

# SEC Journal

42

## 巻頭言

**横塚 裕志**

一般社団法人 情報サービス産業協会 (JISA) 会長

## 所長対談

**車載ソフトウェア開発の今後の方向性**

村山 浩之 株式会社デンソー 技監

## 論文

**チェックリストを用いたコードレビューと判別モデルの組合せによる  
モジュールの不具合リスクのランク付け**

笠井 則充 三菱電機株式会社 通信機製作所 / 森崎 修司 名古屋大学 / 松本 健一 奈良先端科学技術大学院大学

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

角田 雅照 奈良先端科学技術大学院大学情報科学研究科 近畿大学工学部情報学科  
門田 暁人 奈良先端科学技術大学院大学情報科学研究科 岡山大学大学院自然科学研究科  
松本 健一 奈良先端科学技術大学院大学情報科学研究科

## 報告

**国際展示会「CeBIT2015」への参加及び海外有力機関との意見交換  
ドイツ Steinbeis 訪問報告**

**Embedded Technology West 2015 (ETWest2015) 出展報告**

## 連載

**情報システムの障害状況 2015 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

**SWEBOK V3.0 日本語訳版の連続紹介 - 3 の 3**

新谷 勝利 新谷 IT コンサルティング

## 組織の活動紹介

**日本ファンクションポイントユーザ会活動紹介**

藤貫 美佐 日本ファンクションポイントユーザ会 会長 株式会社 NTT データ 技術開発本部 部長

## Column

**イノベーション - 自分の頭で考え、行動する -**

## 巻頭言 ……1

### デジタルビジネス時代を迎えて

横塚 裕志 一般社団法人 情報サービス産業協会 (JISA) 会長

## 所長対談 ……2

### 車載ソフトウェア開発の今後の方向性

村山 浩之 株式会社デンソー 技監

## 論文 ……8

### チェックリストを用いたコードレビューと判別モデルの組合せによるモジュールの不具合リスクのランク付け

笠井 則充、森崎 修司、松本 健一

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

角田 雅照、門田 暁人、松本 健一

## 報告 ……24

### 国際展示会「CeBIT2015」への参加及び海外有力機関との意見交換

八嶋 俊介 SEC システムグループ主任、十山 圭介 SEC 調査役

### ドイツ Steinbeis 訪問報告

片岡 晃 IPA IT人材育成本部 イノベーション人材センター センター長

### Embedded Technology West 2015 (ETWest2015) 出展報告

荒川 明夫 SEC 企画グループ主任

## 連載 ……32

### 情報システムの障害状況2015 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

### SWEBOK V3.0 日本語訳版の連続紹介ー3の3

新谷 勝利 新谷ITコンサルティング

## 組織の活動紹介 ……44

### 日本ファンクションポイントユーザ会活動紹介

藤貫 美佐 日本ファンクションポイントユーザ会 会長 株式会社NTT データ 技術開発本部 部長

## Column ……48

### イノベーションー自分の頭で考え、行動するー

鶴保 征城 IPA 顧問 学校法人・専門学校 HAL 東京 校長

## 書籍紹介 ……49

## 編集後記 ……50

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

# デジタルビジネス時代を 迎えて



一般社団法人 情報サービス産業協会 (JISA) 会長

横塚 裕志

## デジタルビジネス時代の到来

あらゆるモノがインターネットにつながる IoE (Internet of Everything) の時代は、これまででは考えられないほどの大量のデータが収集され、その解析如何では新たなビジネスの誕生を予感させます。

本年4月、産業構造審議会 情報経済小委員会「中間とりまとめ」<sup>\*1</sup>においても、「データ流通量の爆発的な増大を背景として」「企業や個人の行動様式」を「大きく変化させ」「世界各国でビジネスモデルの革新を生み出している」と書かれています。

このような時代の到来は、既存事業を破壊的に変革すると同時に、新しい技術を獲得することで、従来価値に囚われない、イノベティブな事業を引き起こす絶好の機会に直面しているとも言えます。

## ビジネスを創造するソフトウェア

その一方で、現今の情報サービス産業は10月のマイナンバー制度を始め、2020年東京オリンピックや金融分野での大規模システム開発案件など旺盛な需要を受け、供給サイドとして多忙を極めています。また「攻めのIT経営銘柄」に代表されるように、ITは新たなビジネスモデルの創出や収益力強化のためのツールであると考えられるようになってきており、ますますソフトウェアの開発需要は高まるでしょう。

このように社会はソフトウェアに依存し、ソフトウェアによって形成されるようになってきています。

当産業の強みである高品質で、信頼性の高いソフトウェアは、社会及びビジネスのインフラとして大いに力を発揮するはずです。

## エコシステムによる貢献

情報処理推進機構 (IPA) ソフトウェア高信頼化センター (SEC) が、ソフトウェア・エンジニアリングセンターとして設立された当時から、JISAはSECと密接な連携を図ってきました。それぞれの会合に相互に委員派遣を行ったり、SECの成果を現場で活用してきました。ソフトウェア開発の基盤となる技術や統計データは、SECから数多く入手してきたと思います。

今後はこの連携を一層深めて「エコシステム」化していかなければいけないと思います。エンジニアはSECで理論を学び、実証実験を行い、自らの知見とする、その知見を持って現場に戻り実践する、この理論と実践の繰り返しから、ソフトウェア開発の現場はスパイラルアップするのだと思います。

デジタルビジネス時代の社会は、価値観が複雑で高度な知識を要求される社会でしょう。このようなエコシステムを通じて一人ひとりが成長し、社会の高度化に貢献していきたい、SECもJISAもその役割は大きいと思います。

### 【脚注】

※1 経済産業省、中間取りまとめ～CPSによるデータ駆動型社会の到来を見据えた変革～  
[http://www.meti.go.jp/committee/sankoushin/shojo/johokeizai/pdf/report01\\_02\\_00.pdf](http://www.meti.go.jp/committee/sankoushin/shojo/johokeizai/pdf/report01_02_00.pdf), 2015年5月

# 車載ソフトウェア開発の今後の方向性

株式会社デンソー 技監  
村山 浩之



SEC 所長  
松本 隆明

今、自動車は急速に高性能化・高機能化、更にコネクテッド化され、外部とつながるひとつの大規模システムになりつつある。「自動運転」も大きな話題だ。これらを実際にコントロールしている車載システムも大規模・複雑化が進んでいるが、この車載システムの開発の現状はどうなっているのか。あるいは今後の開発はどのように進むのか。その最先端を担う村山氏からお話を伺った。

## ■自動車は走るコンピュータ

**松本：**車載システムというのは、ほとんどソフトウェアで動いていると思って良いのでしょうか？

**村山：**歴史的に見ると、自動車にソフトウェアが導入されたのは、1970年代初めのアメリカで行われたマスキー法による排ガス規制がきっかけです。従来のメカでは制御しきれない、同時にマイコンが自動車の中でも使えるようになってきたということがあって、ソフトウェアで制御することが考えられるようになりました。

ご質問の、どのくらいソフトウェアで動いているのか、ということですが、例えばある自動車メーカーの人は、現状の電子部品の比率は部品全体の40%だと言っています。電子部品のすべてがソフトウェアではないと思いますが、ニアリーコールだと思えば、この40%が目安でしょう。ハイブリッド車になると、

比率はもっと高まって半分くらいになると思います。

**松本：**昔はブレーキを踏むと油圧でブレーキが作動しました。しかし今は、踏まれたペダルの角度などを検出して、ブレーキをソフトウェアでコントロールすることが行われていますね。

**村山：**それも排ガス制御と同じように、メカでやってきたことをエレクトロニクスに置き換える、ということです。いわゆる運動系のソフトウェアによる制御というのは、全体としては歴史が古い。それが最近になって、カーナビもそうですが、もともとなかった機能を追加するものが現れ、これはほとんどソフトウェアで行っています。現在では、高級車には100個以上のマイコンが搭載されていて、自動車は走るコンピュータと言われる時代になっています。

## ■普遍的な仕様をまず見極めたい

**松本：**車載ソフトウェアの開発の特徴というのは、どういう点ですか？

**村山：**わかりやすいのはリアルタイム性ということでしょう。リアルタイム性には、ハードリアルタイムとソフトリアルタイムの2つがあります。これは、反応が早いか遅いかではなくて、どれほどの厳密さが求められるかどうかということです。自動車の特性上、リアルタイム性は非常に厳しく求められます。また、品質の確保も重要で、品質特性の中で以前は、信頼性や効率が重視されてきましたが、ソフトウェアの規模が大きくなって開発が大変になってきたことで、保守性や移植性も求めら



村山 浩之 (むらやま ひろゆき)

1980年 日本電装株式会社(現 株式会社デンソー)に入社。幅広く電子システムの技術企画・研究開発・製品設計に従事した後、1996年プロジェクトを立ち上げ、車載ソフトウェアの改革活動を開始。以後一貫して、車載の電子システムやソフトウェアを中心に、全社横断活動や自動車業界における標準化活動を推進。2001年 ソフトウェア基盤開発室室長、2008年 電子プラットフォーム開発部部長、2008年 常務役員を経て、2012年 技監に就任、現在に至る。

れるようになっていきます。

話を車載ソフトの特徴に戻すと、申し上げたようにリアルタイム性や諸品質があるのですが、自動車ということで少しややこしくなるのは、エンジンやブレーキを制御するようなソフトウェアの特徴とナビのようなものでは、求められるリアルタイム性も違うし使えるリソースも違うということです。自動車というひとつの商品のソフトウェアでありながら色々な特徴が混在している。これが車載ソフトウェアを複雑にさせている要因です。

**松本：**保守性ということも大きな問題でしょうね。パソコンならウィンドウズアップデートなどのように適宜更新することが可能ですが、自動車の場合、一度組み込んだソフトをタイムリーにアップデートすることは考えられません。

**村山：**保守性については色々な見方があると思いますが、開発する立場からすると、自動車のバリエーションは色々な組み合わせで相当広がっていきます。いかにうまく開発するかという意味では、最初に保守性を高めておかなければいけない。例えば自動車そのものの車格の違いがあります。また載せる物、代表的にはエンジンですが、この組み合わせがあり、更に各種のオプションがあり、どの地域向けかという仕向性の違いもある。似たようなソフトウェアだけれど、少しずつ違うということになります。そこが悩みどころであり、知恵の使いどころでもあるわけです。

**松本：**バリエーションの数は大変なものでしょうね。

**村山：**最初に何が普遍的な仕様で、何が変わるのか、そこをどれだけ理解できるかが大切だと思います。

## ■しっかりしたアーキテクチャが必要

**村山：**その意味で最近力を入れているのがアーキテクチャというテーマです。最初に、いかにしっかりしたアーキテクチャが作れるか。

**松本：**車種ごとにアーキテクチャが決まっているのでしょうか？

**村山：**私たちは普遍的なものを作りたいわけです。極論ですがデンソーの立場でいうと、お客様は世界中の主な自動車会社ですがそれも共通にしたい。ただし搭載する自動車もエンジンも装備も違います。ですから最初に本質的なアーキテクチャを作る。違いというのは簡単にいうと実装上の違いであって意味的には同じはずなんです。ところが、それをちゃんと考えられる人間が非常に少ない。

**松本：**エンタープライズ系でも、今はアーキテクチャが

分かる人間が少ないですね。

**村山：**「アーキテクトは育てられない、発掘するのだ」というのが私の基本的な考え方です。その発掘の手段として、電子システム人材という制度を作って若手になるべく距離のある仕事を2、3年ずつ経験させるようにしています。私が考えるアーキテクトの一番の素養は概念化力、抽象化能力です。それには仕事を大きく見渡せるような距離感がないとダメなんです。このような環境において、アーキテクトに必要な思考ができるかを見て発掘するようにしています。

**松本：**抽象化力とか概念化力というのは、本人の持って生まれた素質のようなものも大きいのではないですか？

**村山：**その通りです。例えば、小説を書くなら誰も一行目からいきなり書こうとはしません。まず全体の構成を考えます。ところがソフトウェアでは、不思議なことで開発者はみんな一行目から書きたがる。全体構成のイメージが最初からぱっと浮かぶという人間は非常に少ないですね。

**松本：**最近はモデルベースのように全体をとらえたモデリングをもっと重視すべきだという議論がありますが、そういうところにもつながってくることでですね。

**村山：**モデルベースといっているながら自動車業界で取り組んでいる事例を見ると、往々にしてツールベースですね。本当のモデルではない。乱暴な言い方ですが、そのツールを使うためにソースをリバースしているといったこともあるように思います。

**松本：**まずソースベースで考えてしまって、それを図にしているということですね。自分の過去の経験を振り返ってもプログラムベースで考えているところがありますね。機能を実現するためにどうするか、アルゴリズムから考えてしまうので、全体の構成といったところには思い至らない。

**村山：**例えばコンピテンシーなどの議論をすると



**松本 隆明** (まつもと たかあき)

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

き、論理性は何をするにも必要ですが、概念化力というのはこれとは方向性が逆です。論理性というのは何かがあったときにそれを分解できる力。概念化というのは逆で、色々な要素があったときに「それって本質的に何だ」と上に上げる、いわば抽象化する力です。この両方ができるスーパーマンはなかなかいません。

## ■目的を見ずに手法から入ってしまう

**松本:**「オブジェクト指向」などというように、昔から考え方としては言われてきていますね。それがなかなか浸透していかないのは、作る方を優先してきた、という事情があるからでしょうか。

**村山:**さっきのモデルベースと同じで、手法ありきとなると、何のために、というところがおろそかになります。とくに日本ではまじめさが災いするとか、本来目的があってそのための手法なのに、手法だけを見てしまう傾向があります。研究者も、手法そのものの研究者はいても、上位概念からその適用を考えたりすることはできない。

