア品質モデル

ISO/IEC 9126-1 (Software engineering - Product quality - Part1: Quality model)で定義しているソフトウェア品質モデル(表4)の中から「信頼性、効率性、保守性、移植性」の4つの品質特性を使いソフトウェア品質を評価している。

表4 ソフトウェア品質モデル (ISO/IEC 9126-1)

品質特性	品質副特性	主な内容
機能性	合目的性	目的から求められる必要な機能の実装の度合い
	正確性	
	相互運用性	
	セキュリティ	
	標準適合性	
信頼性	成熟性	機能が正常動作し続ける度合い
	障害許容性	
	回復性	
	標準適合性	
使用性	理解性	分かりやすさ、使いやすさの度合い(いわゆる「使い勝手」、「使いやすさ」、「操作性」の概念)
	習得性	
	運用性	
	魅力性	
	標準適合性	
効率性	時間効率性	目的達成のために使用する資源の度合い
	資源効率性	
	標準適合性	

品質特性	品質副特性	主な内容
保守性	解析性	保守(改訂)作業に必要な努力の度合い
	変更性	
	安定性	
	試験性	
	標準適合性	
移植性	環境適応性	別環境へ移した際そのまま動作する度合い
	設置性	
	共存性	
	置換性	
	標準適合性	

(2)ソフトウェア品質の可視化の目的

ソフトウェア品質について、4つの品質特性を5段階評価したものをレーダーチャートで可視化している。これにより次の効果が得られる。

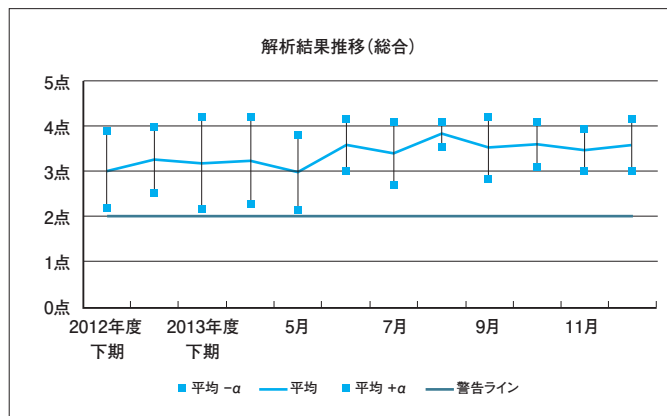


図7 品質状況のモニタリング

3.2.4 複数の市販解析ツールと自社開発ツールの活用

市販ツールには、それぞれ得意な解析分野があるが、一方でアルプス電気のソフト開発では必要としない機能もある。アルプス電気では各市販ツールにおいて解析のためのパラメータの設定をカスタマイズしている。更に独自のフィルターをかけることで、効率的な解析を可能としている。また、市販の解析ツールのカスタマイズで対応できないものはツールを自作し活用している。

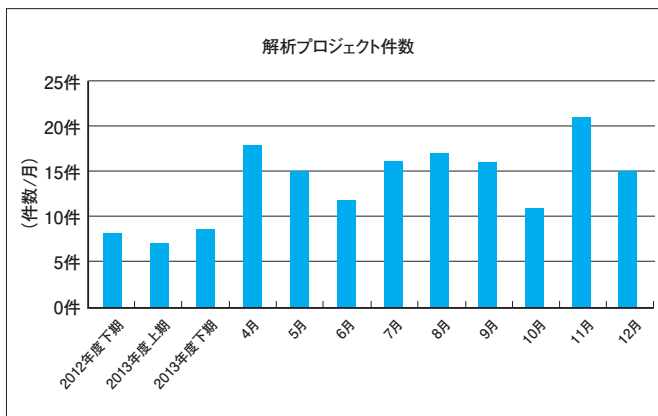
3.2.5 自動生成におけるコード解析の必要性

モデルベース開発は、モデル自体の文法チェックや一

- ① ソフトウェア品質状況を数値化して定量的に把握する
 - ② 定量的データに基づき、品質の良否を判断できる
 - ③ 品質が悪い場合の改善ポイントが明確になる
- また、その結果として、次のような点で品質の良いソフトウェア開発が可能になる。
- ① 不具合が除去されている
 - ② 仕様変更に対応できる
 - ③ 再利用が可能である

3.2.3 コード解析WGの成果

各プロジェクトの品質状況を可視化し、モニタリングすることで、早期に適切な対処を行っている(図7)。従来レビューで活用していたチェックリストを見直し、コード解析で代用できるものを削減した。その結果、30%以上のチェック項目がコード解析で代用できると分かり、レビュー時間の削減に大きく貢献した。また、レビューからコード解析に検証手法を変えたことで、設計者のスキルに依存しない、安定した検証が短時間でできるようになった。



貫性のチェックが実施でき、また、視覚的に理解しやすい開発手法である。しかし、自動コード生成をするためにはプログラム言語に近い記述も必要であり、人為的なミスやスキル不足による間違いがそのまま自動で生成されたコードに反映されてしまう。そのため、自動生成コードでもコード解析で不具合の検証を行う必要がある(図8)。なお、制御系アルゴリズム開発にはMBTDD開発を適用していないため、シミュレーション(図8)が必要だが、MBTDD開発が適用できるアプリケーション開発やプラットフォーム開発では不要になる。

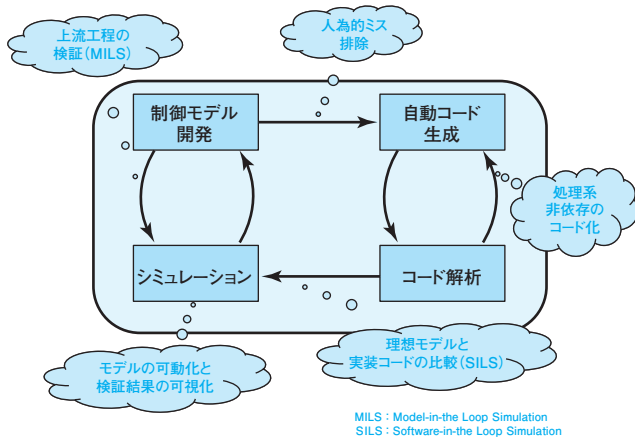


図8 モデルとコード解析を使った開発 (MATLAB®の例)

4

技術や手法の導入後の改善活動、今後の課題

MBD開発導入後は、モデルの中に設計情報を書き入れているので、あいまいになりがちな設計書を作成する必要はなくなった。ソースについても自動生成されるので、コーディング作業は不要になった。今後、モデルベース開発では、モデル用のライブラリの充足、及び、マイコン依存部やデザインパターンなどの実装を簡単にするフレームワークを整備することで、再利用率を上げる予定である。コード解析では、有効警告率を25%から40%程度まで向上させることを目標に、Polyspace®、Imagix4Dなどのツールの解析設定のカスタマイズ、及び、自作のツールの改良を実施する。

5

結果と考察

(1) 品質について

新手法を導入する際に、品質の目標をバグ件数半減とされていたが、結果は表5に示すように大幅に改善できた。バグの数が減ったことと、バグが出ても原因の特定や対策ができるようになりバグ対策の仕事が劇的に減った。打ち合わせの多くを占めていたバグ対策が激減し、モデルやテストの洗練に多くの時間を割く文化が育成された。

表5 品質の結果

プロジェクト	バグ件数
A	約1/4
B	約1/3

(2) 生産性について

新手法を導入する際に、開発速度の目標を2倍としていたが結果は表6に示すように改善できた。

表6 開発速度の結果

プロジェクト	実績
A	約2.5倍
B	約2.0倍

従来レビューで活用していたチェックリストを見直し、コード解析で代用できるものを削減した。その結果、30%以上のチェック項目がコード解析で代用できることが分かり、レビュー時間を大きく削減できた。

(3) 開発工程ごとの専任体制による効果

開発工程ごとに専任者によるコード解析の体制を構築した(図3)ことにより、結合テスト段階で、専任者による高度な解析を行うことができ、ソフトウェアの信頼性を確保できた。

6 まとめ

車載向けECUソフトウェアの開発において、モデルベース開発とコード解析を導入することにより、品質向上と生産性向上の目標を十分達成することができた。具体的には、Statemate®、Rhapsody®、MATLAB®/Simulink®などのモデルベースツールをカスタマイズしTDDと組み合わせて使用することと、QAC®、Polyspace®、Imagix4D、自作ツールなどのコード解析ツールを適材適所に使用することで、従来の人手によるコーディング及び人手による検証に比べ、劇的に開発効率と品質を改善した。また、レビューからコード解析に検証手法を変えたことにより、設計者のスキルに依存しない、安定した検証が短時間でできるようになった。

参考文献

- [1] アルプス電気株式会社 宇治川 尚悟：車載向けECUのコード解析とモデルベース開発への取り組み MATLAB EXPO 2013
- [2] 先進的な設計・検証技術の適用事例報告書 2015年度版 <http://www.ipa.go.jp/sec/reports/20151118.html>

熟練者の知識流通を支える開発環境

株式会社デンソークリエイト 小林 展英
株式会社チェンジビジョン 平鍋 健児

組込みソフトウェア業界にも「つながる世界」が到来し、開発に携わるエンジニアは、つながる先の製品に関する知識も獲得しながら開発を進めることが期待されている。しかしながら、一人のエンジニアの努力だけで獲得できる知識量には限界があり、ほかのエンジニアが獲得した知識を資産化して流通させることが不可欠となりつつある。本稿では、組込みソフトウェア開発に関する知識を概観し、その知識の資産化と流通に用いるモデリング言語とその記述を支える開発環境について紹介する。

1 組込みソフトウェア開発環境が求められる背景

現在の組込みソフトウェア開発では、つながる世界を想定することが当たり前となっている。従来つながりのなかった製品がつながることで、開発の仕方や製品品質に対して異なる文化を持ったエンジニアとの接点生まれ、自分たちの開発では当たり前と考えていた確認作業が履行されないなど予期せぬ問題の発生が予想されている。こうした状況を回避するため、組込みソフトウェアの開発に携わるエンジニアには、自分の開発する組込みソフトウェアが搭載される機器に対する深い知識だけでなく、その機器がつながる様々な機器に対する幅広い知識が求められている。

一方、実際の開発現場では、上述した課題に対して、開発チームを牽引する熟練したエンジニア個人の知識に依存

して対応している状況が少なからず存在している。しかしながら、つながる世界の広がるスピードを考えると、個人の努力だけで対応し続けるのには限界があり、熟練したエンジニアの分析・設計手法や再利用可能なパターンを資産化し、開発チームの誰もが容易に利用できるような流通させることが不可欠となりつつある。図1に筆者が期待するつながる世界の組込みソフトウェア開発における開発環境のイメージを示す。本開発環境が対象とする組込みソフトウェアは、アプリケーション部分とプラットフォーム部分に大別されており、製品共通の機能を実現したプラットフォームを再利用することでエンジニアが開発する範囲をアプリケーション部分のみに限定している。更に、熟練したエンジニアが分析・設計時に用いるモデリングの仕方を「標準設計手法」として資産化し、更に実際のモデリング結果から有用なパターン(ソフトウェア構造や製品に共通して想定されるリスクなど)を抽出して「再利用可能な設計資産」として流用できる環境としている。

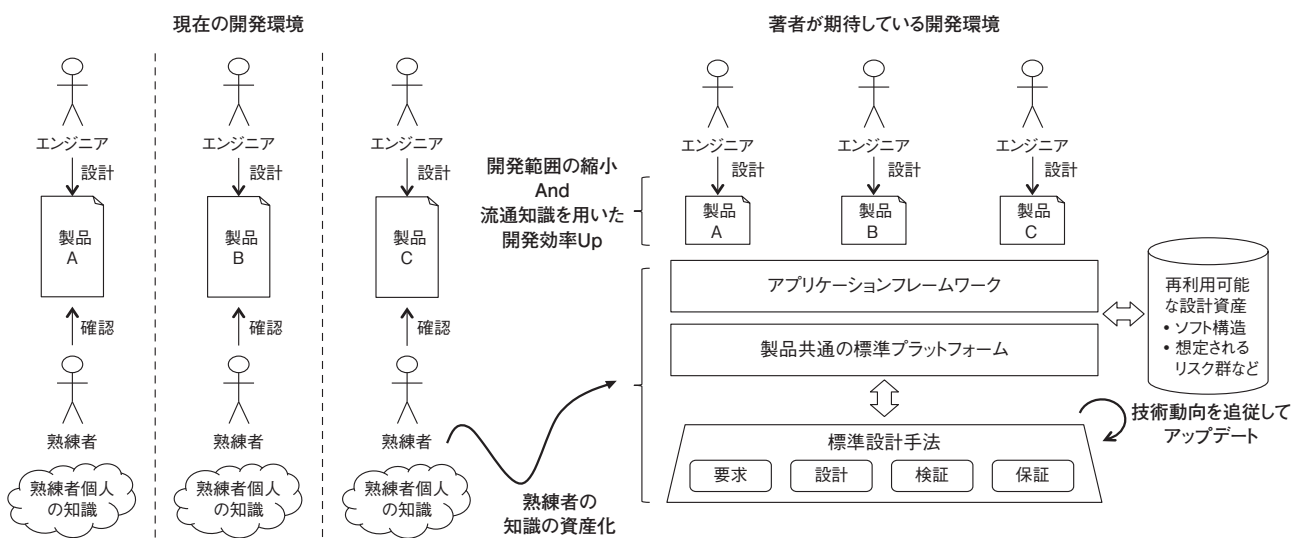


図1 熟練者の知識を活用できる組込みソフトウェア開発環境

次節以降では、つながる世界の組込みソフトウェア開発に携わる熟練エンジニアの知識流通を支えるモデリング言語について2節で概観し、更にその記述を支援する代表的な開発環境について3節で紹介する。

2 組込みソフトウェア開発に用いられるモデリング言語

組込みソフトウェア開発に関連する代表的なモデリング言語を図2に示す。情報システムに搭載されるソフトウェア開発との大きな違いは、開発対象のソフトウェアが搭載される機器に起因する性能要求、安全要求などの非機能要求を満足した上で、実世界の物の挙動を記述した制御モデルの振る舞いをソフトウェアとして実現する点である。

非機能要求を扱うモデリング言語としては、非機能要求を体系的に表現できるNFR Frameworkが存在している。安全要求に対しては、分析手法としてスタンダードなHAZOP、FTA、FMEAに加え、つながる時代に向けて最近注目されている、システム思考に基づく新しい手法としてSTAMP/STPAがある。また、セーフティ要件の分析結果を統合し、システムの安全性の保証のための議論構造を表現する記法としてGSN (Goal Structuring Notation)が考案され、日本ではD-Caseとして拡張されている。そのほか、ハザードや安全要求などのシステムの安全に関する情報を記述できるよう、SysMLを拡張したSafeML、ISO26262などの機能安全におけるシステムの安全設計をアーキテ