**松本:**ソフトウェアの開発工程でいうと、もっと上流工程とか、そもそもそのソフトウェアを何のために開発するのかというところからまず考えるようにしていかないといけないということかも知れないですね。

**村山:**その上流という言葉が、社内で議論していて気がついたのですが、先ほどのソフトウェアプロダクトラインのようなイメージがあると、上流ってV字のラインの工程の上のほうではなくて、時間軸での初めの方のことなのです。機種がたくさんあるとなると、上流は展開が始まる前のところ、それぞれの機種展開の仕様を解釈して……という部分なのです。メカ屋さんと話をしていてなんか話が合わないなと思っていたら、上流の方向がちがうんだなと気付いたんですね。

**松本:**なるほど、でも上流というと私もV字の上の方を意識してしまいますね。

**村山:**実際、エンタープライズ系の方と話をしていると、一品ものが多いので、そのプロジェクトをいかに成功させるかというアプローチになっています。しかし、組込みの世界は、色々な機種展開がうまくいくようにと考える。そのプロジェクトそのものではなくて、色々な展開の仕事をうまくするための仕込みが重要になるんです。中でも一番重要なのがアーキテクチャだと思います。

**松本:**そのアーキテクチャというのは、プロダクトラインのひとつのコンセプトのようなものを考えるというこ

とでしょうか。それが大衆車なのか高級車なのか、どういう層を狙ったものなのか、そういうコンセプトから考えてアーキテクチャに落とし込んでいく。

**村山:**そうですね。かつそれは、世界中のどの自動車会社のどの装備にも対応できるアーキテクチャでなければならない、ということです。ひとくちに車載ソフトといっても色々な種類があります。エンジンなどの駆動系を制御するもの、外部環境の認識をするもの、人への対応に関するもの、エネルギーに関するものなどで、社内では便宜的に6群に分けています。それを実装とは違う論理アーキテクチャ、欧米ではファンクショナルアーキテクチャと呼んでいますが、要するに物ではなくて機能、意味のかたまりとして考えていくことが必要なんです。この点がメカの人となかなか話が合わないところですね。メカの方は当然ですが、物ありきですからね。

例えば駆動系を扱う「パワートレイン制御」系では、ひとつの見方ではこれは走るための動力源です。しかし、別の側面で見るとエネルギーの源なんです。これが顕在化するのには電気自動車です。電気自動車になるとエネルギー源に困る。電気自動車のパッケージの中では、貴重な走るためのエネルギーを、人を暖めるために使わなくてはなりません。今までそれを考えていなかったのは、エンジンが熱を出して、それを担っていたからです。しかし電気自動車ではできない。エネルギーはもっぱら走るために使って航続距離を伸ばしたいのに、エネルギーの一部を人の快適さの確保のために使わなければなりません。そもそもパワートレインとは何か、ということに答えなければいけない。これが論理なんです。

## ■自動車がつながる時代のアーキテクチャ

**松本:**実際にはどうやって検討していくのですか？一人のスーパーアーキテクトが引っ張っていくのでしょうか？

**村山:**現実的にはアーキテクトの素養のあるメンバーが目的という視点で試行錯誤することになります。例えば、今世の中で言われている安心安全ということを取ると、自動車を安心安全な物にするために何をしなければいけないか——それは今走っている自動車の装備とか来年あたりに出るものがどうかということとは別次元の話です。論理的なアプローチをしないとアーキテクチャは決まらないんです。一方で、それがある程度できたとき

に、今の車の装備はどうかとマッピングしてみる。そうすると、論理的な構造やインターフェースがすっかりするだけでなく、本来この機能があるべきだという商品企画にもつながります。

**松本：**アーキテクチャを考えるのはいっそ技術屋でないほうが良いのかも知れないですね。本来自動車は社会の中でどういう意味を持っているのか、といったことを考えられる人のほうが良い発想が出るのかも知れない。

**村山：**自動車が色々な物につながる時代になっているので、そういうことまで考えざるを得なくなっているのも事実です。自動車のつながる先まで対象にしたアーキテクチャを考えるためには、我々自身が従来の考えを飛躍させる事が必要になってくるかもしれません。自動車そのものの目的を考え、その目的次第でモビリティ社会として従来の自動車やシステムに代わる別な手段でも実現できるのではないかと考えるということです。

**松本：**確かに単なる移動手段と考えれば、鉄道でもなんでもあるので、そうした手段と何が違うのかを明確にしていく必要があるということです。

**村山：**今話題の自動運転でもそういう議論が必要になっていくと思います。自動運転の話題が活発になっているのはIT業界の動きが刺激になっているようですが、自動車業界ではあくまでも主体は運転者であると考えています。

**松本：**自動車業界としては、自動車は単なる移動手段ではないと言いたいわけですか？

**村山：**IT業界が考えている自動運転の車というのは、もはや今の形である必要はないでしょう。単なる箱で良い。運べる箱。しかし自動車業界としてこれは受け入れ難いと思います。

**松本：**使う側の嗜好にもよるとと思います。単なる移動手段で良いという人は、その間別のことをしたいと思うでしょうし、自分で運転した方が楽しいと思う人もいます。

**村山：**多様化するのだと思います。

## ■ プロセス品質とプロダクト品質

**松本：**ところでお話の自動運転というのは、今後どうなるのでしょうか？本当に無人運転までいくのでしょうか。

**村山：**今考えられる方向性は2つあって、ひとつは自動車専用道路のようなところで、ゲート to ゲートで自動運転をするというものです。これは技術的にはもうかなりのレベルまでできていると思います。不慮の事態をかな

り限定して考えることができますからね。もうひとつは非常に小規模な地域で、コンパクトな、例えば高齢者のための足といったようなことで走らせる。当面はこの両面で実証的な実験をしていくということになると思います。しかし、最終的に実現の鍵を握っているのは社会的なコンセンサスだと思います。自分が運転していて対向車が無人だったら気持ち悪くないですか？

**松本：**それは嫌ですね。

**村山：**ですからかなり時間がかかると思います。この点では自動車専用道の方が考えやすい。商用車で隊列を組んで運転する実験はやっていますし、技術的にはかなり実現できています。

**松本：**私も筑波でトラックの隊列走行の実験車に乗せていただいたことがあります。前の車の7, 8m 後ろから80km くらいのスピードでついて行く。非常に近いので怖かったですね。

**村山：**怖いですよ。「安心安全」という意味で、あの距離感での隊列走行はとても安心とはいえない。そもそも自動運転も本当にぶつからないというだけなら、色々なやり方があります。しかしそれが人間にとってどうなのか、となると難しい点があるわけです。

**松本：**社会に受け入れられるものにしていかなくてはいけないということです。

**村山：**今のいわゆる運転支援系の機能にしても、安全のためには頻繁にウォーニングを出せば良いんですが、これは人間系が入ってくるとかえって危険になります。

**松本：**慣れてしまうと無視してしまったりする。しかし、自動運転は既に国の成長戦略にも明確に組み込まれていますから、国としても力を入れていくことになるでしょうね。

**村山：**方向性としては自動運転の方向に進むのは間違いないでしょう。そもそも自動運転は、通常時の機能を考えるだけなら従来の延長線上の技術でほとんど実現できると思います。どれくらいのことまで想定するのか、ということが悩みどころというか本質のようなところですね。

**松本：**それはエンタープライズ系の開発でも同じで、正常系の部分というのは処理的にはほんの一部でしかない。普通にこの機能を実現しろというだけなら、それほどたいしたプログラムにはなりません。何か想定外のことが起きたときにどう対処していくのか、その処理の仕組みを考えていくということが開発のかなりの部分を占めますね。

**村山：**とくに自動車の場合、温度、振動、ノイズなど電

子機器が使われる場所としては最悪の環境にあります。ノイズでビットが化けたり、何が起こるか分からない。だから色々な手を打っている。

**松本：**そのきめの細かいところは日本の開発ならではの知らない。

**村山：**日本の場合は、文書で定義されていないことでも現場の力で手を打っていくことが多いと思います。要するに品質にかかわることですが、日本では、プロセスの品質は説明できないけれどもプロダクトの品質が良い。

**松本：**すり合わせできちんと作っているから結果として高品質なプロダクトとなる。

**村山：**日本の特徴として、ほぼ単一の民族で「なあなあ」の世界で、わりあい共通善のような考え方もある。ところが欧米はみんなつながっていて基本的にはお互いが信用できない。何で守るかといったら契約とかプロセスしかない。

**松本：**つまり日本の場合は共通の暗黙知みたいなものを持ちやすいが、欧米ではそれができないから、プロセス単位で考える。すると、最後に物として仕上げたときにそれがだめだったりするケースが出てくるということでしょうか。

**村山：**加えて自動車の場合、今までは分野で閉じていた仕事が多かったんです。ところがそれがネットワークで結ばれるようになり、色々な連携をするようになると、そこでは欧米スタイルが必要になってくるんです。最初にインターフェースを決めないといけない。会話する言葉も違っているわけですからね。今後は更に外につながる。正にそれが始まっています。今までの日本のやり方をうまく取り込まないといけないし、全体としては最初に欧米風に整理しておかないといけない、ということになってきます。

## ■国際標準化に向けて

**松本：**そうすると標準化ということがもう少し重要になってくるということですか？

**村山：**2種類あると思うんです。1つは仕事をする上でつながるようになってくるから、そのインターフェースをなんとかしなければならない、ということ。もう1つは、つながるようになった領域の仕事を誰が担うのかとなったときの事業的な観点。簡単に言えば、守りの標準化と攻めの標準化です。自分の領域を守るためにきちんと決めましょう、ということ、新しい世界のための標準化ということ、2つになる。

**松本：**守る世界の方が考えやすいですね。自分の作ったものはこの標準に基づいてやっているからOKと。日本は攻めの標準化というのはあまり得意ではないでしょう。

**村山：**そこが強いのは圧倒的に欧州、とくにドイツです。とくにデジュールとしての進め方。先ほどの話に関係しますが「何のために」ということがしっかりとあるんです。世のため人のためという「共通善」がある。そこから下には理屈があって、自分の国の産業に落ちていく。ところが日本にあるのは「暗黙知」です。世のため人のためなんて当たり前と思って誰も口にしない。だから日本場合は、いきなりハウツーから始まるわけです。そのため途中で「何のためだっけ？」となったりする。

**松本：**日本は海に囲まれた島国で、みんな共通認識みたいなものがあるから、当たりの世界で、そこからどう落としていくかとなる。EUなんかは陸続きにもかわらなく各国バラバラですからね。それをどうやってまとめていくか。

**村山：**言葉にして形式知にしていくんですよ。そうしなければならぬ事情がある。一方、北米は、先行したものの勝ちのデファクトスタンダードです。日本人が気持ちの上で合うのは欧州の方ですね。

**松本：**日本人はきっちりやっていく世界ですね。そういう教育も受けています。

**村山：**標準化が大事だという話はようやく日本でも議論されるようになって、自動車業界ではJASPARという活動や、ISOのTC22という自動車分野で四輪自動車領域で初の幹事国となり、議長を出すといったことが始まっています。

**松本：**良いことですね。そういうことをしていかないと、いつまでも欧米の決めたものに従って右往左往することが続きかねない。

**村山：**今までは欧米基準の枠の中でもプロダクトの品質で最後に勝てたんです。やり方は欧州流で、最後は品質で勝てた。しかし、これだけ仕事が増えて、しかも車の外につながるようになったら、もう今まで通りにはやれないと思います。

## ■車載システムから見た「Industrie4.0」、「CPS」

**松本：**標準化という意味で言うと、ドイツではIndustrie 4.0ということを使い始めたり、あるいはIoTを使ってなるべく製造工程も自動化して効率化を目指していこう

という動きがありますが、それについてはどう考えていますか？ CPS (Cyber-Physical Systems) という言葉も出てきています。

**村山：** Industrie4.0 をどう見るかということは社内でもまだ共通認識にはなっていません。まだ製造の範疇なんです。Industrie4.0 で言っているコンセプトはもっと広いはず。例えば IoT にしても CPS にしても、どのスコープでどういう目的で、何をしたいのかということをお互いに前提として決めないと、議論がずれ違ってしまいます。組込みというのは小さい世界の CPS といえます。とくに車の場合を考えると、最初のエンジン制御もそうですが、今までメカでやっていたものにサイバーを入れた。つまり小さい世界では CPS は当たり前であって、それが広がっているだけの話です。では何を持って新しいことと捉えるか、というのは、立場によってそれぞれ見方が違う。もの側、組込み側から見るか、サイバーの人から見るかによっても全く違うでしょう。様々な情報が色々使えるようになったね、というだけかも知れないし。

**松本：** 私の理解でいえば、小さい世界では昔から CPS という世界はあったが、外とつながる今の CPS とはコンピューティングパワーが全然違うのだと思います。今までチップとか ECU 単位で処理をしていた世界が、外のハイコンピューティングパワーを使えるようになることによって、様々な新しいことができ、分析もできてフィードバックすることができるということが大きいと考えています。

## ■「つながる時代」のセキュリティの重要性

**松本：** その際、問題になっているのはセキュリティですね。色々なものがつながってくると、どこに脆弱性があるか、攻撃をかけられるかわからない。

**村山：** その問題について、デンソーの中にも専任の組織を立ち上げました。自動車業界でも情報セキュリティは今最大のテーマになっています。自動車業界では以前から、各企業が品質に対する管理システムを持っている。その上に、説明責任的な観点も加えて、機能安全のシステムを追加した。今は更にこの上に、情報セキュリティの仕組みを加えているというイメージです。車にとっての脅威として一番大きいのは人命を代表とする安心安全です。そのため、つながる系の方では、世の中の IT 系

の一番良い物を持ってくる一方で、安心安全の方はきちんと車の中で考えようと段階的に取り組んでいます。

**松本：** つながる世界でいえば、ナビがアタックされて自動車そのものの安心安全にかかわるという可能性もありますよね。

**村山：** もちろんそこはファイアウォールのものがありますが、とくにつながる系では、その時々最高の技術をいかに取り込めるかというところがポイントだと思いますね。

**松本：** ただし、アタックと防御は“いたちごっこ”の面が否めないですね。そこで最近セキュリティの世界でいわれているのは、完全に防ぐのはもう無理だと、だから万一侵入されても、影響範囲を限定できるようにするか、重要な中心部は絶対に侵害されないようにするか、そういった考え方になっています。