クチャ視点から整理し、論じるための記法SCDL (Safety Concept Description Language)なども考案されている。なお、自動車業界を事例に挙げると、これまではセーフティに注力するだけで良かったが、つながる世界の波が自動車業界にも届いた結果、セキュリティに関しても考慮する必要が出てきている。今後、SafeSecのようにセーフティとセキュリティを統合して扱える手法の考案が期待される。

一方、制御モデル設計については、制御対象となるプラントとそのコントローラをモデリングする必要があり、SimulinkやModelicaといったモデリング言語が広く使われている。両者は共に物理世界の数式や制御式を基本としており、本稿では「連続モデル」と呼ぶこととする。また、情報を扱うオブジェクト指向ソフトウェアの一般的な記述から始まったUMLが企業の情報システム構築で使われるようになり、現在では組込みシステム開発でも利用されるようになってきている。特に組込みソフトウェア開発では状態遷移図を中心としたモデルが広く使われており、本稿では「離散モデル」と呼ぶこととする。

最後に、システムズエンジニアリングにもモデルベースが適用され、汎用的に利用できるSysMLや車載ソフトウェア分野でのアーキテクチャ記述言語であるEAST-ADLなども注目されている。今後のシステムズエンジニアリングは、連続モデルと離散モデルの統合、組込みシステムと情報システムの統合、統合された文脈での安全性・セキュリティといったようにつながる世界の広がりに対応していく必要があり、その分野の研究成果としてより良い手法が考案されることを期待している。

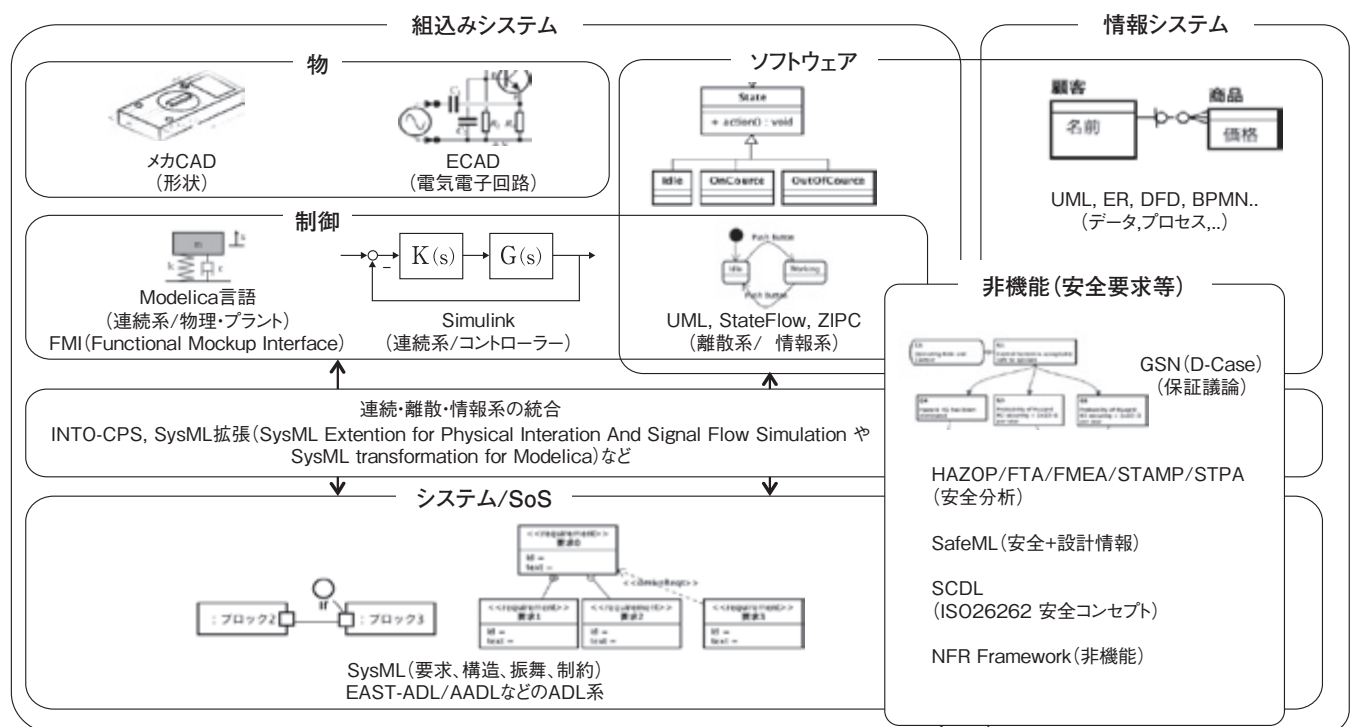


図2 組込みソフトウェア開発に関連するモデリング言語

3 組込みソフトウェア開発環境の現状

本節では、2節で紹介したモデリング言語の記述を支援する開発環境について紹介する。

ソフトウェア開発全般で利用される離散モデルをサポートする開発環境は組込みソフトウェア業界でも浸透しつつあり、astah (Change Vision)、Enterprise Architect (Sparx Systems)などが利用されている。また、状態遷移設計に関してはZIPC (CATS)も利用されている。

一方、組込みソフトウェアの特徴の一つである連続モデルのモデリングにはSimulinkが幅広く利用されており、制御対象の挙動を記述したプラントモデルのモデリングにはModelicaも利用されている。そのほかにも様々な製品が利用されているが、ツールに依存しないモデル交換、統合したシミュレーションを可能とする環境としてFMI (Functional Mockup Interface)が認知されている。近年では、例えばINTO-CPSプロジェクトにおいて、SysMLの拡張言語で離散・連続モデルで構成されるコンポーネントを設計し、その設計情報をFMIの仕組みを利用して現場が

利用する様々な製品に連携させるような取り組みも行われている。また、OMGではSysMLに対する拡張として、SysML Extension for Physical Interaction and Signal Flow Simulation等が提案されている。なお、今後の組込みソフトウェアは、情報システムに搭載されるソフトウェアとの連携もますます増加すると予想される。組込みシステムと情報システムで扱う情報の質は、信頼性、情報の鮮度、セキュリティなどの面で大きく異なるため、双方のシステムが要求する情報の質が相反する可能性も高い。両者を統合したモデリング結果に基づき、適切な解を提示できる開発環境の実現が期待される。

組込みソフトウェアのもう一つの特徴となるシステムの安全性をどのようにして設計に織り込んだか、どのように安全性を担保しているかを表現できる環境については、GSNを記述できるastah GSN (Change Vision)、SCDLに対応したSafilia (GAIO)などのモデリングツールが提供されている(図3、図4参照)。しかしながら、HAZOP、FTAなどの分析手法については、ExcelやVisioなどの汎用ツールを使って記述している場合も多いと予想される。



図3 GSNの例
品質の妥当性を説明するGSN。C (前提)に基づいてG (ゴール)をS (戦略)でサブゴールに分割し、末端にSn (解決)を付けて説明する。

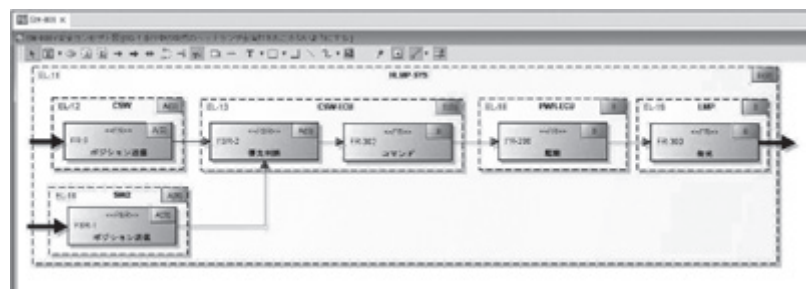


図4 SCDL (Safety Concept Description Language) の例
安全コンセプトを記述するSCDL。エレメント (作りの単位: 矩形点線) の中に要求 (機能の単位: 矩形実線) を配置していく。各要素にはASIL (作りの手厚さレベル: 右肩) が付与され、安全機能や機能間干渉の存在なども明示し、安全コンセプトのアーキテクチャを記述する。

また、完全性やトレーサビリティを重視する組込みソフトウェア開発においては、モデルからソースコードやテストケースなどを自動生成する開発環境に対するニーズは高い。連続モデルからソースコードを自動生成する開発環境としてはTargetLink (dSPACE)、状態遷移図などの離散モデルからソースコードを自動生成する開発環境としては、ZIPC、BrickRobo (富士通コンピュータテクノロジーズ)などが提供されている。テストケースの自動生成に関しては、ソースコードに基づいたテスト設計の支援ツールとしてカバレッジマスター winAMS (GAIO)などが存在しているが、UMLのような表現レベルの高いモデル表現からテストケースを自動生成する開発環境につい

ては今後の発展が期待される領域と予想している。

そのほかの自動生成の事例として、IPA (独立行政法人 情報処理推進機構)のRISE (Research Initiative on Advanced Software Engineering)が公開している保証ケースを自動生成する取り組みを紹介する (<http://www.ipa.go.jp/files/000052723.pdf>)。本取り組みでは、保証対象のシステム構成、システムが満足すべき安全要件、及びシステムに発生し得るリスクに基づいてGSN形式の保証ケースを自動生成する方法を提案している。これらの情報は、前述した開発環境でモデリングすることが可能であり、本提案と組み合わせることで保証ケースの作成時間を大幅に短縮することが期待できる。

最後に、自動車業界が取り組んでいる開発環境の事例について紹介する。車載ソフトウェアも一般の組込みソフトウェアと同様に、大別すると製品固有のアプリケーション部分と製品分野共通のプラットフォーム部分に分けることができる。自動車業界ではAUTOSAR(AUTomotive Open System ARchitecture)が規格化を進めており、プラットフォーム部分については、ECU (Electronic Control Unit) 共通で必要となる機能がAUTOSAR BSW (Basic Software) として定められ、その振る舞いを切り替えるパラメータ

群についても規格化されている。また、AUTOSAR BSWを提供するベンダーは、パラメータを設定できる開発環境を提供しており、それを利用することでECU共通の機能を実現することが可能である。また、アプリケーション部分についても他アプリケーションやプラットフォームとの接続方法が規格化されており、その規格に準拠した開発環境を利用することで、プラットフォームにそのまま搭載可能なアプリケーションを開発することが可能となっている。図5にAUTOSARを用いた開発の流れの概観を示す。

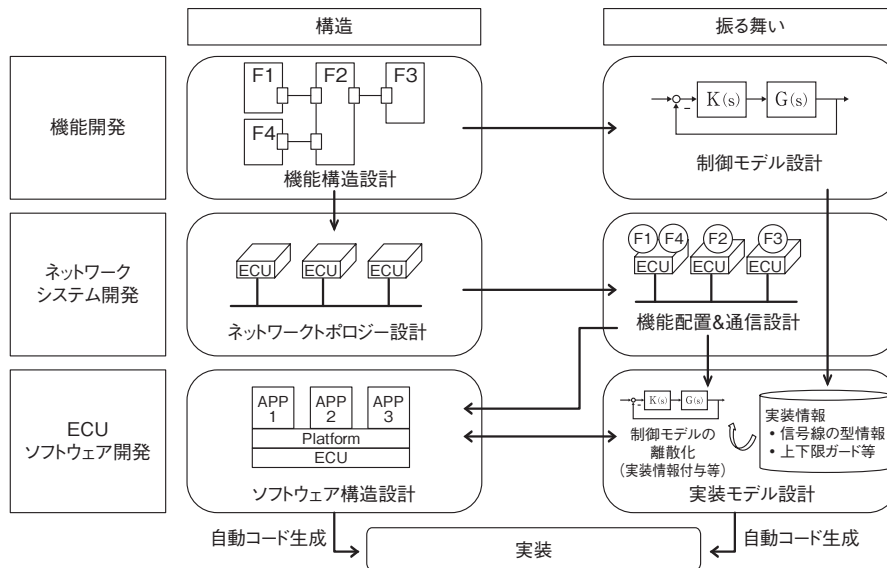


図5 AUTOSARを用いた車載ソフトウェア開発の流れの概観

Vector社が提供する開発環境を事例に挙げると、PREvisionが制御モデル設計を除く機能開発工程とネットワークシステム開発工程を支援し、DaVinciがソフトウェア構造設計を支援している。制御モデル設計についてはSimulink (Mathworks)、実装モデル設計についてはEmbedded Coder (Mathworks)、TargetLink (dSPACE)などが利用されている。そのほか、Mentor社、dSPACE社、Elektrobit社等も自社の得意分野に合わせて開発環境を提供している。これらのツールはすべてAUTOSARが策定したツールチェーン規格に準拠しているため、車両メーカーのニーズに合わせて適切な開発環境を組み合わせて利用することが可能である。これらのツール群は車載ソフトウェアを開発する上で非常に強力であるが、入力しなければならない設計情報が膨大、かつ複雑であり、AUTOSARに関する知見のないエンジニアが利用するにはハードルが高い側面もある。つながる世界において車載ソフトウェアの品質を確保するためには、車載ソフトウェアの構造と振る舞いをつなげる先の製品開発に携わるエンジニアに理解してもらうことが重要になると予想される。他業界のエンジニアと設計成果を共有できる汎用的なUMLツールと車載ソフトウェア開発に特化したツールがそれぞれ

の得意分野で力を発揮し、お互いのモデリング結果をシームレスに連携できる環境の実現が期待される。

4 最後に

本稿では、組込みソフトウェア開発に従事するエンジニアが抱えている課題とその解決を支えるモデリング言語、及び開発環境について概観した。つながる世界を想定することが当たり前となった現代のエンジニアは、一つの機器を熟知すれば良かった時代のエンジニアと比べて、より深く、かつより広い知識に基づいて開発を進める必要に迫られていることを強調したい。こうした現場のエンジニアが抱える窮状を救い、一人一人のエンジニアが自分の製品領域の開発に集中できるようにするためには、様々な分野の熟練したエンジニアの知識が集約され、その知識群が容易に活用できる形で流通していることが不可欠である。つながる世界の組込みソフトウェア開発に携わるエンジニアの一人として、熟練者の知識が容易に活用できる世界が開発環境によって実現されることを期待している。

組み込みとOSSとアーキテクチャ

株式会社アックス 代表取締役 **竹岡 尚三**

1 はじめに

組み込みシステムに使用可能な大規模ソフトウェアで、アーキテクチャや細部が公開されているものは、OSS (Open Source Software) に多いので、本稿では主にOSSについて述べる。

2 人工知能、機械学習

最近、人工知能技術に注目が集まっている。とくに機械学習、深層学習 (Deep Learning) が注目されている。機械学習は、深層学習だけではなく、例えば、Support Vector Machine (SVM) などでも広く使用されている。

これまで、計算機は、人間が現象やデータを解析し、モデル化し、それをプログラムに書き下したものを実行していた。それに対して、機械学習は、人間が解析していない画像などでも、学習用データを多量に用意し、学習させることで、未知のデータを判定できるようになる。人間が解析やプログラミングできないものを対象に、判定などを行ってくれる人工知能は、これまで人間しかできなかった知的作業の一部を肩代わりしてくれるものと、大きく期待されている。

ニューラルネットワークのような人間の脳を模した機械学習自体は、新しいものではなく、1950年代末期に、パーセプトロンが発表され、学習&判定能力があることが分かっていた。続いて、多層パーセプトロンが発表され、それは既にバックプロパゲーション (誤差逆伝播法) を用いて学習を行っていた。1980年代後半にもニューラルネットワーク・ブームが起こり、1990年前後には、日本でも「ニューロ家電」や「ニューロ・ファジー家電」がたくさん作られた。これは、学習結果がROM化され、当時でも低性能なシロモノ家電用マイコンで実行された。その頃には、遺伝的アルゴリズム (Genetic Algorithm) や人工生命もブームとなり、機械が自分自身で学習し、生命維持の本能を獲得する、などの研究も多かった。機械学習は、最近に、目覚ましい進歩を遂げたわけではなく、例えば、国際電気通信基礎技術研究所 (ATR) 脳情報通信総合研究所所長の

川人光男氏は、「1980年代の第2次ニューロブームのときのことを思い出さざるを得ない (中略) 今の人工知能技術の基礎はいずれも当時出てきたもので、そこから本質的な飛躍はない」という主旨の発言をしている。^{*1}

深層学習は、最近のGPGPU (汎用GPU) の出現により、小規模なハードウェア・システムでも実用的に多数のコネクションの計算ができるようになったため、急激に注目を集めるようになった。しかし、GPUほどの演算能力のない、直近の組み込みCPUで扱うには、深層学習は負荷が大きい。

深層学習が注目される直前までの機械学習は、SVMがよく使用されていた。SVMは、ARM11程度のCPUでも十分に高速に動作し、現行の組み込みシステムでも、即座に実用に供せる。弊社株式会社アックスでは、SVMを組み込みに使用した実例として、顔認識を行い、顔を追尾する自走ロボットを試作した。このロボットは、Raspberry Pi (初代) を使用し、CPUはARM11 700MHzである。^{*2}

機械学習は、学習のためのデータを用意して、学習を行わせる。一般に深層学習でもSVMでも、学習にはかなりの演算能力が必要で、通常のx86 PCで行っても、かなりの時間がかかる。深層学習はSVMなどに比べると、非常に演算が多いため、GPGPUの使用が望まれる。通常の深層学習は学習結果のデータの解析が不可能、若しくは困難で、学習させるためのデータをカット・アンド・トライで選択するということが行われている。深層学習のパラメータ調整と、学習データの選択は、ノウハウの固まりである。とくに学習データの選択方法は、今後も工学的アプローチがあまりできないと思われる。SVMは、学習結果はサポート・ベクトルとして得られ、判定対象もベクトルとして表現されるので、解析は可能である。しかし、ベクトルの各要素は、学習時に自動的に決定されるので、人間にとって直感的な解析が容易なわけではない。用途によるが、一般的に、機械学習は、学習結果が96%程度よりも高い精度を持たないと、実用的ではないと言われることが多い。

OSSで公開されている機械学習の多くは、学習コマンドと、認識サブルーチンなどから構成されている。つい最近は、Pythonから呼び出せる形式のものが多い。使用するには、学習データを用意し、学習コマンドを使用して学習を行わせる。得られた学習結果を、認識サブルーチンなどに設定して、認識対象をもって認識ルーチンを呼

び出せば、結果が得られる。認識をサブルーチン呼び出し形式ではなく、バッチ処理で行えるコマンドが用意されていることも多い。機械学習の内容など知らなくとも、機械学習を利用することは可能で、OSSの機械学習を利用することは、かなり容易である。

機械学習は、原理上、学習したことのないデータを入力された場合、何を出力するか分からない。認識率が99%であっても、1%の出力がとんでもない誤りの場合がある。そのため、力の大きな装置などへ機械学習を組み込む場合には、誤った出力が、機器を動かしてしまわないように、機械学習ではない機構（例えば、伝統的な通常プログラムによる動作抑止ルーチン）による安全策を講じておく必要がある。

機械学習は、いわゆる人工知能で、ロボットの頭脳として、非技術者には、何でもできると思われるようである。しかしながら、機械学習は画像認識などには力を発揮するが、不向きな用途もある。弊社では、ルールベースの推論システムと機械学習を組み合わせたハイブリッドなAI「ごまめ」を開発し、使用している。

3 最近の組み込みアーキテクチャ

旧来からの組み込みシステムは、リアルタイム・モニタや小さなOSカーネルに、いわゆるミドルウェアなどの層を積んでいく形式が多かった。例えば、スマートフォンOSであるAndroidなどは、Linuxカーネルに、ミドルウェアの層を積んでいく形になっている。

だが、最近の大規模な組み込みシステムでは、少し様相が変わってきた。例えば、1980年代より「ソフトウェア・バス」というような言葉で言われたものが、現実になってきている。ここで言う「ソフトウェア・バス」とはソフトウェアのモジュール同士をつなぐ役目だけを果たすミドルウェアの一種であり、部品となるソフトウェア・モジュールには、インターフェースの切り口を提供する。各ソフトウェア・モジュールは共通のインターフェースを使用して、機能を提供する。名称は変わりつつも、ソフトウェア・バスの構想は、何度も現れては、消えていった。

その理由の幾つかは、ソフトウェア・バス自身がメモリを消費する、モジュール間の通信コストが有意な負荷となる、(旧来の)組み込みシステムでは、さほど大きなソフトウェア・モジュールが多くなかったハードウェア進化が早く、ソフトウェア資産の陳腐化が早過ぎたなどであろうと思われる。

しかし、現在は、ハードウェア資源(CPU性能、メモリ)が潤沢になり、同時に大規模なソフトウェアが増え、その資産を活かしながら、新しいソフトウェア・モジュールを加えていくようにしたいという時代になってきた。

また、情報処理アーキテクチャとしては、前処理-処理-後処理のように、情報/データの流れとして捉え、前処理や後処理は既存のものを使用し、ここでいう「処理」部分に、自分の作成したモジュールを差し込みたい、という要請がある。

とくに、ロボット分野では、それが成功しており、「ロボット・ミドルウェア」と呼ばれる、国立研究開発法人 産業技術総合研究所(産総研)のRTM(Robotics Technology Middleware)や、ROS (Robot Operating System)が多くのシステムで採用されている。

4 ロボット・ミドルウェア

ロボット・ミドルウェアは、ロボットの内部で、各ソフトウェア・モジュールをつなぐためのフレームワークであり、次のような要請がある。^{*3}

- コンポーネントをつなぐ
- コンポーネントの独立性を高めさせる
- オーバーヘッドなし
- 存在を感じさせない

産総研のRTM (OSS版はOpenRTM-aist)は、オブジェクト通信の組み込み版であり、RTMを使用する各モジュールは、RTC(RT Component)と呼ばれる。RTC間のインターフェースが、OMG (Object Management Group)において2008年に国際規格となり公開されている [図1]。

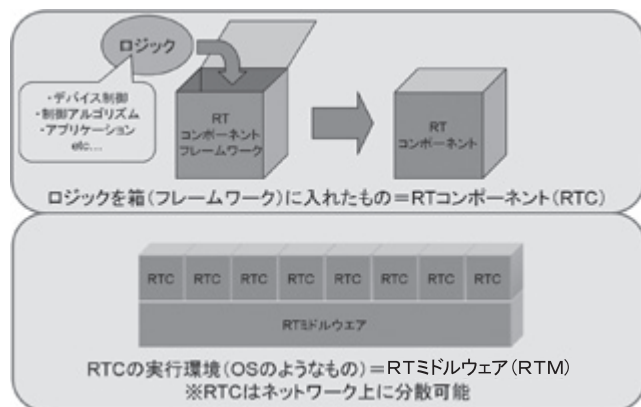


図1 RTコンポーネントとRTミドルウェア^{*4}

ROSは、Google社の自動運転に採用され、著名になったロボット・ミドルウェアである。

ROS自体は、モジュール間の通信を行うソフトウェアであるが、ROSパッケージとして配布されている。

ソフトウェアには、ロボット開発に有用な大規模ソフトウェアがたくさん含まれている。

RTM、ROSとも、存在を感じさせない通信ミドルウェアとなっており、ユーザは、ソフトウェア・モジュールをつなぎ合わせて、所望のシステムが構築できるようになっている。RTMであれば、カメラなどの視覚モジュール、マイクが入った聴覚モジュール、モータなどの運動モジュールを組み合わせてロボットを構成する [図2]。

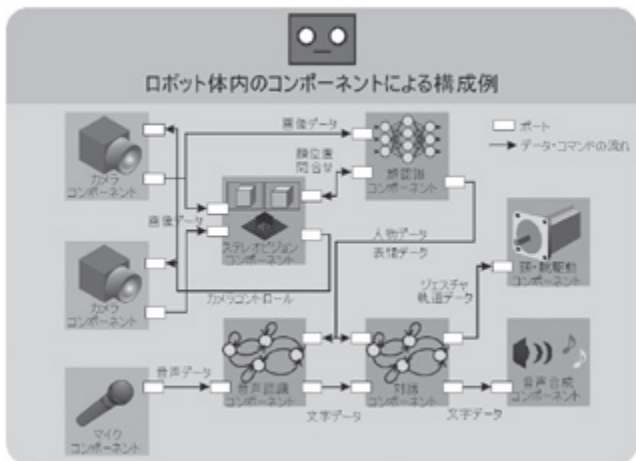


図2 RTコンポーネントによる構成例^{※4}

ROSは、データの処理の流れに合わせて、ソフトウェア・モジュールをつなぎ合わせるというアーキテクチャとなっている。ROSには、知覚、物体認知、セグメンテーションと認識、顔認識、ジェスチャ認識、動作追跡、ステレオビジョン：2台のカメラを通して、奥行き知覚など多くのソフトウェア・モジュールが存在している [図3]。

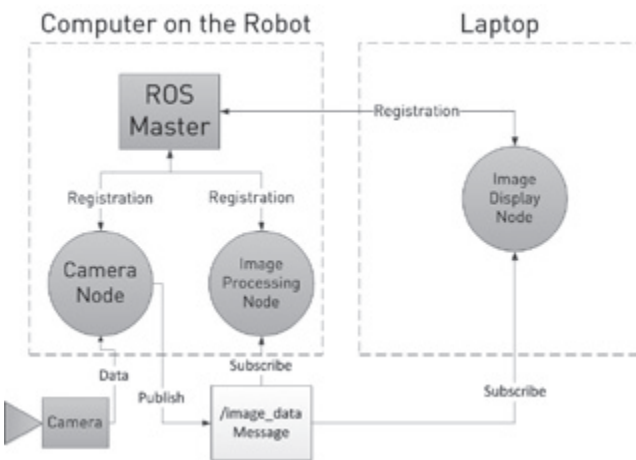


図3 ROS: データの流れに応じて、コンポーネントをつなぐ^{※5}

また、それより低層ではOpenELというハードウェア抽象化層が組込みシステム技術協会 (JASA) などを中心とし

て開発され、同時に国際標準化の活動を行っている [図4]。OpenELにはOSS実装も存在し、配布される予定である。

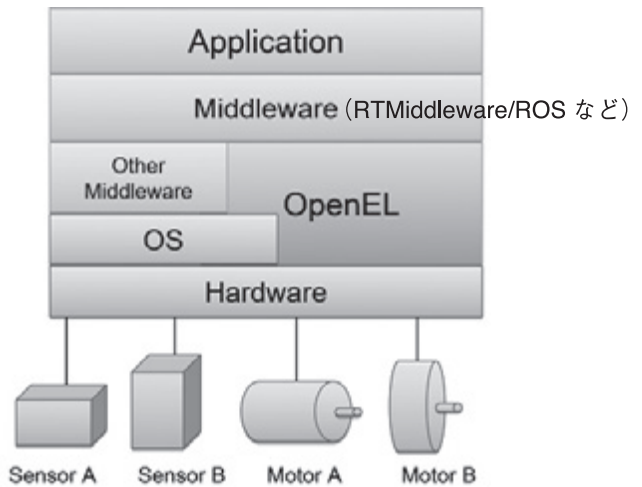


図4 OpenELの位置付け^{※6}

5 自動運転、ROS

自動運転は、ロボットの一種である。株式会社アックスは、日本で最も先進的な研究の一つである加藤真平先生 (東京大学/名古屋大学) のプロジェクトにかかわっている。また、名古屋大学発ベンチャーの株式会社ティアフォーは、そのプロジェクトの成果物であるAutowareの発展を推進している。

自動運転技術には多数の要素がある。Autowareは、Googleの自動運転と同様にROSを元にして、ROSのコンポーネントを開発することで、システムを作り上げている [図5]。

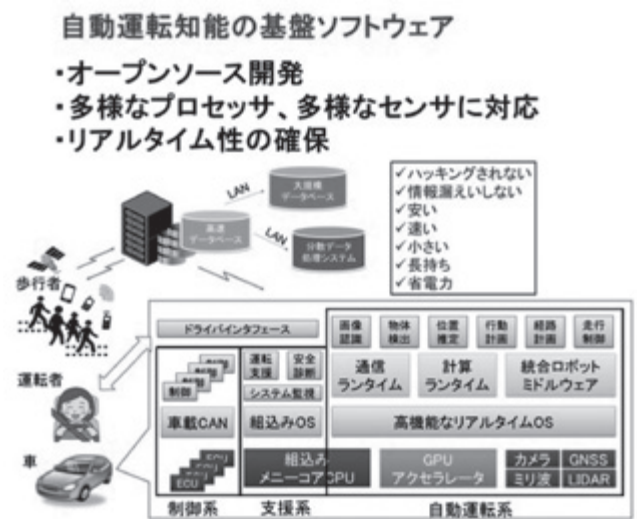


図5 自動運転OSS Autoware 構成・特徴^{※7}

ROSには、そもそも、ロボット開発に必要な、大規模なミドルウェアやコンポーネントが相当数そろっている。そ

の中には、画像認識、立体視、ジェスチャー理解、立体構造理解などを容易に実現できるコンポーネントなどが含まれている。ROSのコンポーネントは、データフローに基づいて組み合わせるようになっている。独自に開発したコンポーネントも、既存のコンポーネントと組み合わせ、一連のデータフロー処理の中に容易に組み込める。

自動運転には、三次元地図、自車がどこに居るかを認識する、信号を認識するなどの技術が必要である。それらに加えて、路上の障害物や、実時間で道路を認識することで、安全に走行することができる。

6 ドローン

最近、ドローンについても注目が集まっている。OSS界では、LinuxFoundationが、ドローンに関するプロジェクトを取りまとめて“Dronecode”に集約した。