**村山：** 今日話に出てきたアーキテクチャの話はセキュリティにとっても重要なテーマなんです。アーキテクチャがしっかりしていればものすごく楽だからです。機能安全の時もそうでしたが、意味的に固まらずに分散していると非常に大変になります。色々なところに色々な仕掛けをいれなければならなくなる。セキュリティも全く一緒で、つながる系でも、ここでちゃんと遮断できる、となっていれば楽なんです。

**松本：** つながる自動車になってくると、ますます色々なケースを考えなければならなくなる。一対一のコンポーネント間ではだめで、通信路全体を考えないといけない。やはりアーキテクチャが重要ということですね。

**村山：** その通りだと思います。私はそれにここ 10 年くらいずっとこだわっていますし、その真価が問われるときだと感じています。

**松本：** そうですね。本日はお忙しいところありがとうございました。



# チェックリストを用いたコードレビューと判別モデルの組合せによるモジュールの不具合リスクのランク付け



笠井 則充<sup>+1</sup>



森崎 修司<sup>+2</sup>



松本 健一<sup>+3</sup>

fault-prone モジュールのランク付け精度を向上させる手法を提案する。ランク付けには fault-prone 判別得点として、判別モデルにより得られる判別得点とコードレビューの評価得点を組合せた値を用いる。まず、判別性能が高いモデルを用いてコードレビューによる精度の向上を評価する。次に精度をできるだけ維持したまま工数を削減する3つの方法を提案する。削減方法は、1：構築コストの小さい判別モデルの利用、2：レビュー対象のモジュール数の削減、3：一部のレビュー観点の省略、である。これらの結果、従来の判別モデルにコードレビューによる評価得点を組合せることで fault-prone モジュールのランク付け精度が向上することが判った。また、レビュー工数の削減については、ランク付け精度を維持したまま 46% の工数を削減できることが判った。

## An Approach for Prioritizing Fault-Prone Modules by Combining a Discriminative Model and Code Review with Checklist

Norimitsu Kasai<sup>+1</sup>, Shuji Morisaki<sup>+2</sup>, Kenichi Matsumoto<sup>+3</sup>

This paper proposes an approach for improving ranking accuracy of fault-prone module by combining discriminative model and code review. First, we evaluate ranking accuracy using support vector machine and random score. Next, we propose three ways to reduce the man-hours, I: using low cost discriminative model, II: reducing modules to review, III: using two aspects of the review to omit parts of review items. The result shows all accuracies are increased by combining code review in trials, and review effort is decreased by 46% without decreasing detection accuracy.

### 1. はじめに

顧客から受注したシステム開発のうち、ソフトウェア開発の一部を外部委託する業務形態において、委託先で開発された成果物に対して委託元がこれを検証するプロセスが重要視されている。委託元ではV字モデルに従い、受注したソフトウェアの要求仕様や外部装置とのインタフェース仕様を策

定し、内部設計やコーディング、単体／組合せ試験を委託先に依頼することがある。委託先から成果物を受領後、システ

#### 【脚注】

- + 1 三菱電機株式会社 通信機製作所
- + 2 名古屋大学
- + 3 奈良先端科学技術大学院大学

ム統合試験として第三者検証を実施し出荷する。第三者検証中に不具合が多発した場合、限られた工期の中で可能な対策としては人員や時間を追加投入するクラッシング (crashing) しか無く、修正や再テストなど作業者の負担が増大する場合がある。これを避けるために委託元が成果物を受領した段階で設計検証を行い、不具合を可能な限り解消した上で第三者検証であるシステムテストを品質管理部門が実施するというプロセス改善が進んでいる。

不具合を含んでいる可能性のあるソフトウェアモジュール (fault-prone モジュール) に対して、モジュールが不具合を含む度合いを得る方法として fault-prone 判別モデルが知られている。判別モデルでは行数やサイクロマチック数といったソースコードメトリクスや、内部仕様書のレビュー指摘件数のようなプロセスメトリクスといった特性値群を入力データとして用い、モジュールが不具合を含む度合いを出力する。これが基準値を超えるとそのモジュールが不具合を含むと推定する。

但し、「コメント行に記述された説明と実行ステートメントが一致していない」、「プラットフォーム依存の実装となっている」といった、潜在的な不具合の存在や兆候をそれら特性値群から捉えることは困難であり、ソースコードメトリクスやプロセスメトリクスから得た判別モデルだけで判別精度を高めることには限界がある。

このような、不具合の兆候となる情報はソースコードを直接確認するコードレビューによって検出することが可能である。しかし、コードレビューには多くの時間・工数が必要であり、実用規模のソフトウェアにおいて全モジュールに対してコードレビューを実施することは現実的ではない。著者らは文献 [kasai2012][kasai2013] において、コードレビューと判別モデルを組み合わせた場合の fault-prone モジュールのランク付け精度を学術面において議論してきた。本論文では、次の順序でコードレビューを組み合わせることの有効性を示し、開発現場での実践を前提としてこれにかかる工数の削減を試みる。

まず、従来の判別モデルにコードレビューによる評価を加えることで、判別モデル単独の場合よりも fault-prone モジュールのランク付け精度が向上するかをケーススタディによって示す。コードレビューの主目的は網羅的に欠陥を検出することではなく、ソースコードメトリクスやプロセスメトリクスからは得られにくい情報を補完的に得ることである。

レビューによりランク付け精度が向上すると判った場合は次の段階として、レビュー対象のモジュールを選択すること、レビュー手順の簡略化による評価工数の削減の効果、より低コストで導入が可能な判別モデルの導入を試行し、それぞれランク付け精度を評価する。

以降、2章で判別モデルにコードレビューによる評価を加える方法とケーススタディにおけるランク付け精度について述べる。3章では工数の削減方法を述べ、ケーススタディによるランク付け精度と工数により評価する。4章で考察を行い、5章でまとめる。

## 2. 判別モデルとレビューの組合せ

### 2.1. 概要

#### 2.1.1. 判別モデル

fault-prone 判別モデルは、入力をモジュール  $m$  のメトリクス、出力を判別結果  $F(m)$  とする。判別結果  $F(m)$  は、fault-prone の度合いである。任意の二つのモジュール  $m_i$  と  $m_j (i \neq j, i, j = 1, 2, \dots)$  に対して  $F(m_i) < F(m_j)$  が成り立つとき、 $m_i$  は  $m_j$  よりも fault-prone の度合いが大きいとする。

本論文では特定の fault-prone 判別モデルを前提としないが、モジュール  $m_1, m_2, \dots, m_n$  は判別結果によって並べられるものとする。

#### 2.1.2. コードレビュー

コードレビューは、ソースコードの作成者やそれ以外の第三者が、作成したソースコードに目を通し確認する作業 [IPA2005] であるが、本論文でのコードレビューは、過去の類似システムで発生した不具合を元にしたチェックリストや評価者であるエンジニアが過去の類似案件で経験した情報等に基づいて評価を実施することを前提としている。

モジュール  $m$  の評価は、後述するチェックリストに含まれる質問項目の集合と各質問項目に対応した得点  $I(m)$  によって与えられる。質問項目や採点手順は、そのアプリケーションの背景などの豊富な経験と知識を有した開発者によってカスタマイズされ決められたものである。

チェックリストは  $l$  個の質問項目  $q_i (i=1, 2, \dots, l)$  からなる。 $q_i$  により潜在的な欠陥の兆候となる可能性のある事項を検出する。各質問項目には採点手順が定められる。評価者は質問  $q_i$  とその採点手順によりモジュール  $m$  に対して得点  $s_{im} (-1 \leq s_{im} \leq 1)$  を与える。モジュール  $m$  に対するレビュー得点  $I(m)$  は、 $s_{im}$  の合計値を質問数  $l$  で割った値とする。

質問項目  $q_i$  は複数のモジュール間にまたがる評価は含まない。質問項目は、例外のハンドリング不足や、既存のソースコードのコメントの更新漏れといったコードスメル (不具合を含みそうなソースコードの兆候) [Fowler1999] によりモジュールを評価するものも含む。

#### 2.1.3. 組合せ

本章による fault-prone 判別得点  $\hat{F}(m)$  は、判別モデルとレビュー評価による得点の合計に等しい。すなわち、 $\hat{F}(m) = F(m) + \beta I(m)$ 。

ここで、 $\beta$  はレビュー評価点の係数である。 $\beta$  は、判別モデルの得点に対するコードレビューの評価点の重み付けであり、判別モデルによる得点とレビュー評価点のどちらに重みを置かずか実績をみながら調整する。初めて提案手法を適用するときのように実績がない場合には、判別モデルの得点とレビュー評価点を同程度 ( $\beta=1$ ) とする。

fault-prone 判別得点  $\hat{F}(m)$  によってモジュールを再度並び替える。

$$(m'_1, m'_2, \dots, m'_n), \hat{F}(m'_1) \leq \hat{F}(m'_2) \leq \dots \leq \hat{F}(m'_n).$$

並び替えた  $m'_t (t=1, 2, \dots, n)$  の順序は不具合を含む可能性が大きいと判別したモジュールの順序である。

## 2.2. ケーススタディ

通信システムを監視、制御し、10年以上に渡って保守、更新されてきたアプリケーションを用いてケーススタディを実施する。アプリケーションはC言語で記述され、125モジュールから成る。総行数は15,100行である。

ケーススタディ対象のプロジェクトは、ウォータフォールモデルによって実施された。コーディング、単体テスト、

組合せテスト、及び、システムテストの一部のフェーズを外部委託している。ソースコードは受入れ検査前の最終版である。受入れ検査の間に15個のモジュールに欠陥が検出された。それぞれの欠陥は修正され、修正モジュールが保管されている。

このシステム開発に従事した熟練技術者がコードレビューのチェックリストを表1のように定義した。チェックリストは、委託先に提示した要求仕様書に記載している機能要求と非機能要求から抽出した。機能要求からはシーケンスタイミングやその条件等からクリティカルな部分を抽出してチェックリスト化した。非機能要求からはISO9126で定められた品質特性からこのシステムに必要とされるものを記載した項目に基づいてチェックリストに反映した。これに加えて、過去に実施したコードレビューで検出された不具合やレビュー時の指摘事項から得られた確認項目、過去に経験した不具合から類推される問題点等を抽出してリストへ反映した。

判別モデルとして、従来研究で判別精度が大きいと報告されているサポートベクタマシン (SVM) [Boser1992] を

表1 チェックリスト

	分類	質問項目
$q_1$	実行環境依存の排除不足	カウンタを用いたループ処理による時間待ち。
$q_2$		OSが入れ替わったとき、使用できなくなる可能性のあるライブラリの使用。
$q_3$		変数を初期化せずに使用。
$q_4$		do ~ while文でwhile条件は有るがdoループ内が空。
$q_5$		for文でループ内以外では使用していないローカル変数をカウンタに使用。
$q_6$	例外処理の対応不備	適切な箇所で構造化例外処理を記述していない。
$q_7$		catchで適切な例外を個別に処理していない。
$q_8$		finallyがなく、defaultの処理が規定されていない。
$q_9$	デバッグ用メッセージ削除漏れ	デバッグ用printf、外部出力命令の削除漏れがある。
$q_{10}$	バッファオーバーフロー対策漏れ	get, gets, sprintf, strcat, strcpy, vsprintfを使用しているとき、バッファサイズの考慮が十分でない。
$q_{11}$	実装プロセス圧縮の悪影響	例外等による分岐によって、確保したメモリやリソースの解放処理が行われないことがないか。
$q_{12}$	可読性	分岐処理やループ処理、処理ブロックの先頭にその処理を説明するコメントがあるか。
$q_{13}$		switch文で整理できる分岐をif文で羅列していないか。
$q_{14}$		複雑な条件判断に下位関数の戻り値を使用しているか。
$q_{15}$		条件分岐後の複雑な処理を下位関数で定義しているか。
$q_{16}$	セルフチェック不足	コードを流用した場合、必要に応じてコメントも更新しているか。
$q_{17}$	実装作業時間不足の兆候	上位関数で実施すべき処理を下位関数で定義し、下位関数の引数を不必要に増やしていないか。
$q_{18}$	コーディングの適切さ	設計ロジックの複雑さに対してコードが必要以上に複雑になっていないか。
$q_{19}$		必要以上にif文が連続し見通しが悪くなっていないか。
$q_{20}$	記述粒度の一貫性	プログラミングのルールを逸脱していないか。
$q_{21}$		記述粒度のばらつきが無い。
$q_{22}$	環境依存の考慮不足	構造体をそのままファイル等に出力する場合、アライメントによる空のデータが混入しないか。
$q_{23}$		ログファイルの最大容量の考慮があるか。
$q_{24}$		ビッグエンディアン、スモールエンディアンの考慮漏れが無い。
$q_{25}$		必要な部分でvolatileの付加漏れが無い。
$q_{26}$		必要でない割り込みの利用が無い。
$q_{27}$		演算結果がオーバーフローする可能性が無い。
$q_{28}$		規定API/インタフェースを使わず実装していない。
$q_{29}$		ネットワーク通信やプロセス間通信に規定のAPIを使っているか。

用いる。サポートベクタマシンは最も判別精度が大きくなる fault-prone 判別モデルの1つであると報告されている [Sebald2000]。他のモデルに比べて優れたパターン認識結果を得られることが知られており [Schölkopf1997]，局所解に陥ることがなく，特殊な場合を除いて解は一意に定まることが指摘されている [Burgess1999]。また，ランダムな判別得点が得られるモデル (RND) を比較対象とする。評価指標は Alberg diagram [Ohlsson1996] の AUC (Area Under the Curve : 曲線下面積) を用いる。AUC は [0,1] の値をとり，fault-prone であると判別したモジュールに対して，実際に不具合が含まれているモジュールの割合が大きいほど大きな値となる。詳細は文献による。