Dronecodeの回りでは、ハードウェアもデファクト・スタンダード化が進んでおり、基板のサイズやコネクタの位置が標準化され、ドローンの機体に数種類の基板を交換して搭載することが可能になっている。

Dronecodeは、機体に搭載する(オンボード)のソフトウェア[図6]と、遠隔操作する側(オフボード)のソフトウェア[図7]に大きく分かれている。

オンボードのソフトウェアは、実時間性が高く、回転翼を制御し、バランスを取ってドローンを飛ばすソフトウェアと、Wi-Fiからの指令を受けるソフトウェアなどから成っている。

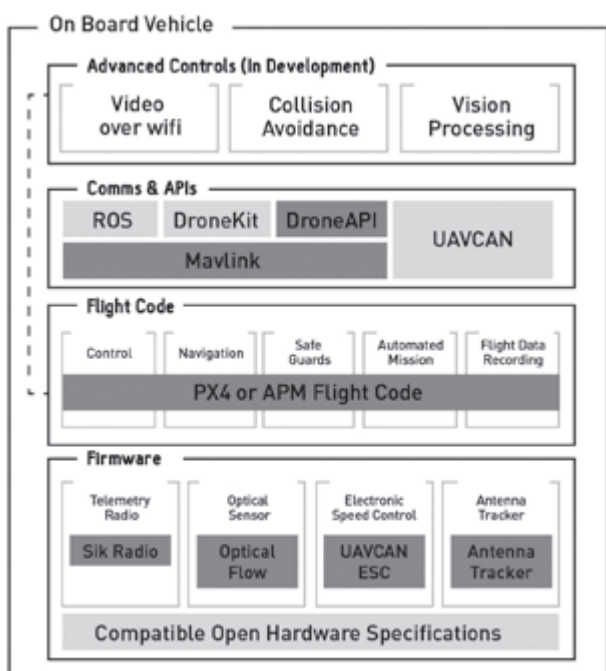


図6 Dronecodeのオンボード・ソフトウェア^{※8}

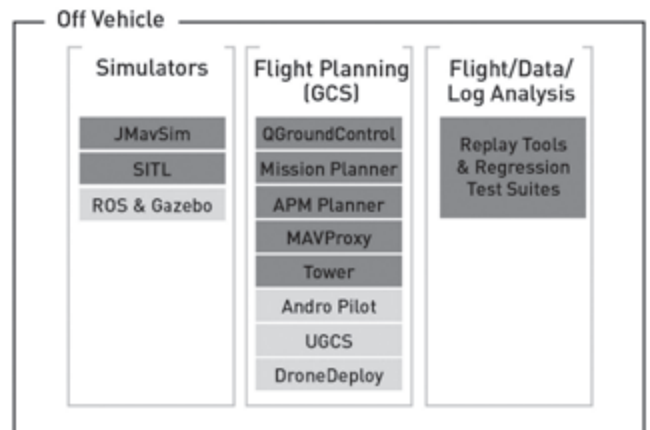


図7 Dronecodeのオフボード・ソフトウェア^{※8}

オフボードのソフトウェアは、Linuxなどで構成されており、スティックなどの入力装置を読み取り、Wi-Fiなどを経由して、ドローンへ送信する。

日本の通常のドローンは、法令により、目視可能な範囲しか飛行できない。よって、通信方式はWi-Fiが主流である。

7 まとめ

このように、現在、注目を集めている技術の多くは大規模なOSSとして公開されており、それを元に試作を行い、技術検証を行うことは、容易で有意義である。また、用途によっては、既存OSSをそのまま組み込んで実用製品にできる。

しかし、生命にかかわるようなクリティカルな分野では、OSSの品質をどう捉えるべきか、など、既存のOSSを実際の製品の中に組み込んでいくことには、まだまだ、取り組むべき問題が多い。

それらの問題を踏まえた上で、組込みOSSを大いに活用していき、また、日本からOSSを発信していきたい。

【脚注】

- ※1 引用元：<http://pc.watch.impress.co.jp/docs/column/kyokai/723879.html>
- ※2 日経Linux 2014年1月号，“Raspberry Piで作る、顔認識でモジモジ近づく自走ロボット”，竹岡尚三
- ※3 引用元：http://en.wikipedia.org/wiki/Robotics_middleware (左記より引用し和訳)
- ※4 引用元：<http://www.openrtm.org/openrtm/ja/content/rt%E3%83%9F%E3%83%89%E3%83%AB%E3%82%A6%E3%82%A8%E3%82%A2%E3%81%A8%E3%81%AF%EF%BC%9F-0>
- ※5 引用元：<http://www.clearpathrobotics.com/blog/how-to-guide-ros-101/>
- ※6 引用元：http://jasa.or.jp/openel/Main_Page/ja
- ※7 引用元：<http://www.pdsl.jp/fot/autoware/>
- ※8 引用元：http://www.dronecode.org/sites/dronecode/files/pages/images/dronecode_figure1_v3_ac-01.jpg (左記より引用し分割)

組込み技術者育成へ初等中等教育からの改善

株式会社オプテック 代表取締役会長
 一般社団法人スキルマネジメント協会 理事長
 東海大学名誉教授 **大原 茂之**

IoTで世界をリードする成否は組込み技術者の存在が大きいが、少子化で労働者人口不足が加速している。その対策は人材の質向上となるが、実は初等中等学校時代に刷り込まれた知識が質向上の阻害要因となっている。本稿ではIoT時代の組込み技術者育成に向けた初等中等教育からの見直しとETSSの活用について述べる。

1 はじめに

IoT、AI、ロボットなどの活用による第4次産業革命が進行する中で、組込みソフトウェア開発力はその成否を左右するコア技術であると言って良い。組込みソフトウェア技術を発展させ、イノベーションを起こしていくことが日本の競争力を高める必須の条件である。こうしたことを継続的に実践していくためには組込みソフトウェア技術者の量と質を高めていくことと、人材を育成及び活用する環境についてマネジメントしていく必要がある。

しかし、こうした人材の量と質の面それぞれについて見直すべき問題が存在し、環境について検討すべき問題が存在する。とくに、人材の量的側面の壁は少子高齢化の問題であり、経済面での足枷にもなっている。もう一つの人材の質的側面においては、初等中等教育時点からの教育内容の問題がある。この問題は根が深く、理工学系の大学生や社会人になってからもこの問題の影響が残っている。初等中等教育を改善し組込みソフトウェア技術者の質的向上を進める取り組みについて、一般社団法人スキルマネジメント協会の活動の紹介を含めて提言する。

2 人材の量と質を高める上での少子高齢化の問題

日本の技術力を世界トップへもっていかうとするとときに、大きな壁となるのが少子高齢化の問題である。図1に少子高齢化の年次推移を示す^{1) 2)}。1950年～2014年までが実際の人口推移で、2015年以降は推計である。この図から以下のことが理解できよう。

口推移で、2015年以降は推計である。この図から以下のことが理解できよう。

- ① ほぼ20年前に65歳以上の人口が14歳以下の人口を上回り少子高齢化の時代に突入し、その後は両者の差は広がる一方である。
- ② GDPを押し上げる労働者人口も20年前から減少が止まらない状態になっている。
- ③ 一方で、14歳以下と65歳以上を加えた非労働者人口は増加の一途である。
- ④ 2060年にはほぼ労働者一人で非労働者一人を支えることになる。2060年は来る東京オリンピックの年から40年後である。
- ⑤ グラフには含まれていないが、18歳人口も減少の一途をたどり、2018年から急激に減少を始めるのである。13年後の2031年には大学への進学者数は約17万人減となり、100校以上の大学が経営危機に陥ると予想されている。このことは、大学関係者の間では2018年問題として認識されているが、2022年以降は企業も新卒者獲得の氷河期に突入することになる。

少子高齢化という人口動態の問題を劇的に解決することは極めて困難である。総人口に関しては、国立社会保障・人口問題研究所の将来推計人口データベースを参考にすると、2005年に1億3千万人弱であった日本の総人口は、出生率2.5の場合で2050年に1億1千万人弱となり、出生率1.29の場合には2050年に9千万人弱まで減少する。

人口の量的問題の解決に関しては政策的な戦略に属するものであり、技術者育成の観点からは質的向上策を講じることになる。ただし、この解決に向けては初等中等教育の段階まで戻って取り組む必要がある³⁾。

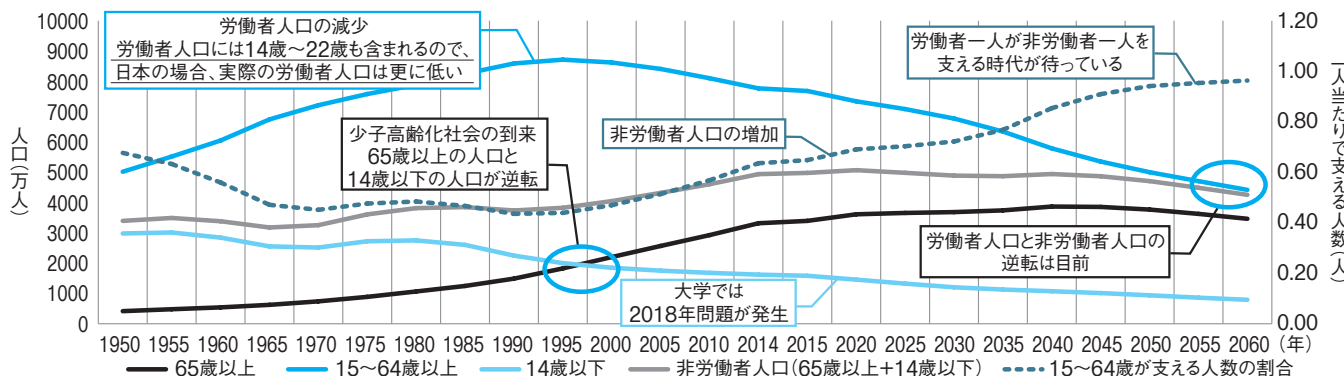


図1 加速する少子高齢化

3 組込み技術者育成の ボディーブローとなる初等中等教育の 問題点

組込み技術者が対象とする専門技術領域は、システム設計、コンピュータアーキテクチャ、ソフトウェア、ネットワーク、電気、機械などであり、基礎技術領域は物理、化学、生物などの理工学系の領域となる。組込み技術者を育成するには理工学系の基礎中の基礎としての数の扱い、演算、関数、電気などの知識が正しく修得されていることが前提となる。しかし、電気系や情報系の学生で2進数における1の補数や2の補数を正しく理解している人数は極めて少ない³⁾。その原因は初等中等学校時代の教育内容が不十分な状態で終わっていることにある。このことは、組込み技術者の育成だけではなく、広く人材育成にかかわるものである。先端的な人材育成を考える前に、まずはこの問題を明らかにする。

3.1 論理的思考を阻害する引き算の内容

学生から社会人まで恐らくは大半の人達が勘違いし、矛盾に気づいていない引き算の問題がある。

例えば、42-17の引き算は、次のように進めるであろう。

- ① 2から7は引けないので、10の位の4から10を借り、4を3に修正する。
- ② 借りてきた10から7を引いて3を得る。
- ③ この3と2を足して1の位の答え5を得る。
- ④ 10の位では3から1を引いて2を得て、最終的な引き算の答え25を得る。

この手順は小学校の教科書に出ているものであり、すべての人の頭の中に刷り込まれているであろう。この引き算の手順は常識となり、その矛盾に気づけないのである。矛盾点は以下の通りである。

- ① 矛盾1：1の位に10を持ってきている。
1の位には0から9までしか入らないことを知っていながらこのような手続きを踏んでいる。論理的には、9と1に分解して持ってくる必要がある。
- ② 矛盾2：上記の矛盾1に目をつぶったとしても、借りてきた10から7を引いている。
2が小さくて7を引けなかったので10を借りてきても、10の1の位は0であり、0から7を引くことはできない。それにもかかわらず、10から7を引くという強引な押し付けを行っていることになる。これでは論理的思考力の育成はできない。正しくは、借りてきた10から7を引こうとしても、10の0から7は引けないので、再び10を借りてくることになり、これを永久に繰り返すことになる。この計算を終わらせるためにはどうしたら良いのかを考えさせるのである。そうすることで、暗黙のうちにアルゴリズムの概念を教えることになる。

小学校の段階から導入されるプログラミング教育で狙う論理的な思考力や問題解決力の育成を成功させるためには、

解き方は暗記するものではなく、考え、直観し、創造していくものだという意識づけを暗黙の内にしていくことである。学生が、1の補数、2の補数の習得を暗記に頼っているのは、小学校での論理を無視した解き方の暗記という刷り込みが影響している可能性が高い。こうした個々の事例は大人になった後でも修正できるものの、人材育成の観点からは論理性と直観力を鍛える芽を摘んでしまっている可能性がある。組込み技術者に限らず人材育成を成功させるには、こうした長期的視点からも見直していく必要がある。

3.2 組込み技術者育成を阻害するその他の要因

組込み技術者を育成する上での阻害要因は引き算以外にも存在する。以下の知識は初等中等教育の段階で出てくるものであるが、定義が曖昧のため正しく理解されていない例である。

- (1) 関数の定義： $y=f(x)$ において、どれが関数であるかを指摘できない。
- (2) 電気：「電圧が大きくなると電流は増加する」という命題の矛盾を指摘できず、電圧と電流は互いに独立であるという関係も説明できない。
- (3) 回路記号：電圧源の記号を電池、ショートを導線だと錯覚している。

こうした状況は組込み技術者を育成する上で大きなブレーキになると言わざるを得ない。小学校からの刷り込みによって、思考する前に解法としての手続きを暗記し、その手続きを実行できるように勉強するほうが試験の点数が良くなるという習慣ができてきている可能性がある。手続きの暗記とその作業効率を高める習慣は、解法や事例がないと何もできない状態に陥ることになる。モデリングやモデルベースの技術を習得する場合でも、ツールを使えばモデルベースを理解できたという錯覚に陥ってしまう。ツールや事例に頼ることは過去の枠に囚われていることになる。発想力、矛盾や問題の発見力、問題解決力、システムの構想力、イノベーションの創造力などを持ち、知的労働生産性を高める人材を育てるには、こうした初等中等教育からの見直しをする必要がある。

SMA(スキルマネージメント協会)では、初等中等教育を含めて組織の中で人材育成と活用の弊害となっている習慣、常識あるいは文化などの環境の問題を発見することと、それによる環境を改善していく研究に取り組んでいるところである。

4 新技術による イノベーション創造への考え方

IoTやAIによるイノベーションが進行中であるが、これらの技術をこれまでのものづくりのアウトプット指向という文化の枠組みで捉えると時代に取り残される危険がある。IoTやAIは図2に示すような利用者側でのアウトカム指向の文化になると考えるべきである。