モデル構築のためモジュール群を次のように2つのグループに分けた。モジュールを規模順に並べ，規模の小さい順に順位をつける。奇数の順位がついたモジュールをデータセット X とする。偶数の順位がついたモジュールをデータセット Y とする。

ケーススタディにおいて受入れ検査で検出された欠陥を含むモジュールの数は15モジュールであった。データセット X には7モジュールに，データセット Y には8モジュールにそれぞれ欠陥が有った。評価手順は以下の通り。

- (1) チェックリストから質問項目と採点手順を決める。
- (2) 以下のモジュールデータを用意する。
  - (a) 全てのモジュールのソースコードメトリクスを測定する。
  - (b) 以下のようにモジュールをデータセット X と Y に分割する。

$$X = \{M_1, M_3, \dots, M_{2i+1}, \dots, M_{125}\},$$

$$Y = \{M_2, M_4, \dots, M_{2i}, \dots, M_{124}\}.$$

データセット X は判別モデル (SVM) に与える学習データである。ここで，判別モデル (SVM) はデータセット Y のモジュールの fault-prone の度合いを判別する。逆もまた同様である。判別モデル (RND) は学習データを必要としない。

- (3) 判別モデル (SVM), (RND) の fault-prone の度合いに従ってモジュールを並べる。即ち，データセット X, Y について集合  $X_{SVM}, X_{RND}, Y_{SVM}, Y_{RND}$  を得る。

$X_{SVM} = (m_{SVM1}, m_{SVM2}, \dots, m_{SVM63})$  は判別モデル (SVM) によって並べられたデータセット X である。

- (4) すべてのモジュールについてチェックリストによって採点を行い，本論文のアプローチによる fault-prone 判別得点  $\hat{F}(m)$  を得，この昇順にモジュールを並び替える。
- (5)  $\hat{F}(m)$  の昇順に並び替えたモジュールに対してそれぞれ AUC を算出する。

モデル (SVM) の構築には，説明変数は McCabe のサイク

ロマチック数，ループや分岐の最大ネスト数，関数呼び出し数，コメントを含まない行数とし，目的変数は出荷判定中に検出された不具合の有無とした。モデル (SVM) の fault-prone の度合いは，統計解析ソフトウェア R を用いて構築したガウシアンカーネルによる非線形サポートベクタマシンを用いて算出した。モデル (RND) の fault-prone の度合いは，R によりメルセンヌツイスタ乱数を生成させて得た。

## 2.3. ケーススタディの結果

表2にそれぞれの判別モデルにおける最大の AUC とそのときの  $\beta$  の値 (複数存在する場合は最小値) を示す。表中，“SVM (レビュー無)”，“RND (レビュー無)” は判別モデルのみによって得られた AUC である。コードレビューを組み合わせることによって得られた  $\hat{F}(m)$  による AUC が，レビューを行わない場合と比較して有意に大きいかを有意水準 5% の片側検定による 2 標本 t 検定によって実施した。いずれの場合においても判別モデルとレビューを組合せたときの AUC がレビューを行わないときの AUC よりも有意に大きい結果であった。

表2 最大の AUC と  $\beta$  の値

判別モデル	データセット	AUC	$\beta$
SVM	X	0.831	1.6
	Y	0.854	1.4
RND	X	0.755	4.7
	Y	0.840	3.6
SVM (レビュー無)	X	0.774	—
	Y	0.773	—
RND (レビュー無)	X	0.512	—
	Y	0.515	—

適切なパラメータ  $\beta$  を与えることによって判別モデルのみを用いた場合と比較して，(SVM) のとき，データセット X では AUC が 0.057，データセット Y では 0.081 向上すること，(RND) のとき，データセット X では AUC が 0.243，データセット Y では 0.325 向上することが判った。ケーススタディでは最大の AUC が得られる  $\beta$  を求めた。 $\beta$  を 1 とした場合の AUC は (SVM) で 0.824 と 0.848，(RND) で 0.664 と 0.679 と小さくなるが，判別モデルのみを用いた場合の AUC と比較して大きな値であった。

## 3. 工数削減

### 3.1. 概要

本章では，予測精度をなるべく維持しつつ，前章で実施したランク付けに必要なコストを低減できるかを試みる。具体的には，コードレビューの対象となるモジュール数を減らし，モジュールの規模に応じてコードレビューのチェックリストの項目数を減らす。加えて，判別モデルの判別得点の代わりに単一のソースコードメトリクスを用いることで，モデル構築コストを低減する。

コードレビューの方法Ⅰ（対象モジュールの削減）：全モジュールをレビュー対象とするのではなく、レビューの効果がより大きいと思われる一部のモジュールを対象とする。2.1.3節で述べた fault-prone の度合い  $F(m)$  を、レビューすべきモジュールを選択する基準として用いる。レビューを行うモジュールの割合を  $\alpha\%$  と表記する。  $n$  個のモジュールに対して  $\alpha\%$  のモジュールにレビューを行うとき、対象のモジュールの個数  $k$  は  $(\alpha n/100)$  の床関数とする。また、fault-prone 判別得点  $\hat{F}(m)$  を拡張し、レビューを行わないモジュールに対しては評価得点  $I(m)$  を 0 とする。  $\alpha$  も  $\beta$  と同様にチェックリストと判別モデルの組合せの実績を見ながら決める。初めて提案手法を適用するときのように実績がない場合には、  $\alpha$  の増加に従って提案手法の fault-prone モジュールのランク付け精度が増加するという前提を置き、コードレビューに割り当てることができる工数によって予測時に決めることを想定している。

コードレビューの方法Ⅱ（レビュー観点の省略）：レビュー観点の省略により個々のモジュールに割り当てられるレビュー工数を削減する。前章で示したコードレビューでは、予め用意した質問項目全てを評価することを前提としているが、モジュールによっては質問項目の一部を評価するだけで十分な場合もある。そこで、モジュール規模に応じた観点を設定し質問項目を観点で分類する。モジュール規模によって省略する観点（質問項目）を決める。たとえば、コード全体を一目で見渡すことができるような小規模モジュールには該当しない観点を省略する。

判別モデル：既存の多くの fault-prone モジュール判別モデルでは、モデル構築のコストに加え、モデルを扱うための学習、教育、環境構築のコストが必要となる。特にモデル構築には過去の不具合がどのモジュールから検出されたかを収集しておく必要がある。そこで、多くの fault-prone 判別モデルの入力パラメータとなっているソースコードメトリクスを用いることによりモデル構築コストを低減する。

## 3.2. コードレビューの方法

### 3.2.1. 対象モジュールの削減（方法Ⅰ）

図 1 に判別モデルの結果を用いたコードレビュー対象モジュールの選択方法を示す。モジュールの選択方法は明らかではないため、4通りを考え、ランク付け精度と必要な工数を比較する。選択方法 1 においては、評価者は fault-prone の度合い  $F(m)$  が大きいモジュールから順にレビュー対象のモジュールをその割合が  $\alpha\%$  になるまで選択する。選択方法 2 においては、評価者は  $F(m)$  が小さいモジュールから順にレビュー対象のモジュールをその割合が  $\alpha\%$  になるまで選択する。選択方法 3 においては、評価者は  $F(m)$  の中央値のモジュールから順にレビュー対象のモジュールをその割合が  $\alpha\%$  になるまで選択する。選択方法 4 においては、評価者は  $F(m)$  が最大のモジュール、最小のモジュール、

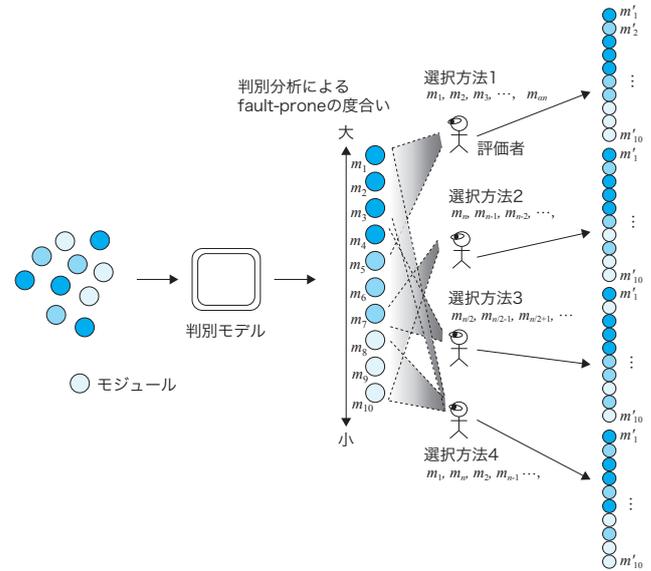


図 1 方法Ⅰによるレビュー対象の選択方法

最大から 2 番目のモジュール、最小から 2 番目のモジュール、とモジュールの割合が  $\alpha\%$  になるまでのモジュールをレビュー対象として選択する。

選択方法 1 と 2 は、それぞれコードレビューが  $F(m)$  の偽陽性、偽陰性を補完することを期待している。選択方法 3 と 4 は  $F(m)$  がとりたてて注目されていないモジュール、及びその逆のものに着目してコードレビューの効果を期待している。

### 3.2.2. レビュー観点の省略（方法Ⅱ）

コードレビューに用いる質問項目は 2 種類の観点から設定する。すなわち、観点 (1) 局所的な目視で不備や不具合の存在自体を即座に発見できる質問項目、観点 (2) モジュール全体にわたって整合性等を確認し、潜在的な不具合を予測しようとする質問項目、である。観点 (1) で直接的な不具合を予測すること、観点 (2) で問題の兆候を予測することを想定している。観点 (2) による評価は規模の大きなモジュールでのみ実施する。観点 (2) の質問項目はモジュール内の一貫性の不備に起因する不具合の兆候を検出しようとしており、規模の小さいモジュールではそのような問題が起きにくいからである。観点 (1) の例を表 3 に、観点 (2) の例を表 4 に、それぞれ示す。

レビューの具体的な手順を図 2 に示す。この手順は 1 つのモジュールに対するものであり、レビュー対象のモジュールの数だけ繰り返す。

まず、観点 (1) によるレビューを実施する。問題があると判断されれば、そのモジュールには不具合や問題が含まれることを意味するので、他の質問項目による評価を止め、評価点を -1 とし、そのモジュールの評価を終了してモジュール内の他の不具合の可能性を考慮した施策対象とする。観点 (1) による評価において問題がないと判断されたモジュールのうち、規模の小さいモジュールは評価点 1 を付与し、

表 3 方法 II によるレビュー観点 (1) の例

分類	質問項目
環境依存の考慮不足	プロセッサ処理速度に依存した時間待ち処理 (ハードウェアのリプレイス時に本来の時間よりも短い時間で処理が終わる)。
例外処理対応不備	例外の発生する可能性のある箇所での例外処理が未定義。
デバッグ用メッセージ削除漏れ	printf, beep 等, 実行通知のための外部出力命令の削除漏れ。
バッファオーバーフロー	バッファオーバーフローが発生する可能性の大きい関数を使用しているとき, バッファのサイズをチェックする等の対策をしていない。

表 4 方法 II によるレビュー観点 (2) の例

分類	質問項目
セルフチェック不足	コメント不備 / 更新忘れがないか。
	コーディングルール非準拠箇所がないか。
例外処理の検討不足	例外処理の一貫性 (同一の例外に対して同一の例外処理) を維持しているか。
考慮不足	異常終了時等のログ出力情報は一貫しているか。外部からの入力に一貫したサニタイズを実施しているか。

そのモジュールの評価を終了する。

観点 (2) は含まれる全ての質問項目について評価を実施し, モジュール毎に “問題なし”, “問題あり”, “該当なし” を判定し, それらの数によって評価点を決定する。問題ありと問題なしに該当する質問項目数を比較し, 数が多い方を結果とする。表 5 に評価点とその付与条件を示す。なお, 方法 II ではモジュールによってレビューを途中で終了する場合があります。判別得点とレビュー評価点を合算しないため,  $\beta$  は使用しない。

### 3.3. ケーススタディ

前節のコードレビューの方法 I (対象モジュールの削減), 方法 II (レビュー観点の省略) によるレビュー工数の削減, 判別モデルの判別得点に代わり単一のソースコードメトリクスを用いた方法の有効性を確認する。対象とするアプリケーションは 2 章と同じものを用い, 使用した判別モデル (SVM と RND) やチェックリストは 2.2 節及び 3.1 節と同じとする。

方法 I (対象モジュールの削減) において, レビュー対象とするモジュールの選択方法は 3.2 節の 4 種類の方法とする。方法 II (レビュー観点の省略) において, 観点 (2) の対象は 70 行以上のモジュールとした。70 行は, 対象ソフトウェアの開発においてディスプレイにソースコードを表示したとき, スクロール無しで見通すことができる行数を想定しており, これを超えるものに一貫性の不備が混入しやすいと考えた。

表 5 方法 II による観点 (2) の評価点と付与条件

評価点	付与条件
1	“問題あり” が無く, “問題なし” がある。
0	(i) “問題あり” は有るが, “問題なし” の方が多い。 (ii) “問題あり” も “問題なし” も無い。
-1	“問題あり” の方が “問題なし” よりも多い。

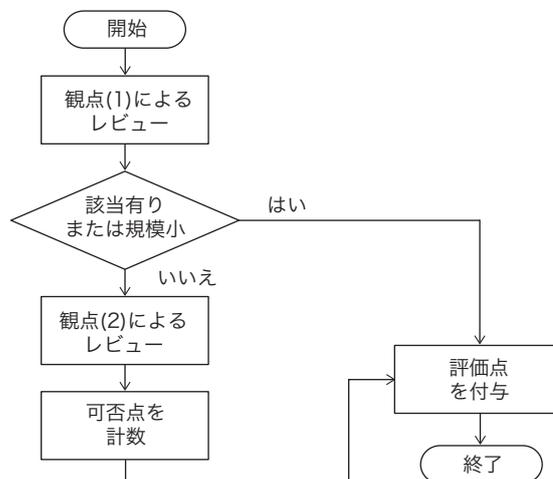


図 2 方法 II によるレビューの手順

判別モデルの判別得点の代わりにソースコード行数 (LOC) を選んだ。モジュール  $m_i (i=1, 2, \dots)$  に対する  $F(m_i)$  を LOC に (-1) を乗じた値とした。LOC を選んだ理由は, 多くの fault-prone 判別モデルの入力パラメータであること, 先行研究により LOC と複雑度メトリクスとの相関が大きいこと [Basili1984], 複雑度メトリクスが大きいと不具合が含まれやすいこと [Ohlsson1996], LOC と不具合に相関があること [Selby1991] が報告されているからである。