- 開発・製造がこれまでのものづくりのフェーズであり、エッジ(物理的な製品)をアウトプットしている。エッジとはセンサ、アクチュエータあるいはカメラなどの製品である。
- これらのエッジはフォグ化(ソフトウェアでパッキング)されてサイバー空間の要素となる。サイバー空間と実空間の間で様々な接続が自由に行えるようになる。ただし、実空間の要素のすべてがサイバー空間につながるわけではない。
- この接続された集合体を活用して利用者は利用者自身が描く夢若しくはアウトカムを実現するようにシステムを構想できるようになる。
- この利用者が望むアウトカムを実現するプロセス(技術)を提供するサービスがIoTの利用価値を高める。

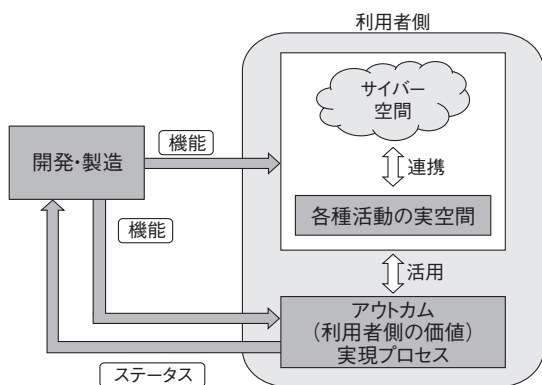


図2 IoTとアウトカムの実現プロセス

このようにIoTにおいてはアウトカム指向でシステムを考えていくべきであるが、IoTを活用するにはエッジ、エッジフォグを設計若しくは活用でき、物理空間のシステムを構築し、更にサイバー空間と連携させる技術やスキルは組込み技術者の得意分野でもある。従って、これから

の組込み技術者はアウトカムと要求仕様を設計できるようになっていく必要がある。

5 組込み技術者育成と活用のための環境について

ここでは、IoTに関してSMAが取り組んでいる人材育成と環境の関係について述べる。

5.1 ETSSの改良版

ETSS (組込みスキル標準)の分かりにくい構造を改良した構造を図3及び図4に示す。

ETSSはフレームが標準であり、記入する内容は各社各組織において定義することが基本である。ただし、その内容を定義する場合、IPAが提供するiCDを活用すると効率良く作業を進めることができる。以下、ETSSのフレームについて説明する。

- 技術/知識のシートは階層化して記入する構造である。技術/知識の分野数は任意であり図では3分野までを示している。
- 分野を詳細化する粒度についても任意であり、図では第3階層までを示している。
- スキルレベルを記入するスキルレベルシートはほとんどの場合、レベル1からレベル3までを使うことになる。
- レベル4は新たな技術/知識の創造が期待できる場合にチェックする場合と、実際に新たな技術/知識を創造した場合はETSSのシートそのものを修正する場合がある。

5.2 スキルレベルの定義

スキルレベルの定義を図4に示す。

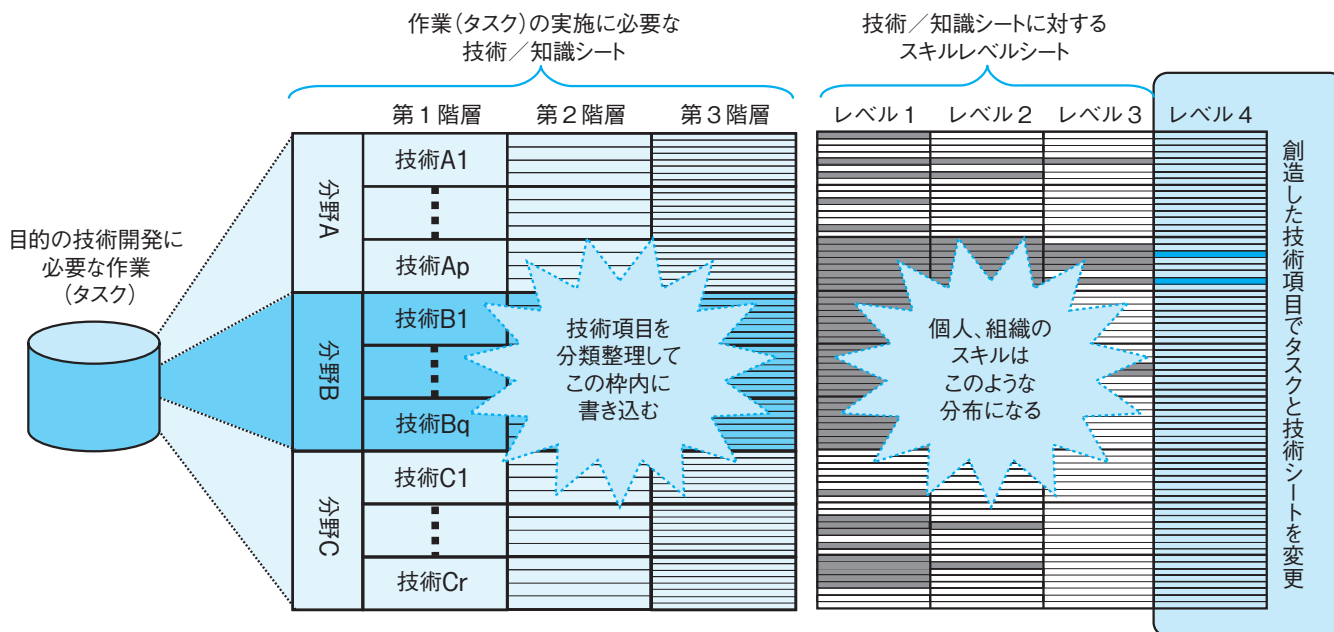


図3 ETSS(組込みスキル標準)のフレームワーク

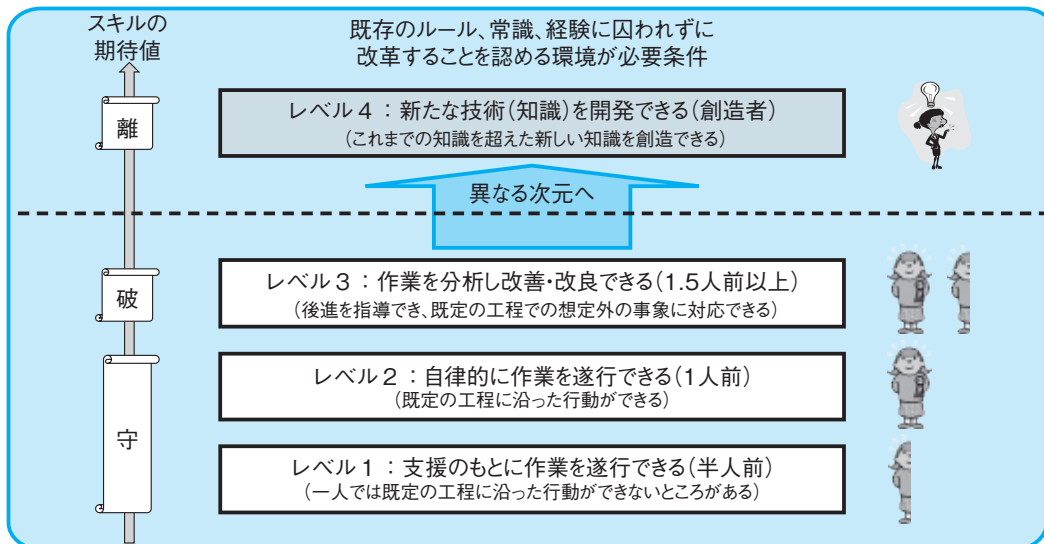


図4 スキルの期待値と育成・活用の環境

ETSSのスキルレベルは期待値であり、過去の実績による評価ではない。ETSSのスキルレベルの思想は、日本の伝統的な人材育成指標である守破離に基づいている。守破離のゴールはイノベーションを起こせる人材を育成することである。教えた師を超えて、その流派を超える新しい流派を打ち立てる可能性を持つ弟子を育成することが師のミッションである。このミッションを達成するために徹底的に洗練した所作の型を生み出すのである。この守破離の思想を実践するには、誰もが離というクリエイティブな方向を理解し、互いに切磋琢磨できる環境が必要条件となる。伝統的な守破離の観点からすると、最初に述べた初等中等教育という守の段階の内容を正しい型に修正していく必要がある。

5.3 IoTへ向けたETSSの応用

ここではSMAで研究を進めているIoT技術者を育成するための技術／知識シートの作成方法について紹介する。

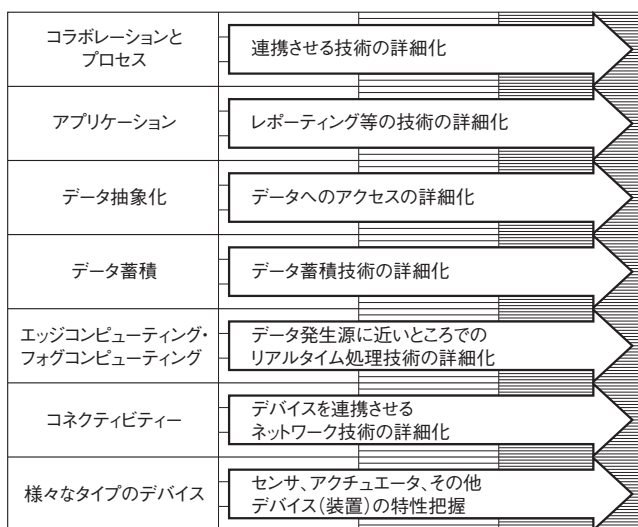


図5 シスコの7階層モデルを参考にしたIoT向け技術／知識シートの構造

図5はIoTに関するシスコの7階層モデルをETSSにマッピングした場合の例である。7階層がそのまま7つの技術／知識の分野として記入してある。この技術／知識項目に対応できる人材を育成するには、図に示すようにデバイス、ネットワーク、エッジコンピューティング、データ蓄積などの技術を更に詳細化していくことになる。IoTの各技術分野の詳細については、JASA(組込みシステム技術協会)とも連携して研究を進めており、近いうちに技術項目の詳細を明らかにできると考えている。

6 おわりに

本稿ではIoTやAIを推進する主役とも言える組込み技術者の育成を中心に検討を行った。人材育成を行うには、少子高齢化という大きな壁があり人材の質を飛躍的に高めることが必須の条件であること、長期的には初等中等教育から見直す必要があること、そして人材の育成と活用のためには、日本の文化を通して個々人の中に無意識に刷り込まれている「改善改良を通してイノベーションを起こしていく」という、伝統的な守破離に倣うことが重要であることを述べた。更にIoTやAIの活用はこれまでのものづくりのようなアウトプット指向ではなく、ユーザ主体のアウトカム指向で取り組むべきであることについて述べた。最後に、SMAで取り組んでいるETSS活用によるIoTの定義などについて紹介した。

参考文献

- 1) [総務省平成27年版] 総務省統計局 人口の推移と将来人口 <http://www.stat.go.jp/data/nihon/02.htm>
- 2) 「内閣府」推計統計人口でみる50年後の日本 http://www8.cao.go.jp/kourei/whitepaper/w-2013/zenbun/s1_1_1_02.html
- 3) [J.of JSEE] 大原茂之：工学教育 第64巻 第3号、工学教育推進に向けた基礎教育改善への提言、公益社団法人日本工学教育協会、PP.30-36、2016

Embedded Technology West 2016 (ETWest2016) 出展報告

SECプロモーショングループ 主任 荒川 明夫

IPA/SECは、2016年7月7日、8日の2日間、グランフロント大阪コングレコンベンションセンターにて開催された「Embedded Technology West 2016 / 組込み総合技術展 関西 (ETWest2016)」に出展した。また、展示会隣接会場にて、IPA特別講演及びIPAセッション各2枠、計4講演を実施した。

1 展示会概要

Embedded Technology West (ETWest)とは、一般社団法人組込みシステム技術協会 (JASA)が主催する西日本で唯一となる最新テクノロジーの専門技術展であり、組込みシステム開発にかかわる技術者や開発者向けに最新技術などの情報を発信しており、今回で11回目の開催となる。また、今回よりIoT Technology West が併催され、両展示会の共通テーマとして「“ものづくり”がわかる、“IoT”がわかる」を掲げ、開催された。

2 出展概要



IPA/SECは、事業成果の普及・啓発を目的として、本展示会に毎年出展している。

今年は、IPA展示ブースでのブースプレゼンを2日間で26回、SEC先端技術入門ゼミブースでの講義を2日間で12回、IPA特別講演を2日間で2回、IPAセッションを2日間で2回、計42回のセッションを実施した。

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

3 IPA展示ブース



本出展では、展示ブースでブースプレゼン・パネル展示・資料配布などを実施し、隣接した小間にSEC先端技術入門ゼミブースを構え、運営した。

展示コーナーでは、障害対策・再発防止・未然防止をテー

マに「はじめてのSTAMP/STPA～システム思考に基づく新しい安全性解析手法～」「社会インフラのシステムトラブルを防ぐ～情報処理システム高信頼化教訓集～」のパネルを配置、つながる世界の開発指針をテーマにしたゾーンでは、「IoT開発者必見!! 国内初の『つながる世界の開発指針』」「『つながる世界の開発指針』を実現する対策技術」を配置、品質向上をテーマにしたゾーンでは、「品質改善に取り組みば生産性もアップ!」のパネルをそれぞれ掲出した。そのほか、「組込みソフトウェア開発の組織力を高める 組込みソフトウェア開発データ白書」「ソフトウェアの品質向上を目指して」「IoT開発におけるセキュリティ設計の手引」「情報セキュリティマネジメント試験」などのパネルを展示した。

「つながる世界の開発指針」や「組込みソフトウェア開発データ白書」への関心が高く、関連したブースプレゼンを聴講する来場者の姿が多く見受けられた。

4 SEC先端技術入門ゼミ



IPA展示ブースと隣接設置したSEC先端技術入門ゼミブースでは、本出展の検討会で選出された外部有識者による初心者向けのチュートリアルを昨年に引き続き実施した。

ソフトウェア開発などの先端技術や取り組みを解説するSEC先端技術入門ゼミの今年の演目は、「Ruby入門」「CEP入門」「HEMSの現実と勘所」「サイバーセキュリティ入門」「初歩から学ぶアジャイル開発」「システムズエンジニアリング入門」を1日目に、2日目は、「要求分析の勘所」「セキュリティ設計(リスク分析)入門」「はじめてのSTAMP/STPA」「セーフティ&セキュリティ by デザイン入門」「定数管理のススメ」「『つながる世界の開発指針』の解説」と

いった内容で、どの講義も非常に多くの来場者が聴講し、2日間で536名が参加した。

セミナーやカンファレンスと異なり、参加者が直接、講師に質問を投げられるこの企画は、参加者に好評であり、アンケートには「知りたかった技術が短時間で理解できた」「もっといろんなテーマを企画して欲しい」「IoT時代における変化が理解できた」という意見が寄せられた。

5 IPA特別講演



グランフロント大阪内にあるインターコンチネンタルホテル大阪2F HINOKIでは、IPA特別講演を実施した。ご登壇いただいた方は、1日目に慶應義塾大学大学院システムデザイン・マネジメント(SDM)研究科 准教授の白坂成功氏、2日目は、名古屋大学大学院 未来社会創造機構教授の高田 広章氏。