### 3.4. ケーススタディの結果

表 6 に最大の AUC 及び  $\alpha$  と  $\beta$  の値 (複数存在する場合は最小の  $\beta$  の値とそのときの  $\alpha$  の値を示す) を示す。表において, 方法 I の列はレビュー対象とするモジュールの選択条件を 4 通り試行したときの AUC が最大となるときの値とパラメータを示し, 方法 II の列は, レビューを観点 (1) と観点 (2) に基づいて実施した結果である。すべての選択方法とモデルにおいて, 方法 I の AUC の値のほうが方法 II の AUC の値よりも大きくなった。RND を除くと, 差の最大は 0.059 であった。

方法 I (対象モジュールの削減) において, レビューを行うモジュールの割合 ( $\alpha$ ) は 1-SVM のデータセット X を除いていずれも 70% 以上である。3つの判別モデルとレビュー対象モジュールの選択方法 1 ~ 4 において AUC が最大となるのは, 判別モデルが LOC で選択方法 4 のとき 0.871, このとき  $\alpha$  は 70% である。次に AUC が大きい組合せは, 判別モデルが SVM で選択方法 1 のとき 0.860, このとき  $\alpha$  は 40% である。

表 6 最大の AUC とパラメータ

選択方法とモデル	データセット	方法 I			方法 II	
		AUC	$\alpha$	$\beta$	AUC	$\alpha$
1-SVM	X	0.860	40	1.6	0.840	65
	Y	0.854	85	1.4	0.850	65
1-LOC	X	0.826	100	5.9	0.814	75
	Y	0.788	100	5.2	0.771	95
1-RND	X	0.852	95	4.7	0.740	95
	Y	0.840	100	3.6	0.817	100
2-SVM	X	0.831	100	1.6	0.819	100
	Y	0.863	95	2.2	0.854	95
2-LOC	X	0.833	95	4.2	0.812	95
	Y	0.788	100	5.2	0.771	100
2-RND	X	0.755	100	4.7	0.748	95
	Y	0.846	95	5.7	0.819	95
3-SVM	X	0.831	100	1.6	0.831	75
	Y	0.858	90	1.5	0.854	90
3-LOC	X	0.826	100	5.9	0.812	90
	Y	0.788	100	5.2	0.771	95
3-RND	X	0.845	90	3.0	0.760	95
	Y	0.840	100	3.6	0.817	100
4-SVM	X	0.852	85	1.6	0.819	100
	Y	0.854	100	1.4	0.838	100
4-LOC	X	0.871	70	1.8	0.812	100
	Y	0.800	75	3.8	0.771	100
4-RND	X	0.755	100	4.7	0.738	100
	Y	0.840	100	3.6	0.817	100

方法 II (レビュー観点の省略) において, 規模の小さなモジュールに対する評価工数を削減させることにより評価に必要なコストを低減させた. 一般には,  $\alpha$  を大きくすると, より大きな AUC が得られることが期待されるが, その分, 評価に必要なコストが大きくなる. ケーススタディでは,  $\alpha=75\%$  のとき, データセット X では AUC が 0.840, データセット Y では 0.841,  $\alpha=100\%$  のときそれぞれ 0.819 と 0.838 であり, fault-prone モジュールのランク付け精度の差はほとんどなかった.  $\alpha$  を 75% とすることにより, 約 31 モジュールのレビューを省くことができる. 方法 II (レビュー観点の省略) では, 評価に要するコストは観点 (1) による評価で 1 モジュールあたり 0.16 [人時] 程度, 観点 (2) で 0.33 [人時] 程度であった. いずれの判別モデルにおいても  $\alpha$  を 75% とするのに必要な評価コストはほぼ 8 [人時] でよく, 現場への適用には無理が無いと考えられる. また, 方法 I (対象モジュールの削減) のうち,  $\alpha$  が 40% と小さな値となった選択方法 1 によるレビューコストと, 方法 II (レビュー観点の省略) の各判別モデルの場合を比較して  $\alpha$  が 25%, 50%, 75%, 100% のそれぞれの値 (人時) を表 7 に示す. いずれにおいても評価コストは方法 II のほうが方法 I よりも小さくなり,  $\alpha$  が 100% の場合で評価コストは約 46% 削減した.

表 7 各判別モデルにおける目視コスト [人時]

$\alpha$	データセット	方法 I	方法 II		
			SVM	LOC	RND
25	X	5.12	3.59	3.76	2.74
	Y	5.12	3.59	3.59	2.57
50	X	10.23	5.99	6.16	5.31
	Y	10.23	5.99	5.99	5.48
75	X	15.35	8.56	8.56	7.88
	Y	15.35	8.39	8.39	8.05
100	X	20.46	10.96	10.96	10.96
	Y	20.46	10.96	10.96	10.96

表 8 判別モデルの比較

項目	SVM	LOC	RND
予測精度	大	中	小
モデル構築作業と必要な知識	要	不要	不要
学習データ (過去の蓄積情報)	要	不要	不要
導入容易性	やや難	易	易

#### 4. 考察

判別モデルとレビューの組合せに関して考察する. 今回のケーススタディでは表 1 のチェックリストの質問項目には, 偽陽性がなかったものが含まれていた. 質問項目  $q_6$  と  $q_{16}$  は真陽性が 3 件, 偽陽性は 0 件であったことから, コードレビュー結果は判別モデルを補完し得ることが解る.

ケーススタディでは, SVM と LOC をランダムな順序と比較して評価した. 比較を表 8 に示す. SVM は他のモデルに比べて優れたパターン認識結果を得られることが知られている. モデルに与えるメトリクスなど, 過去のデータを持っている場合や, 判別分析が実施可能な環境にあれば, SVM によるモデルを構築することで高い精度での fault-prone モジュールのランク付けが可能となる. 一方で類似ソフトウェアによる蓄積情報などが必要だったり判別モデルの構築作業にある程度の知識が必要であったりするため, 現場への導入が容易とは言えない.

工数削減に関して考察する. コードレビューの方法 I (対象モジュールの削減) において, コードレビューを行うモジュールの割合が大きくなると概ね AUC が増大する結果となった. しかし, AUC は  $\alpha$  の増加に対しすべての場合において単調増加するとは言えず, 方法 I によってレビュー工数が削減できるとは必ずしも言えない. 原因はレビューの質問項目に含まれる偽陽性であり, こうした質問項目を減らすことが今後の課題である.  $\alpha$  の増加に対して AUC が単調増加すれば, コードレビューに投入できるコストに応じて  $\alpha$  を決めることができる. また, コードレビュー対象モジュールを選択方法 1 によって選んだときに  $\alpha$  の値が最小となった. レビューの質問項目は, 判別モデルに与える

ソースコードメトリクスでは捉えることが難しいものを前提とし、ケーススタディで用いたソフトウェアに求められる非機能要件に合わせてテラリングしたものである。普遍的なものではないので対象とするソフトウェアにあわせてテラリングする必要がある。

更なる工数削減に向けてレビューの質問項目の自動化の可能性を考察する。ケーススタディで用いた観点 (1) に含まれる質問項目には、OS 依存のライブラリの存在、do ~ while 文や for 文で想定される問題のように検出を自動化できるものや、変数の初期化漏れ、例外処理中の finally や default 処理の不足、デバッグ用の printf 文の存在、get や strcpy などのようなバッファサイズの考慮が必要な関数の使用といった、問題点の候補を自動的に列挙し、目視で確認できるものがある。これらの観点について、ケーススタディでは、自動化のための準備のコストのほうが自動化によって省略できるコストよりも大きかったため、コードレビューの質問項目としたが、将来のバージョンでの再利用等を加味すれば、自動化によるメリットが得られるものもある。そこで、レビューではなく判別モデルの説明変数に加えることができる3つの質問項目をレビューにおける質問項目から外し、3つの質問項目をSVMのモデル構築に用いて試行した。αの値を25%,50%,75%,100%と変化させて3つの質問項目をレビューにおいて実施した場合と比較したところ、AUCが同等か若干小さくなった。判別モデルをLOCとした場合においても同様に試行したところ、LOCを判別モデルとした場合よりもAUCが大きくなった。

本論文による提案手法が有効に使用できた場合、受入れ検査における不具合の早期発見によって以下のメリットが生じる。検出された不具合が複数ある場合にその回帰テストを1回にまとめることができる。あるモジュールで検出した不具合箇所の修正作業と、残りの部分のテストを並行して実施できるため、手戻りを回避しながらテスト期間を短縮することができる。特に、受入れ検査終盤において、納期に間に合わせることを優先するための場当たりの修正を防げるため、出荷後の保守や次バージョン以降の拡張において、変更コストを小さくできる点においてもメリットがある。

## 5. まとめ

本論文では、従来の判別モデルによる fault-prone モジュール判別にコードレビューを加えることで fault-prone モジュールのランク付け精度が向上するかを、ケーススタディによって確認した。また、ランク付け精度を維持したままレビュー工数を削減する方法についてケーススタディによって確認した。

まず、従来の判別モデルの得点とレビューによる評価得点の和を fault-prone 判別得点とし、その判別得点によりモ

ジュールをランク付けした。レビューによる評価得点に適切な荷重βを乗じて加えることで判別モデルの得点のみによるランク付けよりも精度の高いランク付けができることをケーススタディを通じて確かめた。

判別モデル構築の必要がないモデル（モジュールのコード行数に(-1)を乗じた値を与えるモデル）を評価対象に加えて、次の方法でランク付け精度を維持しつつレビュー工数を削減できるか確かめた。方法I（対象モジュールの削減）としてレビュー対象を全モジュールのα%として評価した。レビューするモジュールの選び方は判別得点の昇順、降順といった4種類の方法を比較した。適切なα、βを与えることにより、判別モデル単体よりも高精度な fault-prone モジュールのランク付けができることを確認した。サポートベクタマシンでの精度は、殆どの場合で大きくなり、αが40%のとき最も大きくなった。

方法II（レビュー観点の省略）として、規模の小さいモジュールにおいてレビューによる評価の一部を省略する方法をケーススタディにより試行した。ケーススタディの結果、方法Iと同程度のランク付け精度を保ちながら、方法Iよりも小さいレビュー工数で方法IIを実施できることが判った。αが100%の場合、方法IIが方法Iよりも46%の工数を削減できた。

### 【参考文献】

- [Basili1984] V. R. Basili and B. T. Perricone, Software errors and complexity: An empirical investigation. Communications of the ACM, Vol.27, No.1, pp.42-52 (1984)
- [Boser1992] B. E. Boser, I. M. Guyon, and V. N. Vapnik, A training algorithm for optimal margin classifiers, In Proceedings of the 5th annual ACM workshop on computational learning theory, pp.144-152 (1992)
- [Burges1999] C. J. C. Burges and D. J. Crisp. Uniqueness of the svm solution. In NIPS, pp.223-229 (1999)
- [Fowler1999] M. Fowler. Refactoring: Improving the design of existing code. Addison-Wesley (1999)
- [IPA2005] 独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター編. 組込みソフトウェア開発における品質向上の勧め (コーディング編). 翔泳社 (2005)
- [Kasai2012] 笠井則亮, 森崎修司, 松本健一. 目視評価と判別モデルを組み合わせた fault-prone モジュールのランク付け手法. 情報処理学会論文誌, Vol.53, No.9, pp.2279-2290 (2012)
- [Kasai2013] N. Kasai, S. Morisaki, K. Matsumoto, Fault-prone module prediction using a prediction model and manual inspection, In Proceedings of the 20th Asia-pacific software engineering conference, pp.106-115 (2013)
- [Ohlsson1996] N. Ohlsson and H. Alberg. Predicting fault-prone software modules in telephone switches. IEEE Transactions on Software Engineering, Vol.22, No.12, pp.886-894 (1996)
- [Schölkopf1997] B. Schölkopf, K. Sung, C.J.C. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. IEEE Transactions on Signal Processing, Vol.45, No.11, pp.2758-2765, nov (1997)
- [Sebald2000] D. J. Sebald and J. A. Bucklew, Support vector machine techniques for nonlinear equalization. IEEE Transactions on Signal Processing, Vol.48, No.11, pp.3217-3226 (2000)
- [Selby1991] R. W. Selby and V. R. Basili. Analyzing error-prone system structure. IEEE Transactions on Software Engineering, Vol.17, No.2, pp.141-152 (1991)

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

角田 雅照<sup>†1,†2</sup>門田 暁人<sup>†1,†3</sup>松本 健一<sup>†1</sup>

本研究では、組込みソフトウェア開発の設計工程から得られるメトリクスを用いて、テスト後半（結合テスト、総合テスト）における欠陥数の予測を試みた。モデルの説明変数として、基本・詳細設計におけるレビュー工数、検出欠陥数といった品質保証に関するメトリクスに加えて、試験仕様書に関するメトリクスを用いた。さらに、各仕様書のドキュメント量に関するメトリクスも採用した。分析の結果、*BRE* 平均値がおおむね 35% 程度となり、比較的高い予測精度となった。また、基本設計終了後に予測を行っても、比較的高い予測精度が高くなることがわかった。

## Predicting Faults After Unit Testing Using Design Phase Metrics in Embedded Software Development

Masateru Tsunoda<sup>†1,†2</sup>, Akito Monden<sup>†1,†3</sup>, Kenichi Matsumoto<sup>†1</sup>

This research tried to predict the number of faults after unit testing using design phase (i.e., integration testing and system testing) metrics in embedded software development. As explanatory variables of the model, we used review effort in basic and detail design phase, metrics related to quality assurance such as number of found faults. In addition, we used metrics related to test specification documents and metrics about amount of specification documents. Experimental results shows that average *BRE* (balanced relative error) of the models were almost 35%, and we can say that prediction accuracy of them were relatively high. Also, the results show that when the number of faults is predicted after basic design phase, the prediction accuracy is relative high.