どちらの講演も事前申込みの開始早々に満席となり、当日はキャンセル待ちの待機列ができるほどの盛況ぶりだった。白坂氏のテーマは「IoT時代のシステムデザインアプローチ～いかにしてIoTシステムをデザインするか～」、高田氏のテーマは「IoT時代の到来に向けた安全・安心な社会の構築への取り組み～分野横断的に活用できる『つながる世界の開発指針』の紹介～」である。

両名には、IPA特別講演の後にIPA展示ブースへ移動していただき、SEC先端技術入門ゼミの講師も務めていただいた。

IPA特別講演ではそれぞれのテーマの全体像の話を、SEC先端技術入門ゼミでは、具体的な事例などを交えた話をしていただいた。

6 IPAセッション



IPAセッションでは、「システムモデリングの新潮流：STAMP/STPA紹介～システム理論に基づく新しい安全解析手法～」と「組み込み業界初『組み込みソフトウェア開発データ白書2015』の公開と狙い～個人依存から組織力活用型に向かうIoT時代の組み込みシステム開発～」をテーマに講演を行った。どちらの講演も満席となり、組み込みソフトウェア開発データ白書に期待する組み込み業界の関係者や、STAMPを新たな開発手法の参考とする参加者が多く見受けられた。

7 ETWest2016を振り返って

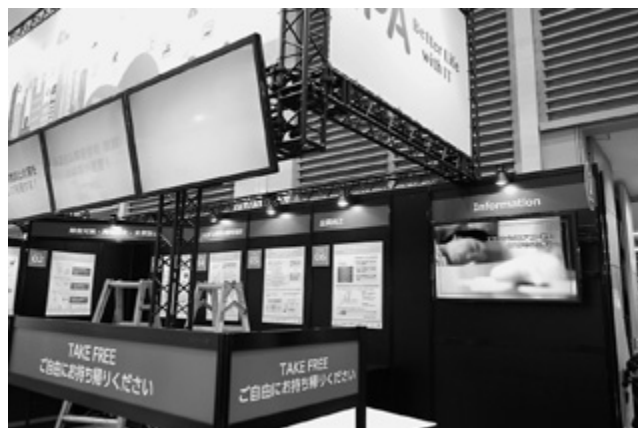


本展示会の開催場所がグランフロント大阪に移り、3度目の開催となった。大阪駅からのアクセスが良く、今回の出展も、過去最高の動員を記録できた。

今回は、新たな試みとして、事業内容を分かりやすく来場者に伝えるために制作した、「つながる世界の開発指針」の動画がある。残念ながら展示会場の中では、じっく

りと視聴いただける環境ではなかったが、パネル展示や資料配布以外にも我々の活動を知っていただくツールとして、今後も大いに役立てていき、来場者・参加者の理解度・満足度の更なる向上に努めていく。

また、今回も展示会場で来場者から数多くの貴重なご意見をいただいた。これらのご意見は、以降の出展の参考にすると共に、今後のIPA/SEC事業活動に活かしていきたい。



8 謝辞

IPA特別講演、IPAセッション、SEC先端技術入門ゼミ、ブースプレゼンにご登壇いただいた外部講師の皆様、並びに関係団体の皆様には、本展示会出展に際し、多大なるご支援を賜りました。ここに深謝いたします。

- ▶ ETWest2016 IPA/SECウェブサイト
<http://www.ipa.go.jp/sec/events/20160707.html>
 - IPAブースプレゼン・SEC先端技術入門ゼミ・IPA特別講演・IPAセッションの講演資料がダウンロードできます
 - IPA特別講演・SEC先端技術入門ゼミ(一部)の動画を公開しています
- ▶ 「つながる世界の開発指針」動画
<http://sec.ipa.go.jp/seminar/ondemand/index.html>
 - IPA展示ブースで投影した動画がご覧いただけます

情報システムの事故データ

情報システムの障害状況

2016年前半データ

IPA顧問 松田 晃一

SECシステムグループ 主任 八嶋 俊介

2016年1月から6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、昨年後半に引き続き、マイナンバー制度に関連したシステムの障害が多かった。その上、その他の重要な社会インフラシステムの障害も重なっており、発生数は相変わらず多い。

1. はじめに

本稿では、2016年1月から6月までの2016年前半の半年間に報道された情報システムの障害状況について、まず次節で今期の概況について述べる。続く3節では、今期も引き続き多数発生したマイナンバー関連の障害についてとりまとめて報告する。4節においては業務処理の誤りが長期間にわたって認識されずに運用されてきたものが発覚した事故例について、更に5節では設計上の上限値を上回った入力が生じたために障害となった事例を取り上げる。これらはいずれも過去に取り上げた事例と同種のものであるが、今期も相変わらず発生しているため、重ねて取り上げる。

2. 2016年前半の概況

2016年1月から6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26件であるが、そのうち6件は2015年に発生した障害が2016年に入って報道されたものである。従って、2016年に発生した障害件数は20件となり、その全体は表1に示す通りである。なお、2016年以外に発生した障害は別表として示している。報道時期と実際の障害発生時期とが大きくずれた事例は過去の連載にも散見されるので、今回それらを実際の障害発生時期に修正した上で年ごとの発生件数及び月平均データを見直してみた結果が図1である。これまでの連載データに比べて各年の件数に数件の増減があるが全体の傾向は大きく変わらない。2016年前半の障害発生件数20件を月平均にすると3.3件/月となった。今後は図1に示すデータをもとに連載を継続することとしたい。

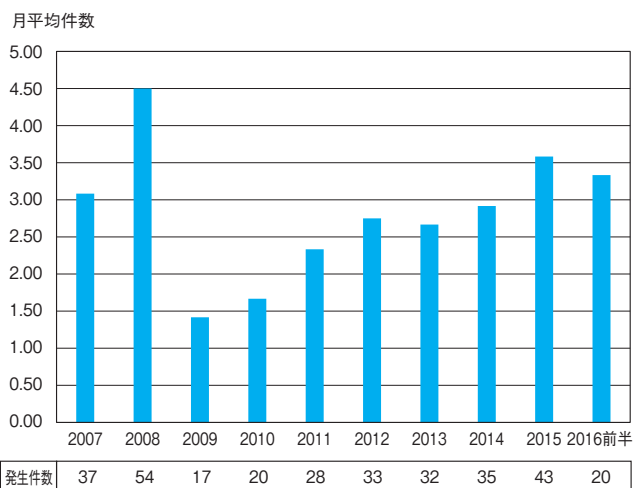


図1 情報システムの障害発生件数の推移

マイナンバー関連システムについては、昨年10月に運用が開始された後、多数の障害が発生して問題となっていたが、今期に入っても引き続き障害が発生した。本年4月末には問題が解決したとの発表があり今後は解消される見込みであるが、なお状況を見守る必要がある。その概要について次節で取り上げる。

また、今期中には日本を代表する大手航空会社2社のシステムが相次いで障害を起こし(事例1614、1615、1616)、多数の利用客に大きな影響を与えた。とくに、事例1616は過去にも障害を起こした同じシステム(事例1422 [松田2 2014])が再度障害を起こしたものであり、前回の障害時に対策されたはずの再発防止策が結果として活かせなかったわけで、その原因をしっかりと点検する必要がある。

今期の事例の中には、以前からシステムに内在していたながら見逃されたまま長期間運用されてきた不具合が偶然発見された事故が3件(別表 事例2、3、5)報道されている。この種の事故は昨年にも6件発生しており、その影響を考えると無視できない状況である。以前にも同

種の問題を取り上げた[松田1 2015][松田2 2015]が、あらためて4節で取り上げ、注意を喚起したい。

また、事例1618は設計上の上限値を上回る入力があったために発生した障害である。これも以前に同種の障害が発生しており、その障害をもとにSECで作成された教訓も公開されているが、再発を招いた。システムを取り巻く環境が激しく変化する今日では、このような事象は今後もますます頻発する可能性があり、過去にも取り上げたが[松田2013][松田2 2014]、再度第5節で取り上げる。

3. マイナンバー関連事故

2016年1月から6月に報道された、マイナンバー関連の障害は合計8件である。実際にはこれよりも更に多くの報道がされていたが、システムの主体である地方公共団体情報システム機構(以下、J-LIS)から公表された3件のプレスリリースごとに項目をまとめたため、本稿では8件とカウントしている。

事例1604、1606、1612については、表1の通りJ-LISからプレスリリースが発表され、のちにトラブルの詳細が報告された事例である。J-LISの中継サーバー内の障害により、当該システムに過度に通信が集中すると、全国の各市区町村から送付されるデータの処理が大幅に遅延し、カード管理システムに接続できない状態となった。結果、個人番号カードの受け取りのため、自治体の窓口に来庁した住民に、カードを渡せない状況が多発した。自治体からJ-LISに届いた問い合わせは、合計で約580回に及んだ。

個人番号カードは、発行時に利用者が暗証番号を決め、

窓口の端末にて自分で番号を打ち込んで登録する仕組みとなっている。システムが利用できないときは、暫定対応として、自治体の職員が住民から暗証番号が書かれたメモを受け取り、代わりに登録するといった対応もされた。しかし、この対応に起因して、職員が預かった暗証番号のメモを、誤って全く別の住民に郵送してしまったという人為的ミスも発生した。

事例1613は、人為的ミスにより、全く別の2人の人物に、同じ個人番号が割り当てられてしまった事例である。過去の転居に伴う転入手続きの際、自治体の職員が住民票コードに関するミスをしてしまっていた。個人番号は、住民票コードを元に生成されるため、今になってその影響(不具合)が発現し、本来重複することがない個人番号が、全く別の2人に重複して発行される事態となった。

マイナンバーのシステムは言うまでもなく、これからの行政に必要不可欠なものであり、かつ極めて重要な個人情報を取り扱われているシステムである。障害やヒューマンエラーによるシステムの長時間の停止・情報漏えいは、制度自体の信頼性の低下を招きかねない。前回の記事では、システムの開発時、操作ミスがないようユーザビリティを向上させることについて言及した。しかし、このような重要な生活の基盤を担うシステムを取り扱う際、これからは、障害やヒューマンエラーは起こるものとして認識しておくことも必要ではないだろうか。その上で、万が一障害やエラーが発生したとき、影響を最小限にするようなしなやかなシステムを構築すること、レジリエンスエンジニアリングの考え方を取り入れることも、高信頼/安全なシステムを実現する上で、今後は重要な選択肢の一つとなり得ると考えられる。

表1 2016年前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				
1601	和歌山県 緊急速報メール	2016	1	1	14時08分	県全域の携帯電話利用者に「大きな津波の観測があった」との緊急速報メールが自動配信され、県は約1時間後に誤報だったと発表した。	午後2時8分、「大きな津波の観測があった」と避難を呼びかけるメールが一斉に送信された。更に3時1分、「津波が更に大きくなっている」と送信した。誤報と確認し、3時15分に訂正のメールを配信した。 津波の高さを測るために必要な潮の満ち引きのデータを、県職員がシステムに入力しないミスがあり、潮位の変化を津波と誤認識したとみられる。	ヒューマン エラー	・日本経済新聞朝刊 (2016.1.1)
		2016	1	1					
1602	鳥取県 情報センター マイナンバー カード 管理システム	2016	1	7		県内の一部自治体で個人番号カードを交付するシステムに不具合が発生した。 県内でカードの交付はまだ始まっていなかったため、大きな問題は確認されなかった。	南部町が7日、システムの保守・運営を委託している「鳥取情報センター」に連絡して発覚した。同町を含め、県内8つの自治体で同様の現象が起きたと見られる。役場に届いた個人番号カード情報の確認のため端末からサーバーに接続を試みても、セキュリティ設定の誤りで作業ができなかった。	設定ミス	・毎日新聞(地方版) (2016.1.16)
		2016	1	15	午前				
1603	J-LIS マイナンバー カード 管理システム	2016	1	13	正午頃	一部自治体で一時的にカード交付ができなくなった。	障害があったのは、番号カードの申請や交付を管理する全国共通のシステムで、カード交付の際に市区町村がアクセスし、必要な情報をやり取りする。 富山県舟橋村では正午頃から約3時間半、山梨県中央市では正午頃から約1時間半、システムが使えなかった。	アクセスの 集中	・産経新聞朝刊 (2016.1.14)
		2016	1	13					