### 1. はじめに

近年、組込みソフトウェアの開発は非常に増加しており、その重要性が増している。組込みソフトウェアの開発費は 2004 年度には 2 兆円であったが、2008 年度には 3 兆 5 千億円に増加した。さらに、同年には組込みソフトウェアがインストールされた機器の生産額は製造業の生産額の 50% 以上、GDP では 13% 以上の比率となっている [Tamaru 2008]。

本研究では、近年大きな課題となっている組込みソフト

ウェアの品質確保 [Mihara2013] に着目し、開発の設計工程から得られるメトリクスを用いて、テストの後半（結合テ

#### 【脚注】

- † 1 奈良先端科学技術大学院大学情報科学研究科  
Graduate School of Information Science, Nara Institute of Science and Technology
- † 2 近畿大学理工学部情報学科  
Department of Informatics, Kindai University
- † 3 岡山大学大学院自然科学研究科  
Graduate School of Natural Science and Technology, Okayama University

表 1 分析対象のメトリクス一覧

	メトリック	プロセス	試験	詳細	詳細	関連 RQ
v <sub>1</sub>	基本設計仕様書分量	N	N	N	基本設計時の仕様書の分量を示す.	1, 2, 3
v <sub>2</sub>	基本設計仕様書レビュー工数	Y	N	N	基本設計時の仕様書をレビューした際の工数を示す.	1, 2
v <sub>3</sub>	基本設計仕様書レビュー欠陥数	Y	N	N	基本設計時の仕様書をレビューした際に発見された欠陥数を示す.	1, 2
v <sub>4</sub>	基本設計試験仕様書分量	N	Y	N	基本設計に関してシステムテストにて試験するための仕様書の分量を示す.	1, 2, 3, 4
v <sub>5</sub>	基本設計試験仕様書レビュー工数	Y	Y	N	基本設計のテスト仕様書をレビューした際の工数を示す.	1, 2, 4
v <sub>6</sub>	詳細設計仕様書分量	N	N	Y	詳細設計時の仕様書の分量を示す.	1, 3
v <sub>7</sub>	詳細設計仕様書レビュー工数	Y	N	Y	詳細設計時の仕様書をレビューした際の工数を示す.	1
v <sub>8</sub>	詳細設計仕様書レビュー欠陥数	Y	N	Y	詳細設計時の仕様書をレビューした際に発見された欠陥数を示す.	1
v <sub>9</sub>	詳細設計試験仕様書分量	N	Y	Y	詳細設計に関して結合テストにて試験するための仕様書の分量を示す.	1, 3, 4
v <sub>10</sub>	詳細設計試験仕様書レビュー工数	Y	Y	Y	詳細設計のテスト仕様書をレビューした際の工数を示す.	1, 4
v <sub>11</sub>	欠陥数	-	-	-	結合テストとシステムテスト時に発見された欠陥数を示す.	-

スト、総合テスト)における欠陥数の予測を試みた結果について報告する。開発初期にテストの後半に発見される欠陥を予測することで、出荷後品質を確保するためのテスト計画立案や、手戻り工数の確保などに役立つと期待される。また、欠陥数の増大に寄与するメトリクスを明らかにすることで、欠陥を減らすためのプロセス改善にも役立つと期待される。

本研究では、ある組込みソフトウェア開発企業で収集されたデータに基づき、テスト後半の欠陥数がどの程度定量的モデルに基づいて予測可能であるかを分析する。また、テスト後半の欠陥数を予測するために優先的に収集すべきメトリクスが存在するのかどうかを明らかにする。モデルの説明変数に用いるメトリクスの候補として、基本・詳細設計におけるレビュー工数、検出欠陥数といった品質保証に関するメトリクスに加えて、試験仕様書に関するメトリクスを用いる。さらには、各仕様書のドキュメント量に関するメトリクスも採用し、ドキュメントの分量が欠陥予測に有効であるかを分析する。

これらのメトリクスは多くの企業において計測されており、テスト時に発見される欠陥数の予測に取り組んでいる研究も存在する [Takata1994][Tsunoda2009][Komuro2011]。ただし、我々の知る限り、プロジェクトの早期にテスト後半の欠陥数を予測することを目的として、設計ドキュメントに関するメトリクスのみを用いて予測モデルの構築を試みた研究は存在しない。

以降、2章では分析対象のデータについて説明し、3章では分析方法について説明する。4章では分析結果について述べ、5章では関連研究について説明する。最後に6章でまとめを述べる。

## 2. 分析に用いたデータ

分析に用いたデータは、ある組込みソフトウェア開発企業の2つの部署において収集されたデータである。2つの部署で開発される組込みソフトウェアが対象とするものは、有線・無線通信、画像処理、公共交通関連などであり、開発プロセスはウォーターフォールとなっている。2つの部署はそれぞれ独立しており、別々の顧客に対してソフトウェアを開発している。大部分のプロジェクトでは、顧客から与えられた要求に基づき、請負開発を行っており、基本設計、詳細設計、コーディング、単体テスト、結合テスト、システムテストを行っている。なお、システムテストではハードウェアのテストは含んでいない。開発規模については部署Aでは新規開発部分の中央値が約10,000ステップ、流用部分が約21,000ステップであり、部署Bではともに約20,000ステップであった。

データは2009年から2012年に収集されたものであり、部署Aには53件、部署Bには54件のデータが含まれている。データセットに含まれるメトリクスを表1に示す。メトリクスは、基本設計時のドキュメントに関するものと、詳細設計時のドキュメントに関するものが含まれる。前者は列「詳細」がN、後者はYとなっているメトリクスが該当する。ドキュメントには、設計書と試験仕様書の両方が存在する。前者に関するメトリクスは列「試験」がN、後者はYとなっている。さらに、それぞれに対してレビューが実施されており、それらに関するプロセスメトリクス、すなわちレビュー工数とレビュー時の欠陥数が記録されている。これらのメトリクスでは列「プロセス」をYとし、それ以外をNとしている。なお、試験仕様書のレビュー時の欠陥数は記録されていなかった。

予測対象であるテスト後半の欠陥数は、結合テストでの欠陥数とシステムテストでの欠陥数を合算したものである。データセットにはソースコードの行数、テストケース数、コードレビュー工数なども記録されていたが、これらのメトリクスは分析に使用しなかった。本研究ではできるだけ早期に、すなわち遅くとも詳細設計後にテスト後半の欠陥を予測することを前提としており、それ以降に収集されるメトリクスは予測モデルの説明変数として利用しないため、これらを分析から除外している。

分析では、表1のメトリクスに欠損を含むケースを除外して分析した。これは一般的にリストワイズ除去法 [Little 2002] と呼ばれる方法であり、欠損、すなわち変数に値が記録されていないケースを分析から除外する。欠損値処理の方法には、リストワイズ除去法に加え、欠損値に各変数の平均値を挿入する平均値挿入法 [Little2002] などがあるが、Strikeら [Strike2001] は、ソフトウェア開発データを用いて予測モデルを構築する場合、リストワイズ除去法による欠損値処理が妥当であることを示している。そこで本研究ではStrikeらの実験結果に基づき、リストワイズ除去法を適用した。その結果、部署Aのデータ件数は24件、部署Bのデータ件数は19件となった。

### 3. 分析方法

#### 3.1. リサーチクエスチョン

本研究の目的は、組込みソフトウェアの品質管理を支援するために、設計ドキュメントに関するメトリクスを用いてテスト後半の欠陥数の予測を行うことである。そのために、以下の4つのリサーチクエスチョンを設定した。

- RQ1: 設計ドキュメントに基づくメトリクスのみを用いて、テスト後半の欠陥数を予測できるのか？
- RQ2: 基本設計時に収集されるメトリクスのみで欠陥数を予測できるのか？
- RQ3: プロダクトメトリクスのみを用いて欠陥数を予測できるのか？
- RQ4: 試験仕様書に関するメトリクスを用いることにより、欠陥数の予測精度が高まるのか？

表1の列「関連RQ」に、それぞれのリサーチクエスチョンの分析において各メトリクスを用いるかどうかを示している。我々の知る限り、組込みソフトウェアにおいて、ドキュメントに関するメトリクスを用いてテスト後半の欠陥数を予測できるのかどうかは明らかでない。そこでRQ1を設定した。基本設計時に収集されるメトリクスのみを用いて予測することができれば、より早期にテスト計画を立案することができる。そこでRQ2を設定した。一般に、プロセスメトリクスはプロダクトメトリクスよりも収集するほうがコストが掛かるため、収集が容易ではない。実務者がプロ

セスメトリクスを収集するかどうか判断材料の一つとできるように、RQ3を設定した。設計書に対応した試験仕様書は必ずしも作成されるわけではなく、また、設計時に作成されることも限らない。もしこれらを作成していた場合、これらのメトリクスを活用したほうが良いのかどうかを実務者が判断できるようにRQ4を設定した。

実験では、各リサーチクエスチョンに答えるため、以下の3つを組み合わせることでモデルを作成し、それぞれのモデルの予測精度を比較した。

- 詳細設計時のメトリクスの有無
- プロセスメトリクスの有無
- 試験仕様書に関するメトリクスの有無

#### 3.2. 重回帰分析

欠陥数予測モデルは、重回帰分析を用いて構築した。重回帰分析はソフトウェアプロジェクトにおけるメトリクスを予測するモデルを構築する際に広く用いられる方法である。重回帰分析では、以下のようなモデルが構築される。

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \varepsilon \quad (1)$$

ここで $y$ は予測対象である目的変数を表し、本研究では欠陥数に該当する。 $x_1, x_2, \dots, x_k$ は $k$ 個の説明変数であり、本研究では設計ドキュメントに関するメトリクスとなる。 $\beta_0$ は回帰定数、 $\beta_1, \beta_2, \dots, \beta_k$ は偏回帰係数、 $\varepsilon$ は誤差項である。偏回帰係数同士を比較可能なように標準化したものを標準化偏回帰係数と呼ぶ。偏回帰係数の絶対値が大きいほど、目的変数に対する該当の説明変数の影響が大きいことを示す。

**対数変換:** 重回帰分析を用いてソフトウェアプロジェクトのメトリクスを予測する場合、モデルの構築前に説明変数、目的変数に対し、対数変換が適用される場合がある。ソフトウェアプロジェクトのデータセットには多数の小規模プロジェクトと少数の大規模プロジェクトが存在するケースが多く、そのためメトリクスの分布も偏っている場合が多い。このような場合、対数変換を適用することにより、モデルの予測精度が高まることが多い。そのため、本研究でも対数変換を適用する。

**多重共線性:** 構築されたモデルに多重共線性が発生していないかどうかを確認するために、VIF(Variance inflation factor)と条件指標を用いる。多重共線性とは、説明変数間に強い線形関係が存在していることであり、多重共線性が発生していると、偏回帰係数の分散が大きくなり、偏回帰係数の推定が不安定になる [Onodera2004]。一般に、各変数のVIFが10を超える場合やモデルの条件指標が30を超える場合に、多重共線性が発生しているとされる [Tabachnick1996] [Tanaka1995]。

**外れ値:** 予測モデルを構築する際、外れ値を除外するこ

表2 構築された予測モデル一覧 (部署 A)

プロセス	試験	詳細	Adj. R <sup>2</sup>	p 値	条件指標	Cook の距離	v <sub>1</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>4</sub>	v <sub>5</sub>	v <sub>6</sub>	v <sub>7</sub>	v <sub>8</sub>	v <sub>9</sub>	v <sub>10</sub>
Y	Y	N	0.62	0.00	16.0	0.66	3			1	2	-	-	-	-	-
Y	Y	Y	0.62	0.00	16.0	0.66	3			1	2					
N	Y	N	0.56	0.00	12.4	0.47	2	-	-	1	-	-	-	-	-	-
N	Y	Y	0.56	0.00	12.4	0.47	2	-	-	1	-					
N	N	N	0.42	0.00	8.0	0.49	1	-	-	-	-	-	-	-	-	-
N	N	Y	0.42	0.00	8.0	0.49	1	-	-	-	-					
Y	N	N	0.42	0.00	8.0	0.49	1			-	-	-	-	-	-	-
Y	N	Y	0.42	0.00	8.0	0.49	1			-	-					

表3 構築された予測モデル一覧 (部署 B)

プロセス	試験	詳細	Adj. R <sup>2</sup>	p 値	条件指標	Cook の距離	v <sub>1</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>4</sub>	v <sub>5</sub>	v <sub>6</sub>	v <sub>7</sub>	v <sub>8</sub>	v <sub>9</sub>	v <sub>10</sub>
Y	Y	Y	0.75	0.00	15.4	1.16						1			3	2
N	Y	Y	0.68	0.00	11.7	1.84		-	-	2	-	1	-	-		-
N	N	Y	0.65	0.00	9.1	1.66		-	-	-	-	1	-	-	-	-
Y	N	Y	0.65	0.00	9.1	1.66				-	-	1			-	-
N	Y	N	0.53	0.00	13.1	0.45	1	-	-	2	-	-	-	-	-	-
Y	Y	N	0.53	0.00	13.1	0.45	1			2		-	-	-	-	-
N	N	N	0.50	0.00	10.4	0.31	1	-	-	-	-	-	-	-	-	-
Y	N	N	0.50	0.00	10.4	0.31	1			-	-	-	-	-	-	-

とが行われる。外れ値とはマトリクスに何らかの特異な値を含むケースである。特異な値とは、他のプロジェクトと比較して大きく異なる値のことであり、発生する理由は単なる記録ミスの場合もあれば、仕様変更などの特殊な事情により起こるなど様々である。重回帰分析において外れ値を特定して除外する方法として、Cook の距離がある。Cook の距離とは、あるケースをモデルの推定の計算から除外した場合に、すべてのケースの残差がどの程度変化するかを示す距離である。Cook の距離が大きい時は、そのケースを除外して回帰統計量を計算した場合に係数が大きく変化したことを示す [Onodera2004]。一般に、Cook の距離が 1 より大きいケースはモデルの分析結果に大きな影響を与えているとみなされ、モデル構築から除外される。

### 3.3. 変数選択

重回帰分析により予測モデルを構築する際、変数選択を行った。変数選択は説明変数の候補から有用な変数を選択してモデルを構築する方法である。本研究では、変数増減法により変数選択を行った。具体的には以下の手順により行う [Tanaka1995]。

1. どの変数も入っていない状態から開始する。
2. もし、すべての変数が含まれていれば、取り込むべき変数はないという情報を持ってステップ3に進む。すべての変数が含まれていなければ、残りの変数を順番に1つずつ採用して見て、偏回帰係数の検定のためのF値を計算し、その値が最大となる変数を選ぶ。そのF値に対応する確率が指定された $p_{in}$ より小さく、かつその変数を採用することによって多重共線性が生じない

(VIFが10を超えない)ならば、その変数を取りこんで次のステップに進む。 $p_{in}$ より大きければ、取りこむべき変数はないという情報を持ってステップ3に進む。

3. モデルに含まれている変数について、偏回帰係数の検定のためのF値を計算し、F値が最小となる変数を選ぶ。そのF値に対応する確率が指定された $p_{out}$ より小さい時、取りこむべき変数がないという情報があれば終了する。そうでなければ、どの変数も落さずステップ4に進む。F値に対応する確率が $p_{out}$ より大きい時、その変数を落とし(取りこむべき変数がないという情報があれば、それをキャンセルして)、再びステップ3に戻る。
4. すべての変数が取りこまれていれば終了する。そうでなければステップ2に戻る。

$p_{in}$ ,  $p_{out}$ は0.05から0.5の範囲で、重要な変数を落さないことに重点をおくなら大きい値、余分な変数を取りこまないことに重点をおくなら小さい値を指定する [Tanaka1995]。本研究では重要な変数を落さないことに重点をおき、 $p_{in}$ に0.2、 $p_{out}$ に0.4を指定する。

### 3.4. 予測モデルの評価

**自由度調整済み決定係数**：構築されたモデルがどの程度適切に構築されたのかを確かめるために、本研究では自由度調整済み決定係数を用いる。決定係数はデータに対するモデルの当てはまりの良さを評価する指標であり、1に近いほどモデルのデータに対する適合度がよいことを示す [Misono2007]。ただし、決定係数は説明変数の数が増加するに従って高くなる傾向がある [Misono2007]。この問題を

表 4 各予測モデルの精度評価 (部署 A)

プロセス	試験	詳細	AE 平均値	AE 中央値	BRE 平均値	BRE 中央値	R1	R2	R3	R4	平均順位
N	N	N	9.4	6.1	34.7%	28.2%	1	3	1	1	1.5
Y	N	N	9.4	6.1	34.7%	28.2%	1	3	1	1	1.5
N	N	Y	9.4	6.1	34.7%	28.2%	1	3	1	1	1.5
Y	N	Y	9.4	6.1	34.7%	28.2%	1	3	1	1	1.5
N	Y	N	11.9	4.9	37.6%	35.9%	5	1	7	7	5.0
N	Y	Y	11.9	4.9	37.6%	35.9%	5	1	7	7	5.0
Y	Y	N	12.5	6.7	36.8%	33.3%	7	7	5	5	6.0
Y	Y	Y	12.5	6.7	36.8%	33.3%	7	7	5	5	6.0

表 5 各予測モデルの精度評価 (部署 B)

プロセス	試験	詳細	AE 平均値	AE 中央値	BRE 平均値	BRE 中央値	R1	R2	R3	R4	平均順位
N	Y	Y	42.2	35.6	35.9%	30.0%	1	4	1	1	1.8
N	Y	N	46.3	34.4	36.1%	32.5%	3	2	2	2	2.3
Y	Y	N	46.3	34.4	36.1%	32.5%	3	2	2	2	2.3
Y	Y	Y	43.2	23.6	36.3%	41.0%	2	1	6	8	4.3
N	N	Y	46.4	36.7	36.8%	32.7%	5	5	7	4	5.3
Y	N	Y	46.4	36.7	36.8%	32.7%	5	5	7	4	5.3
N	N	N	49.6	39.3	36.3%	36.9%	7	7	4	6	6.0
Y	N	N	49.6	39.3	36.3%	36.9%	7	7	4	6	6.0

回避するために、データの数と説明変数の数によって決定係数を補正した、自由度調整済み決定係数を用いる。

**リーブワンアウト法:** 構築されたモデルの予測精度を評価するために、リーブワンアウト法 [Kocaguneli2013] を適用する。リーブワンアウト法とは、データセットからあるケースを 1 件取り出して予測対象のテストデータとし、残りのケースをモデル構築に用いるラーニングデータとする。これを、すべてのケースに対して繰り返す方法である。テストデータでは欠陥数を未知とみなし、予測後に実際の欠陥数と比較して予測がどの程度正確であるかを評価する。

**予測精度評価指標:** 構築されたモデルによる欠陥数の予測精度を評価するため、AE(Absolute Error), MRE(Magnitude of Relative Error)[Conte1986], BRE (Balanced Relative Error) [Miyazaki1994] の 3 つの指標の平均値と中央値を用いた。欠陥数の実測値を  $x$ , 予測値を  $\hat{x}$  とするとき、それぞれの指標は以下の式により求められる。

$$AE = |x - \hat{x}| \quad (2)$$

$$MRE = \frac{|x - \hat{x}|}{x} \quad (3)$$

$$BRE = \begin{cases} \frac{(\hat{x} - x)}{x}, & \hat{x} - x \geq 0 \\ \frac{(x - \hat{x})}{\hat{x}}, & \hat{x} - x < 0 \end{cases} \quad (4)$$

それぞれの指標の値が小さいほど、予測値と実測値との差が小さい、すなわち予測精度が高いことを示す。直感的には MRE と BRE は実測値との相対誤差であるといえる。ただし、MRE は過大予測に対し、アンバランスな評価にな

るという問題がある。過少予測の場合、MRE は最大でも 1 にしかならない。例えば実測値が 10, 予測値が 0 の場合、MRE は 1 となる。そこで本研究では MRE に加え、過大予測と過少予測をバランスよく評価する指標 [Mølokken-Østfold2005] である BRE を予測精度の評価指標に用いる。

## 4. 分析結果

部署 A と部署 B で目的変数である欠陥数の値域が大きく異なっていたため、データの層別を行い、それぞれの部署向けに予測モデルを構築した。データを層別して個別のモデルを構築することは、精度の高い予測モデルを構築するために一般に行われる方法の一つである。

構築したモデルの概要を 4.1 節で述べ、4.2 節において BRE などの予測精度を中心にモデルを評価するとともに、リサーチクエスチョンに対する答えを述べる。

### 4.1. モデルの概要

部署 A 向けに構築した予測モデルの概要を表 2 に、部署 B 向けに構築した予測モデルの概要を表 3 に示す。プロセス、試験、詳細の各列は、それぞれプロセスメトリクス、試験仕様書に関するメトリクス、詳細設計時のメトリクスを説明変数の候補として含めたかを示す。含めた場合には Y とし、含めない場合は N としている。その結果、各変数が採用されたかどうかを  $v_1$  から  $v_{10}$  の列に示す。 $v_1$  から  $v_{10}$  に対応する変数は表 1 に示している。列中の数値は、構築されたモデルにおいて、標準偏回帰係数の絶対値の大きさの順位を示している。数値が小さいほど偏回帰係数が大きい、すなわち目的変数の値に対する影響が大きかったことを示

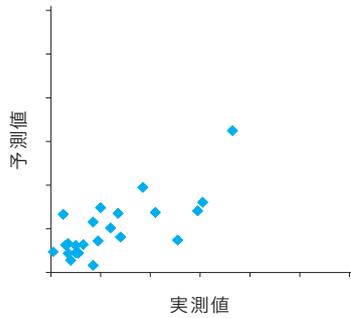


図1 最も精度の高いモデルによる予測 (部署 A)

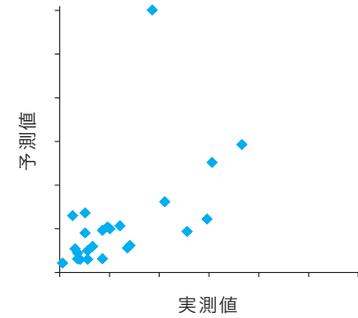


図2 最も精度の高いモデルによる予測 (部署 B)

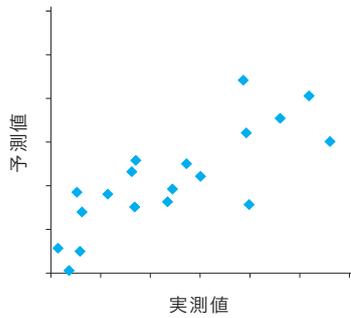


図3 最も精度の低いモデルによる予測 (部署 A)

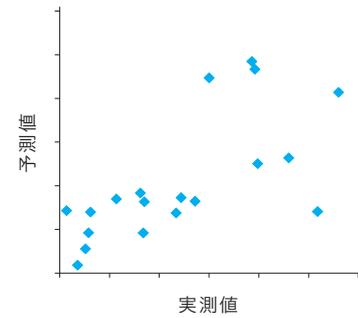


図4 最も精度の低いモデルによる予測 (部署 B)

す。空白は採用されなかったことを示し、「-」は説明変数の候補に含まれていなかったことを示す。Adj.  $R^2$  は自由度調整済み決定係数を示し、この値が大きい順に各行、すなわちモデルを並べ替えている。

モデルが適切に構築されたことを示すために、p 値、条件指標、Cook の距離を示している。p 値は構築されたモデルが統計的に有意であったかどうかを示し、全て有意水準 5% で有意であった。表に示すように、すべてのモデルの条件指標は 30 を下回っていた。また、結果は省略するが、各変数の VIF は 10 を下回っていたことから、多重共線性は発生していないと考えられる。Cook の距離は、各モデルにおける Cook の距離の最大値を示す。部署 A では全てのモデルが、部署 B では半数のモデルにおいて 1 を下回っていたことから、除外すべき（明らかな）外れ値プロジェクトは含まれていないと考えられる。

自由度調整済み決定係数によってモデルを並べ替えた場合、部署 A では説明変数に関して以下の結果が観察された。それぞれの項目は RQ2 から RQ4 に関連しているが、各リサーチクエスチョンに対する答えは次節の分析結果に基づいて述べる。

- 全てのモデルにおいて、詳細設計時のメトリクスは説明変数として採用されなかった。
- 自由度調整済みの上位 2 つのモデルにおいて、プロセスメトリクス  $v_3$  が採用されたが、その他のモデルでは採用されなかった。
- 上位 4 つのモデルにおいて、試験仕様書に関するメトリクス  $v_4$  が採用され、かつ偏回帰係数が最も大きくなっ

ていた。

同様に、部署 B では以下のような結果となった。

- 上位 4 つのモデルにおいて、詳細設計時のメトリクス  $v_6$  が採用され、かつ偏回帰係数が最も大きくなっていった。
- 最上位のモデルにおいて、プロセスメトリクス  $v_{10}$  が採用されたが、その他のモデルでは採用されなかった。
- 最上位のモデルにおいて、試験仕様書に関するメトリクス  $v_9$ 、 $v_{10}$  が採用された。上位から 2 番目のモデルにおいて同様のメトリクス  $v_4$  が採用された。

## 4.2. モデルの予測精度

部署 A 向けに構築したモデルの予測精度を表 4 に、部署 B 向けに構築したモデルの予測精度を表 5 に示す。プロセス、試験、詳細の各列の意味は表 2、表 3 と同様である。列 R1 から R4 は、AE 平均値から BRE 中央値の 4 つの指標それぞれに関して、全モデルの中で何番目に精度が高かったのかを順位付けしたものである。平均順位は、その 4 つの順位を平均したものであり、それぞれの表は平均順位の昇順、すなわち予測精度が高かった順にモデルを並び替えている。

**RQ1 に関する分析：**最も精度の高かったモデルにおける予測精度に着目する。部署 A、部署 B において、最も精度の高いモデルでは BRE 平均値が 35% 程度、BRE 中央値が 30% 程度となっており、誤差が小さかった。部署 B の場合、AE の平均値と中央値が部署 A よりも大きくなっているが、これは欠陥数の値域が異なるためであり、相対的な誤差 (BRE) を考慮すると、予測精度が低いわけではない。図 1、図 2 は、予測精度を視覚化するために、横軸を実測値、縦軸を予測値として散布図としたものである。図には部署

A, Bにおいて最も精度が高かった（平均順位が最も小さい）モデルを示している（守秘義務上、各軸に値を示していないが、各グラフの縦軸と横軸の値域は同一としている）。図からも良好な予測結果といえる。これらの結果から、RQ1に対する答えはYesとなる。なお、部署A, Bにおいて最も精度が低かった場合でも、比較的予測精度は良好であった（図3, 図4）。

**RQ2に関する分析：**部署Aでは、最も精度の高いモデルでは基本設計時に収集されるメトリクスのみが採用されていた。部署Bの場合、予測精度の上位2番目と3番目のモデルにおいて、基本設計時のメトリクスのみが採用されており、1番目のモデルと比べて予測精度に大きな差はなかった。外れ値の影響を避けるため、Wilcoxonの符号付き順位検定を用い、予測精度に差があるかを検定したが、有意水準5%で有意差がなかった。よって、RQ2に対する答えはYesとなる。