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				
1604	J-LIS マイナンバー カード 管理システム	2016	1	18	15時40分	一部の市区町村の統合端末において、個人番号カードなどに関する業務が実施できない事象が生じた。二日間計約280の市町村から問い合わせがあった。窓口に来た住民に個人番号カードを渡せなかったり、市町村から機構に住民情報を送れなくなったりした。	住基ネットとカード管理システム間に設置している住基ネット中継サーバー2号機が不安定な状態となっていた。このことにより、カード管理システムに接続できる市区町村の数が制限され、一部の業務の処理について統合端末において処理できない状態となった。	ハードウェア 障害	<ul style="list-style-type: none"> ・J-LISプレスリリース(2016.1.19) ・朝日新聞朝刊(2016.1.20)
				19	08時30分				
1605	J-LIS マイナンバー カード (ICチップ)	2016	1	21		マイナンバーカードのICチップに不具合があり、約2万6,000枚が再発行されていた。	住民票交付などで使うICチップ内の個人を認証するデータに不具合があった。カードの製造後、データを入力する過程のミス。自治体が、住民に手渡す前の確認作業で気づいてJ-LISに連絡した。	データ入力 ミス	<ul style="list-style-type: none"> ・朝日新聞朝刊(2016.2.25)
1606	J-LIS マイナンバー カード 管理システム	2016	1	21	18時40分	市区町村の統合端末において、個人番号カードなどに関する業務を行った際にエラーメッセージが表示され、一部の市区町村において業務が実施できない事象が生じた。三日間で約100の自治体から問い合わせがあった。	住基ネットとカード管理システム間に設置している住基ネット中継サーバー2号機が不安定な状態となり、カード管理システムに接続できる市区町村の数が制限され、一部の業務の処理について統合端末において処理できない状態となった。22日には、中継サーバーの状態を常時監視して、事象が発生した場合にはOSの再起動を迅速に行えるようにする即応体制を構築すると共に、24日には、中継サーバーを1台増設する措置を講じた。	ハードウェア 障害	<ul style="list-style-type: none"> ・J-LISプレスリリース(2016.1.25) ・朝日新聞朝刊(2016.1.26) ・日本経済新聞(2016.1.26) ・日経コンピュータ(2016.2.4)
				22	09時40分				
1607	NTTドコモ 料金システム Mobills	2016	1	28	04時42分	約10時間にわたり、契約者向けwebサイト「My docomo」などで料金やポイントの情報が閲覧できなくなったほか、キャリア決済サービス「ケータイ払い」が利用できなくなったりした。「ケータイ払い」が利用できなかったのは最大で6万人。2月3日までに約2,500件の問い合わせがあった。	DB管理ソフトの不具合が原因。更にそれをきっかけに、同ベンダ製のミドルウェアで構築したAPサーバーでも障害が発生した。(共有メモリでのデータ損傷)立て続けにパッケージの不具合が顕在化したことにより、長時間サービスを止める結果となった。	パッケージの 不具合	<ul style="list-style-type: none"> ・NTTドコモお知らせ(2016.1.29) ・日経コンピュータ(2016.3.17)
				28	15時00分				
1608	山形県 緊急速報メール	2016	2	1		北朝鮮からミサイルが発射されたという誤った内容のメールを、地元のテレビ局やラジオ局など44個所に送信した。	約3分後に訂正のメールを送った上で電話で謝罪した。緊急速報が報道されることはなかった。	不明	<ul style="list-style-type: none"> ・朝日新聞朝刊(2016.2.2)
				1					
1609	みずほ銀行 宝くじシステム	2016	2	1		1日に予定していた「ロト6」と「ナンバーズ」の抽せんができなかった。通常は午後6時45分に抽せんを開始し、8時頃には結果を公表するが、今回は抽せんが翌日の午前1時10分頃にずれ込んだ。	販売実績を集計している外部委託先のオンラインシステムが停止。当せん口数の計算ができず、抽せんができなかった。一部の売り場でくじの発券端末機を入れ替えた際、くじの通し番号に重複が生じた。システムは東京と大阪で二重系となっているが、アプリケーションの実装が異なっていたため、処理結果の不整合が生じた。どちらのセンターのデータが正しいのか判断できなかったため、抽せんができなかった。	ソフトウェア 障害	<ul style="list-style-type: none"> ・朝日新聞朝刊(2016.2.2) ・日本経済新聞朝刊(2016.2.2) ・毎日新聞朝刊(2016.2.2) ・日本経済新聞夕刊(2016.2.2) ・朝日新聞朝刊(2016.2.3) ・日経コンピュータ(2016.4.28)
				2	01時10分				
1610	JR東日本 特急券予約 サイトシステム	2016	2	5	15時55分	「えきねっと」と「モバイルスイカ」において、約4時間にわたって新規の予約や予約の取り消しなどができなくなった。本サービスにおいては、一日平均約5万席の予約利用がある。	不具合の原因は、システムのサーバー間のやり取りが正しく機能しなかったためとみられる。	サーバー間 エラー	<ul style="list-style-type: none"> ・読売新聞朝刊(2016.2.6)
				5	19時55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				
1611	NTTデータ 地銀共同センター (地銀7行のATM)	2016	2	15	09時34分	全国の地方銀行7行にてシステム障害が発生し、ATMの取引ができなくなるトラブルがあった。ATMの利用中に障害が発生した取引は、7行合わせて約4,000件。7行は秋田、足利、池田泉州、鳥取、四国、大分、西日本シティの各行。	1回目(2月15日)…取引制御プロセスの処理が遅延し、待ち行列が滞留。ATMの処理がタイムアウトした。ベンダから提供された製品マニュアルにミスがあり、取引量増加に対応するための作業に設定ミスがあったことが原因。 2回目(2月26日)…プロセスの滞留発生を、重点監視体制にて監視した結果、ログの書き込みが大量となった。大量ログデータの転送を開始した際、メモリが足りなくなりプロセスがタイムアウト、ATMの停止に至った。	<ul style="list-style-type: none"> サーバーの設定誤り 処理の集中 	<ul style="list-style-type: none"> 日本経済新聞朝刊(2016.2.16) 日経コンピュータ(2016.5.12)
				26	14時28分				
1612	J-LIS マイナンバー カード 管理システム	2016	2	22	13時00分	市区町村の統合端末からJ-LISのシステムにアクセスができず、多くの市区町村において個人番号カードの交付前作業、交付作業が実施できない事象が生じた。全国の自治体から約200件の問い合わせがあった。	個人番号カードの発行管理などを行っているカード管理システムが不安定な状態となったため、市区町村から送信されるデータの処理が大幅に遅延し、多くの市区町村の統合端末において業務を実施できない状態となった。	不明	<ul style="list-style-type: none"> J-LISプレスリリース(2016.2.22) 朝日新聞朝刊(2016.2.23) 産経新聞朝刊(2016.2.23)
				22	16時30分				
1613	長野市 住民票システム	2016	2	23	香川県坂出市在住の男性と、長野市在住の別の男性に同じ個人番号が割り振られていた。	坂出市の男性と氏名の読み方、生年月日が同じ別の男性が、都内から長野市へ転入する手続きを2010年にした際、長野市の職員が「住民票コード」をめぐってミスをしていた。(坂出市の男性の情報と間違えて交付した)その結果、住民票コードを踏まえて作られるマイナンバーの重複につながった。	ヒューマン エラー	<ul style="list-style-type: none"> 朝日新聞夕刊(2016.2.23) 	
1614	ANA 搭乗手続き システム	2016	2	24	13時00分	東京・羽田空港を含む全国48空港で搭乗手続きのシステムに不具合が起き、一部の便の出発が遅れた。	国内線の搭乗手続きを行うシステムが一時ダウンし、カウンターや自動チェックイン機の手続きができなくなった。	不明	<ul style="list-style-type: none"> 朝日新聞朝刊(2016.2.25)
				24	13時30分				
1615	ANA 総合旅客 システム	2016	3	22	03時44分	全国の国内線で搭乗手続きが行えなくなった。国内線148便が欠航、391便が遅延し、計約7万2,000人に影響が出た。	データベースサーバー間(4台)の同期処理を中継するネットワーク中継機の故障が原因。具体的には以下の2点の故障。 ・中継機能の故障 データベース間の同期処理が正常に終了せず、データの整合性が保てなくなるため、データベースを自動的に停止する機能が働いた。 ・「故障シグナル」の発信機能の故障 装置が故障した場合、「故障シグナル」を発信し、予備機に自動的に切り替わる設計になっているが、「故障シグナル」を発信せず、予備機に自動的に切り替わらなかった。	ハードウェア 障害	<ul style="list-style-type: none"> ANA プレスリリース(2016.3.22,30) 朝日新聞電子版(2016.3.22) 日本経済新聞夕刊(2016.3.22) 読売新聞朝刊(2016.3.23) 朝日新聞朝刊(2016.3.23,31) 日本経済新聞朝刊(2016.3.23,31) 毎日新聞朝刊(2016.3.31) 日経コンピュータ電子版(2016.4.1,4)
				22	11時30分				
1616	日本航空 重量管理 システム	2016	4	1	07時48分	乗客や貨物などの重さから航空機の重心位置を計算する「重量管理システム」がダウンした。国内の各空港で航空機の出発が遅り、計50便が欠航し約7,000人に影響が出た。30分以上遅延した便は137便で、最大約1万7,000人に影響が出た。 ※本システムは、導入直後の2014年6月にも障害を起こしている。	2016年3月23日に適用したシステムの更新プログラムに不具合があったことが原因。 開発元から提供されたパッチの中に、キャッシュの排他制御を追加する設計変更があり、もともと実装されていたディスクの排他制御との間でデッドロックが発生した。 待機系を稼働させたものの、処理性能が本番系の7～8割しかなく、処理しきれなかった。	ソフトウェア 障害	<ul style="list-style-type: none"> 朝日新聞夕刊(2016.4.1) 日本経済新聞夕刊(2016.4.1) 毎日新聞夕刊(2016.4.1) 朝日新聞朝刊(2016.4.2,7) 日本経済新聞朝刊(2016.4.2,7) 日経コンピュータ電子版(2016.4.8) 日経コンピュータ(2016.4.28)
				1	09時40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				
1617	四国電力 検針・電気料金 関連システム	2016	4	4		以下に示す複数のトラブルが 相次いで発生した。 1. 約2万件の電気料金振込票 を使用量を誤って印字して 送付 2. 約170件の口座振替やカード 払いの誤請求 3. 約170万件の振込票の送付 や口座振替が遅延 4. 約1,600件の検針結果デー タの喪失	1. 実際の使用量よりも余分に加算されて いた。原因はプログラムの変更ミス。 新料金メニューの追加に伴い、誤って 変更した。 2. 誤請求につながった原因は以下の2点。 ・オペレーションミスにより、顧客の二 重登録やファイルの二重作成を行い、 本来の使用量の2倍で請求したケース があった。 ・新規に開発した機能のプログラムミス により、本来の2倍の料金を請求する ケースが生じた。 3. 料金の計算結果にチェック機能を設け ていたが、正しく機能していないケー スがあった。確認作業のため、口座振 替を当面停止したため、遅延が生じた。 4. 担当者が誤操作を行ったため、デー タを消失した。	ソフトウェア 障害 オペレーション ミス	・四国電力 プレスリリース (2016.5.17) ・日経コンピュータ (2016.6.23)
		2016	4	25					
1618	JR東日本 新幹線運行 システム (電光掲示板)	2016	5	4	始発	東北、上越、北陸の各新幹線 の計44駅すべてで、発車時 刻などを示す電光掲示板が表示 されなくなった。	5月3日、4日の運行本数が1,606本に達 し、システムの処理可能本数を6本上回っ たことが原因。ゴールデンウィークの臨 時列車や、北陸、北海道新幹線の開業が 影響した。 ダイヤへの影響はなかったほか、駅員を 増やして対応したため、乗客にも大きな 混乱はなかった。	ソフトウェア 障害	・朝日新聞朝刊 (2016.5.5) ・日本経済新聞朝刊 (2016.5.5)
		2016	5	4	終電				
1619	ゆうちょ銀行 決済システム	2016	5	13	09時30分	同行の口座から他行の口座へ の自動振込みのほか、保険金や 株式売却代金の受け取りで最 大1万件程度の遅れが生じた。	直接の原因はシステムのサーバーがダウ ンしたため。	ハードウェア 障害	・毎日新聞朝刊 (2016.5.14) ・日経コンピュータ 電子版 (2016.5.14) ・日本経済新聞朝刊 (2016.5.15)
		2016	5	14					
1620	JR東日本 電光掲示板	2016	6	18	15時00分	JR会津若松駅にて、列車の発 車時刻や乗り場を伝える電光 掲示板に、「喜多方ラーメン」 と表示された。	駅員が「さよなら 485系(特急あいつ)」 と駅舎内の端末で打ち込んだが、正しく 表示されるかどうかの確認を怠った。 12年前に端末に入力されたデータが誤っ て送信されたことが原因。 2004年2月に仙台-喜多方駅間を走った 列車「喜多方ラーメンフェスタ号」のデー タが端末に残っていた。	データ誤送信	・朝日新聞朝刊 (2016.19) ・日本経済新聞朝刊 (2016.6.20) ・朝日新聞朝刊 (2016.6.21)
		2016	6	18	15時30分				

4. 長期間の不具合放置

別表 事例3は、特定の会社の中で使われるシステムで発生した事故であるが、有価証券報告書を5期分さかのぼって訂正するという結果を引き起こし、社会に対して少なからず影響を与えた事例である。その原因は新たに開発した在庫管理システムに不具合があったため、仕掛品の在庫についてシステムで管理している在庫データが実際の在庫よりも約10億円過大計上されたまま運用され、誤った財務諸表が作成されてきたとのことである。

社内調査委員会によれば、システム構築に当たって経理に関する専門家の参画がなかったことが大きな原因として挙げられている。開発に当たってシステムに関する専門家と共に、業務の専門家の参画と両者の協働作業の重要性が以前から指摘されているが、それが欠落した結果が現れた典型的な事例であろう。両者の協力体制ができたとしても、更に両者間でのコミュニケーションや合意形成の難しさもあるが、その困難を軽減する方法の一つとして合意形成ガイドが取りまとめられ公表されている[SEC2010]ので参考にしていきたい。

別表 事例3は企業内で使われるシステムでの事故であったが、別表 事例5の年金支払いシステムや別表 事例2のカードキャッシングシステムなどのように我々の身近なシステムにおいてデータの信憑性が疑われるような問題が発生すると、システムに対する信頼性が一気に揺らぎ影響は大である。とくに、事例1530 [松田2 2015] や事例1329 [松田1 2014]のようにシステムの不具合によって、誤った与信情報が長期にわたって流通したよう

な事故は、データの正しさを本人自らがチェックのしようがないために影響はより重大である。

システムがダウンするなどの現象が発生する不具合の場合は、直ちに不具合が検知される。しかし、システムは一見正常に運用されているが、実は重大な問題を抱えたままで誰も気がつかない中で誤りが生じているようなケースはやっかいである。このようなシステムの開発、保守、運用に携わる方々の責任は極めて重い。

別表 2016年以外に発生した障害

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				
1	ノバルティス ファーマ 副作用報告 システム	2015	1			刷新した「副作用報告システム」に障害が発生し、同社が製造販売する薬品についての報告が遅れた。業務を人手に切り替えたが、約5,600症例の報告が期限に間に合わず、厚生労働省から行政処分を受けた。	2014年までは日本法人が独自に開発したシステムを利用してしたが、2015年1月に、グローバルのグループ標準で使うシステムに変更。このシステムに障害が発生した。 情報閲覧画面を開くのに時間がかかる、データベースの過負荷でシステムが停止するなどした。また、標準規格で作成された報告書を作成し、報告先であるPMDA(医薬品医療機器総合機構)へ伝送する機能も正しく動作しなかった。	ソフトウェア 障害	<ul style="list-style-type: none"> 日本経済新聞夕刊(2015.11.11) 朝日新聞朝刊(2015.11.14) 日経コンピュータ(2016.1.7)
		2015	5						
2	イオン銀行 イオンカード キャッシング サービス	2015	4			イオンカードのキャッシングサービスにおいて、利息を誤って請求していた。 過去10年分の調査で、対象契約数は2,401件、返金額は約600万円。	誤請求が発生したのは、キャッシングをして、その一部を約定日前に返済したケース。このケースでは、利息を日割りで計算するが、システムには日割りの計算機能がなく、手作業で計算していた。手作業での事務オペレーションにミスがあり、過剰請求につながった。 顧客対応や事務処理は、現場の判断で個別に対応していたので、情報システム部門や経営陣は、業務実態を把握していなかったとみられる。	オペレーション ミス	<ul style="list-style-type: none"> 日経コンピュータ(2016.5.12) 日経コンピュータ(2016.6.9)
		2016	1						
3	日新電機 生産管理 システム	2015	10			棚卸し資産を約10億円過大計上しており、2011年3月期から2015年3月期まで5期分の有価証券報告書と四半期報告書を訂正した。	加工途中の材料である仕掛品の在庫に関して、ソフトウェア障害により、システムで管理しているデータと実際の在庫状況との間に食い違いが生じていた。 2015年10月に問題が発覚した際、同社では社内調査委員会を設置し、発生した経緯などを調査した。まとめられた調査報告書では、不具合が発生した最大の理由として、「システム構築プロジェクトに、経理部や経理システム関連の知識を持つ本社情報システム部が参画しなかった」点が挙げられている。	ソフトウェア 障害	<ul style="list-style-type: none"> 日経コンピュータ(2016.2.18)
4	J-LIS マイナンバー カード 管理システム	2015	10			東京都葛飾区のマイナンバー通知カード約5,000世帯分が未作成だったにもかかわらず、機構のシステム上では正常終了と認識されていた。	葛飾区から持ち込まれた住民データを「中継サーバー」から「管理サーバー」に移行したときにシステムが一時停止した。その際、実際にデータ処理が行われていなかったにもかかわらず、機構のコンピューター端末上では「終了」と表示されていた。葛飾区分のデータ移行を表すメモリ容量の変化もモニター上で確認済みで、システム上は正常に作動したことでなっていた。結果、データのない通知カードは印刷されず、住民に郵送されなかった。	ソフトウェア 障害	<ul style="list-style-type: none"> 産経新聞朝刊(2016.1.1)

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				
5	共済年金 年金一元化 システム	2015	10			元国家公務員ら約1万2,000人分の共済年金の支給額に誤りがあった。対象は2015年12月と2016年2月で、総額は1億8,700万円。	年金制度の一元化に対応するプログラムの設定ミスが原因。2015年10月に共済年金と厚生年金が統一され、支給停止額の計算方法が変わったことで生じた。年金と給料を合わせた所得が一定以上だと、年金は一部または全額の支給が止まる。支給停止の対象者の抽出に新しいシステムで対応したが、設定ミスで漏れる人が続出した。	設定ミス	・朝日新聞朝刊 (2016.4.7)
		2016	2						
6	リクルートHD、 ロイヤリティM ポイントサービス	2015	11	24		11月24日にリクルートポイントとPontaを統合した新しいポイントサービスを開始する予定だったが、統合を延期した。新サービスは翌年2月2日に開始された。	統合作業中、システム担当者が想定外の不具合に気づき、対処方法を検討した結果、新ポイントサービスの開始を延期した。2社は、具体的にどのような不具合があったかは明らかにしていない。	システム 連携の不具合	・日経コンピュータ (2016.3.3)
		2016	2	2					

5. 環境変化への対応遅れ

今年5月に発生した事例1618は、新幹線の各駅の電光掲示板で列車名や発車時刻などすべての表示ができなくなった事故である。ゴールデンウィークの臨時増発などによって列車運行本数が1,606本と、システムの処理可能件数1,600本を超えたことが原因であった。幸い大きな混乱は生じなかったが、以前にもこの事例と同じくシステムの上限值を超えた入力が生じたために異常となり、新幹線のすべての列車の運転を中止するという大きな混乱を起こした事故があった(事例1102 [松田2011])。すなわち、列車の運行トラブルによって運行ダイヤの修正を行ったところ、必要な修正数が運行管理システムの設計上限値を超えてしまい、システムが不安定な状態になったため、すべての列車の運転を中止した事例である。これらは、列車数の増加など環境条件が年々変化してきたのに対し、システムの上限值が開発当初のまま見直されなかったことが原因であり、同種の事例は過去にもかなり発生している。

これらを受けてSECでは、情報処理システム高信頼化教訓集 [SEC1 2016] において「システム全体に影響する変化点を明確にし、その管理ルールを策定せよ」との留意点が教訓T4として示されている。システムは開発の終了後、長期にわたって運用されるが、その間にシステムを取り巻く環境は大きく変化するため、当初は妥当であった設計条件が、最新の利用条件を満足しなくなってくることはよく起こることである。環境変化に対応した適切なシステムの増強や保守などを実施すること、そのためにシステムを継続的に監視・点検し必要な更新を行う管理ルールを策定し、実行することが重要である。

6. むすび

2016年前半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・運用に当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、安全・安心な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「情報処理システム高信頼化教訓集」[SEC1 2016]として公表している。更にこの教訓を活用するための方法を情報処理システム高信頼化教訓活用ガイドブック [SEC2 2016]として公開しているので併せて参考にさせていただきたい。また、障害事例から教訓を学ぶ活動は継続的に実施しており、教訓がまとまるごとに逐次Webで公開しているので適宜参照していただきたい。

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

参考文献

- [松田2011]松田晃一・金沢成恭:情報システムの障害状況 2011年前半データ、SEC journal No.27、Vol. 7、No4、pp.150-152、Jan.2012
- [松田2013]松田晃一・鈴木三紀夫他:情報システムの障害状況 2013年前半データ、SEC journal No.34、Vol. 9、No3、pp.20-24、Sep.2013
- [松田1 2014]松田晃一・目黒達生他:情報システムの障害状況 2013年後半データ、SEC journal No.36、Vol. 10、No1、pp.25-28、Mar.2014
- [松田2 2014]松田晃一・八嶋俊介他:情報システムの障害状況 2014年前半データ、SEC journal No.38、Vol. 10、No3、pp.42-47、Sep.2014
- [松田1 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.32-37、Sep.2015
- [松田2 2015]松田晃一・八嶋俊介:情報システムの障害状況 2015年後半データ、SEC journal No.44、Vol. 11、No4、pp.48-53、Mar.2016
- [SEC 2010]情報処理推進機構 SEC:機能要件の合意形成ガイド、2010年03月31日
- [SEC1 2016]情報処理推進機構 SEC:情報処理システム高信頼化教訓集(2015年度版)(ITサービス編)、2016年3月
- [SEC2 2016]情報処理推進機構 SEC:情報処理システム高信頼化教訓活用ガイドブック(ITサービス編)、2016年2月

IoTイノベーションの期待

IPA顧問、学校法人・専門学校HAL 東京 校長 鶴保 征城

2015年2月に発表されたPwCの予測によれば、2050年のGDPランキングでは1位中国、2位インド、3位米国と続いて日本は7位になっている。メキシコにもインドネシアにも負けると予想されている。

戦後、ソニー、ホンダ、キヤノンなど、日本を代表するベンチャーが大企業に成長し、日本経済の発展を支えた。松下やシャープもその一角を支えていた。

ところが日本経済は1990年頃をピークに下降を始めた。現在までの30年近く、経営の中心課題がコストダウンと効率化になり、革新的な取り組みをしてこなかった。ニワトリが先か卵が先かはわからないが、結果的に企業のトップは守りが得意な人たちが占めている。

電気製品で言うと、新しいものはかならずソニーから出てきたが、今やソニーからワクワクするような商品は出なくなった。

辛うじて車はトヨタががんばっているが、車好きはアウディやBMW、ポルシェに乗っている人が多い。品質や価格では負けないが、魅力に欠けるのだと思う。

翻ってIoTを見てみると、2013年10月、調査会社のガートナーはインターネット接続デバイスについて次のように発表している。

- 2009年時点で、25億個のデバイスがインターネットに接続していた。うちパーソナルデバイスが16億個、IoTが9億個。
- 2020年には、300億個を超え、パーソナルデバイスが73億個、IoTが300億個となる。
- 2020年までのIoT増大によって得られる経済価値は1兆9000億ドルに及ぶ。

その後、2020年の300億個を500億個に増加させたレポートも発表されている。ファーウェイ・ジャパン副社長は2050年1000億個と予測しているが、これも控えめな数字になる可能性すら予想される。

これはまさしく革命が起ころうとしていると言っても過言ではない。このチャンスを生かし、低下する一方のGDPを押し上げるためにはイノ

ベーションが必要だ。

イノベーションを起こす企業には3つの共通項がある。

1番目は、卓越したリーダー。ソニー創業者の井深大氏、ホンダ創業者の本田宗一郎氏が代表的だ。

2番目は、変わり者社員とそれを許容する風土。不良社員かもしれないが、新しいことに粘り強く取り組む社員が必要だ。往々にして成績の良い社員は、頭の良さが邪魔をして、この点が弱い。

3番目は、サントリーのように「やってみなはれ」精神が社内に横溢していること。

イノベーションの実例を挙げると、ソニーのウォークマンだ。32年前ほど前、テープレコーダーから当時付いているのが当たり前だった録音機能を削り、その分良い音で聞けるようにしようというアイデアが出た。ソニーの全役員と技術エキスパートが大反対した。しかし当時会長だった盛田昭夫氏が「良い音だから売ってみろ」といった。すると大ヒットして、ウォークマンを聞きながらジョギングするといった新しい文化まで作ってしまった。このヒットが現在のソニーの基盤を作った。

ホンダのエアバッグも同じだ。社長以外の全員が反対したものを、小林三郎氏が16年かけて実現した。エアバッグが車の標準装備になる前は、年間1万1000人以上が交通事故で亡くなっていた。それが今は、5000人を切っているらしい。

では、IoTの世界でどのようなイノベーションが必要なのだろうか。

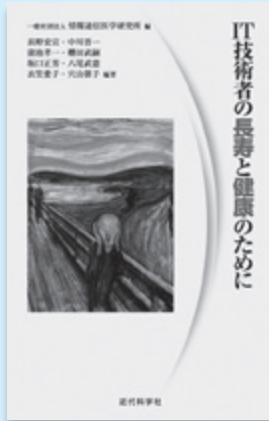
今後爆発的にネットにつながるデバイスが増えてくる状況で、中央集権的なサーバーコントロールで果たして世界が回るのかという問題がある。そのリスクを誰が担保するのか、維持費を誰が負担するのか？

どのようにして応答時間を保証するのか。

安定性や経済性、セキュリティ、応答性などの問題をはらんでいる中央集権的な仕組みではなく、端末同士が会話をして自律的に制御し、有機的な社会を築いていくことが必要ではないだろうか。

リーダーに人を得て、社風を変えられれば、日本の出番がくるのではと期待している。

IT技術者の長寿と健康のために



一般社団法人
情報通信医学研究所 編
長野 宏宣 他著

ISBN : 978-4-7649-0513-9
近代科学社刊
A5変型版・224頁
定価2,400円(税抜)
2016年6月28日刊

本書は長い間NTTにおいてソフトウェア開発関連の研究及び実務に従事されていた長野宏宣氏はじめ8名の著者により書かれたものであり、うち4名はお医者さん。本書は以下の章立てである。

- 第1章 IT企業における産業医学
- 第2章 IT技術者と健康
- 第3章 わが国のIT産業の実態
- 第4章 IT産業の技術者を大切にす—適切な開示制度を
- 第5章 IT企業の未来について

第1章では、情報通信医学研究所代表の中川博士が1994年にITの世界に触れ、IT研究者として参画し、産業医として医学に帰り2011年に当研究所の代表となる経緯が説明されている。医学研究者、IT研究者、IT業界の産業医としてIT現場の実務の観察を通して、IT技術者が「健康で長生きするための生活習慣というプログラム」を提案されている。

第2章では編者らの体験も含め、IT技術者の罹患しやすい病例が臨場感をもって記述されている。

第3章では、IT技術者がメンタルな病気になる可能性の高い労働環境であることが紹介されている。

第4章は、IT業界では人材が極めて重要であるにもかかわらず、多重下請け構造もあり、社員の健康状態が会社により適切に管理されていないのではないかと状況報告書の記載事項とすることを提言している。

第5章は、前章までと全く異なり、IT産業そのものの今後に対して展望を記述している。

第4章にて提案されている有価証券報告書での開示制度が早急には実現しないと想像される現在、IT産業にかかわるものは、「なによりも、人が大切」という帯の安西悠一郎博士の言葉を心にとめることにしよう。

(新谷 勝利)



山岡 俊樹 著

ISBN : 978-4-320-07198-8
共立出版
A5判・244頁
定価3,200円(税抜)
2016年6月27日刊

サービスデザイン

フレームワークと事例で学ぶサービス構築

モノがあふれる現代社会では、画一的な物作りからサービスを含む価値をユーザーに選んでもらうことを重視する方向へと動いている。一製品の機能ではなく、製品やシステムを統合化して、サービスをデザインすることで、他にはない価値を創造・提供できるが、そのサービスをデザインするにはどうしたらよいのか。本書は汎用システムデザインプロセスによりこれに答えるものとなっている。特徴は個人の直感やセンスに依存することなく、ユーザーが満足するサービスを論理的にデザインする手法にある。

近年、ワークショップや顧客との協創などが提唱されているが、その限界を考える必要があり、全面的に依存するのではなく参考程度にとどめ、自力で発想できる目利き力を付ける必要があると述べている。本書で紹介しているフレームワークと関連知識はその目利き力を作るための基本であり、体験を通じて、この枠組みを何度も回し目利き力を高めていくことが可能であるということを紹介している。

本書の前半はサービスデザインの考え方やデザイン方法について書かれており、後半は様々な企業でのサービスデザインの活用事例で構成されている。活用事例にはIoTを活用したサービスデザイン戦略などといった最新事例も含まれる。理論と事例を一冊で手に取ることができる書籍である。

(遠藤 秀則)

編集後記

今号は組込みソフトウェア開発を特集しています。組込み系の特集はこれまで何度か実施しているのですが、本格的なIoT時代を迎えての発行は今回が初めてです。組込み系技術者が置かれた環境は、これまで同特集が発行された時代とは異なる部分も多いのではないのでしょうか。巻頭言のJASA築田会長からの「IoTの核心技術を担う組込みシステム技術」、所長対談のESIP組込み適塾塾長の井上先生からの「IoTの台頭で、組込み技術者にも包括的な知識が必要」といったメッセージは、まさに現在、そしてこれからの組込み技術者に向けての示唆をいただいたと思っています。

本誌が届いているころ、酷暑は過ぎ去ってくれているのでしょうか。夏休みを年間の折り返し地点と捉え、後半戦を走り始めている方も多いと思います。夏の疲れを持ち越さず、爽やかな秋を満喫されますことをお祈り申し上げます。（編集長）

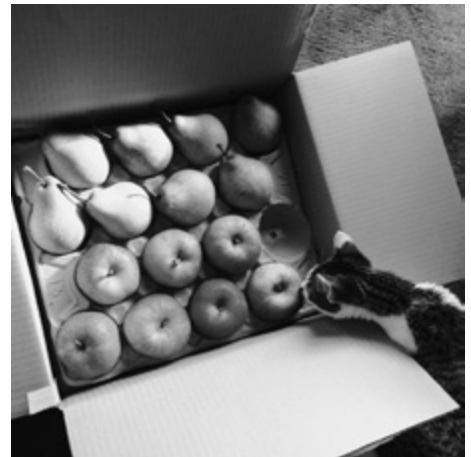
編集部より

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

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

SEC journal 編集委員会

編集委員長	遠藤 秀則
編集委員 (50音順)	荒川 明夫
	石橋 正行
	日下 保裕
	佐藤 康彦
	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



秋の味覚

撮影：K.Hasegawa

SEC journal 第12巻 第2号 (通巻49号) 2016年9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.46
第12巻第2号(通巻49号)
2016年9月1日発行

©独立行政法人情報処理推進機構

ISSN 1349-8622