**RQ3に関する分析：**部署A, 部署Bとも、最も精度の高いモデルではプロダクトメトリクスのみが採用されていた。このことから、RQ3に対する答えはYesとなる。ただし表2, 表3において、プロセスメトリクスを含んでいるモデルのほうが説明力（自由度調整済み決定係数）が大きかったことから、これらのメトリクスを収集している企業の場合、説明変数として用いることを検討してもよいと考えられる。

**RQ4に関する分析：**部署Aでは、最も精度の高いモデルでは試験仕様書に関するメトリクスが採用されていなかった。部署Bの場合、予測精度の上位4つのモデルでは、全て試験仕様書に関するメトリクスを用いていた。ただし、予測精度に大きな違いはなく、Wilcoxonの符号付き順位検定においても、有意水準5%で有意差がなかった。このことから、RQ4に対する答えはNoとなる。なお、表2, 表3において、説明力が上位のモデルでは試験仕様書に関するメトリクスを含んでおり、また、有意差がなかったとはいえ、予測精度がわずかながら高かったことから、これらのメトリクスが既に収集されている場合、活用を検討してもよいと考えられる。

### 4.3. 考察

目的変数に単体テストの欠陥数を含めて予測した場合のモデルの精度を表6に示す。表には、説明変数の候補として全ての変数を用いた場合と、基本設計仕様書分量のみを用いた場合の精度を部署別に示している。単体テストの欠陥数を含めない場合（表4, 表5）と比較して、部署Aでは予測精度が大きく低下し、部署Bでは予測精度の若干の低下が見られた。Wilcoxonの符号付き順位検定を用い、単体テストの欠陥数を含めない場合と予測精度に差があるかを検定すると、部署Aの場合、AE, BREとも有意に悪化していた。なお、検定では各表で最も精度の高いモデル同士を比較している。

表6 単体テスト欠陥数を含めた場合の予測精度

部署	説明変数候補	AE 平均値	AE 中央値	BRE 平均値	BRE 中央値
A	$v_1$	31.5	16.2	159.4%	74.4%
A	$v_1 - v_9$	46.4	13.6	121.8%	56.5%
B	$v_1$	61.9	33.8	43.2%	26.0%
B	$v_1 - v_9$	67.1	36.6	40.7%	36.5%

表7 単体テストの欠陥数と結合テスト以降での欠陥数との相関係数

部署	相関係数	p値
A	0.38	0.07
B	0.71	0.00

単体テストの欠陥数と、結合テスト以降での欠陥数について、部署ごとにスピアマンの順位相関係数を求めた結果を表7に示す。部署Bでは相関が高かったのに対し、部署Aでは相関が低く、かつ有意水準5%で有意となっていなかった。部署AとBで単体テストの欠陥数を含めた場合と含めない場合で予測精度に違いが生じた原因は、単体テストの欠陥数と、結合テスト以降での欠陥数の関連の強さの違いが原因であると考えられる。

この分析結果より、設計関連メトリクスを用いてプロジェクトの早期に欠陥数を予測する場合、単体テストの欠陥数を含めると予測精度が低下することがあるといえる。

## 5. 関連研究

これまで、プロセスに関するメトリクスを用いてテスト時の欠陥数の予測を試みた研究はいくつか存在する。小室ら[Komuro2011]は、レビュー実績などを用いてテスト工程で発見される欠陥数を予測するモデルを提案している。ただし、説明変数に設計ドキュメントに関するプロダクトメトリクスを用いておらず、また、テスト工程全体の欠陥数を予測対象としている点が異なる。文献[Tsunoda2009]では、システムテスト欠陥密度を予測するために、コードレビュー指摘密度などを説明変数に用いている。ただし、ソフトウェアのコード行数などを説明変数として用いているため、示されているモデルを設計完了時に適用することはできない。

設計ドキュメントに関するプロダクトメトリクスとプロセスメトリクスの両方を用いて品質の予測を試みた研究もわずかながら存在する[Katayama2009][Takata1994]。ただし、試験仕様書に関するメトリクスについては、我々の知る限りこれまで用いられておらず、従って予測精度に寄与するのかわかりかたではなかった。また、それらの研究では目的変数に単体テストの欠陥数を合算した場合と合算しない場合の予測精度についても比較していない。

片山ら[Katayama2009]は、組込みソフトウェア開発に

において、低品質モジュールを予測するために、基本・詳細設計での、設計ドキュメントに関するプロセス・プロダクトメトリクスを説明変数としてモデルを構築している。ただし、片山らはテストの順番を決めるために、単体テストからシステムテストまでに発見される欠陥の多いモジュールを予測に基づきランキングしており、予測モデルの使用目的が本論文と異なる。このため、欠陥数の予測精度が明らかでない。また、各メトリクスの予測に対する効果も評価していない。

高田ら [Takata1994] は単体テストからシステムテストにおいて発見される欠陥数を予測するために、片山らと同様のメトリクスを説明変数として用いている。ただし、コードレビューでの欠陥数も説明変数に含めているため、モデルを設計完了時に適用することはできない。

当然ではあるが、従来研究では本研究のリサーチクエスチョンに答えることはできない。テスト後半の欠陥数に着目し、各種のメトリクスを組み合わせることで欠陥数の予測精度を評価し、それぞれのメトリクスの予測精度に対する寄与を確かめたことが本研究の主要な貢献である。

## 6. おわりに

本研究では、組込みソフトウェアの開発における定量的な品質管理を支援することを目的とし、設計ドキュメントに関するメトリクスを説明変数として、テスト後半の欠陥数を予測することを行った。テスト後半の欠陥数とは、結合テストとシステムテストにおいて発見される欠陥の合計である。実験では、組込みソフトウェアを開発しているある企業において収集されたデータを用いた。予測モデルの説明変数として用いたものは、仕様書の分量などのプロダクトメトリクス、レビュー工数やレビュー欠陥数などのプロセスメトリクス、試験仕様書の分量など、試験仕様書に関するメトリクスである。これらのメトリクスは基本設計時と詳細設計時に収集される。データは2つの部署から収集されており、それぞれの部署において予測モデルを作成し、精度を評価した。その結果より、以下のことがいえる。

- どちらの部署においても、リープワンアウトにより予測した場合、BRE 平均値がおおむね 35% 程度となり、比較的高い精度となった。よって、表1のようなメトリクスを収集している企業では、欠陥数の予測を試みる価値があると考えられる。この場合、基本設計終了後に予測を試みても、ある程度精度が高いことが期待される。
- 実験結果からは、どのメトリクスを優先的に用いるべきかは一概にはいえなかった。ただし、採用された説明変数が異なっても(表2, 表3)、予測精度に大きな違いはなかった。このことから、表1のようなメトリクスを一部でも収集している場合、欠陥数の予測を試みるとよいと考えられる。

- 部署によって、どのメトリクスを説明変数の候補とするとよいのかが大きく異なっていた。このことから、最初から各部署で共通の予測モデルを構築することを目指すのではなく、部署ごとに別個に構築することも検討すべきであるといえる。
- 単体テストの欠陥数を目的変数に合算すると予測精度が低下することがあった。このため、設計関連メトリクスのみを用いて予測モデルを構築する際には、単体テストの欠陥数を目的変数に合算しないほうがよいといえる。

欠陥数の値域が異なる2つの部署どちらにおいても類似の結果が得られたことから、結果の信頼性は比較的高いと考えられる。今後の課題は、設計ドキュメントのメトリクスとソフトウェアの品質に関連があるかどうかを明らかにすることである。

## 謝辞

本研究の一部は、文部科学省科学研究補助費(基盤C: 課題番号 25330090)による助成を受けた。

### 【参考文献】

- [Conte1986] S. Conte, H. Dunsmore, and V. Shen, Software Engineering, Metrics and Models, Benjamin Cummings, 1986.
- [Katayama2009] 片山真一, 大蔵君治, 伏田享平, 川口真司, 名倉正剛, 門田暁人, 飯田元, “ソフトウェアタグを用いた設計文書メトリクスからの低品質モジュールの予測,” 電子情報通信学会技術研究報告, Vol.109, No.343, pp.67-72, 2009.
- [Kocaguneli2013] E. Kocaguneli, and T. Menzies, “Software effort models should be assessed via leave-one-out validation,” Journal of Systems and Software, Vol.86, No.7, pp. 1879-1890, 2013.
- [Komuro2011] 小室睦, 薦田憲久, “ピアレビューデータに基づく品質予測モデル,” 電子情報通信学会論文誌 D, Vol.J94-D, No.2, pp.439-449, 2011.
- [Little2002] R. Little, and D. Rubin, Statistical Analysis with Missing Data, 2nd ed., John Wiley & Sons, 2002.
- [Mihara2013] 三原幸博, “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],” <http://www.ipa.go.jp/files/000030363.pdf>
- [Misono2007] 御園謙吉, 良永康平(編), よくわかる統計学 II 経済統計編, ミネルヴァ書房, 2007.
- [Miyazaki1994] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki, “Robust Regression for Developing Software Estimation Models,” Journal of Systems and Software, Vol.27, No.1, pp.3-16, 1994.
- [Møløkken-Østfold2005] K. Møløkken-Østfold, and M. Jørgensen, “A Comparison of Software Project Overruns-Flexible versus Sequential Development Models,” IEEE Transactions on Software Engineering, Vol.31, No.9, pp.754-766, 2005.
- [Onodera2004] 小野寺孝義, 山本嘉一郎(編), SPSS 事典: BASE 編, ナカニシヤ出版, 2004.
- [Strike2001] K. Strike, K. El Eman, and N. Madhavji, “Software Cost Estimation with Incomplete Data,” IEEE Transactions on Software Engineering, Vol.27, No.10, pp.890-908, 2001.
- [Takata1994] 高田義広, 松本健一, 鳥居宏次, “ニューラルネットを用いたソフトウェア信頼性予測モデル,” 電子情報通信学会論文誌 D-I, Vol. J77-D-I, No.6, pp.454-461, 1994.
- [Tamaru2008] 田丸喜一郎, “10年で変わった「組込み」、変わらない「組込み」,” ZIP WATCHERS, pp.6-7, 2008.
- [Tabachnick1996] B. Tabachnick, and L. Fidell, Using Multivariate Statistics, 3rd Edition, Harper Collins College Publishers, 1996.
- [Tanaka1995] 田中豊, 垂水共之(編), Windows 版 統計解析ハンドブック 多変量解析, 共立出版, 1995.
- [Tsunoda2009] 角田雅照, 玉田春昭, 森崎修司, 松村知子, 黒崎章, 松本健一, “コードレビュー指摘密度を用いたソフトウェア欠陥密度予測,” 情報処理学会論文誌, Vol.50, No.3, pp.1144-1155, 2009.

# 国際展示会「CeBIT2015」への参加 及び海外有力機関との意見交換

SEC システムグループ 主任

八嶋 俊介

SEC 調査役

十山 圭介

## 1. はじめに

ハノーバー（ドイツ）で2015年3月16日～3月20日に開催された国際展示会「CeBIT<sup>\*1</sup>2015」に参加し、第四次産業革命（Industrie4.0）と呼ばれるドイツの政策が、製造現場にどのような影響を与えているかについて、IoT時代を見据えた今後のIPA/SECの活動に活かすべく調査を行ったので、その模様を報告する。

また、国際連携活動の一環として、2014年に引き続き、オランダTNO-ESI<sup>\*2</sup>及びイギリスMISRA<sup>\*3</sup>を訪問したので、現地での意見交換の内容について報告する。

## 2. CeBIT2015におけるトピック

### ① Industrie4.0が目指す標準化戦略と企業間連携

15万平方メートルを超える広大な展示面積を誇る会場（参考：東京ドームの約12倍）の中でひととき目立つのが、SAP社、ドイツテレコム社をはじめとしたIndustrie4.0を推進／牽引している企業<sup>\*4</sup>のブースであった。M2Mの技術を用いたファクトリーオートメーション（写真2）や、物流における自動追跡のソリューションなどが展示される中で、日本の展示会ではあまり見られず特徴的だったのが、企業間でのコラボレーション展示である。Industrie4.0は、メルケル首相の強力な主導のもと、企業間の連携を行いながら進められているとのことであるが、各企業のブースがそれを示していた（写真3）。

ドイツテレコム社の幹部は、SAP社との連携について現地のインタビューで「我々の目的はできるだけ早く、デファクトスタンダードを確立することだ」と述べており、SAP社と連携することで、M2Mの通信技術の標準化を取ることが必要だと強調していた。

また、今年のCeBITのパートナー開催国である中国とドイツの連携も、大きなトピックのひとつである。

メルケル首相は開会式にて、技術開発や標準化で中独が協力していくことを表明した。一方の中国サイド、CeBIT Global Conferenceに登壇したファーウェイ社法人向けICTソリューション事業グループプレジデントのヤン氏は、SAP社と連携して開発したシステムを使用することで、ミキサー車の故障を予知したメンテナンスが可能になり、1カ月間における故障時間の平均（MMT: Mean Malfunction Time）が1/4に減少したとこれまでの実績を発表した（写真4）。ファーウェイ社は、これまで主に米国企業が主導するIIC（Industrial Internet Consortium）に属していたが、今回の発表でIndustrie4.0サイドにも加盟したことになる。

### ② スタートアップ企業のIoTへのアプローチ

ドイツでは多くのスタートアップ企業がIoTに取り組んでいた。日本の展示会では、企業の規模と展示ブースの大きさがおおむね比例していることが多いが、CeBITにおいては日本と異なり、新しい試みが行われていた。「CODE\_n」と呼ばれるビジネスプランコンテストで優秀な成績を収めたスタートアップ企業が、特設会場で大きな展示を行っていた。

とくに目立っていたのが、ミュンヘンのデザイナーチームによって開発された「ROBOCHOP」で、スマート

#### 【脚注】

- ※1 CeBIT (Centrum für Büroautomation, Informationstechnologie und Telekommunikation)：ドイツ語でビジネスオートメーション・情報技術・テレコミュニケーションセンターの意。ドイツのハノーバーの世界最大級の見本市会場で毎年春に開催される世界最大級のコンピュータエキスポ。IT分野における最先端の指標ともいえる。
- ※2 TNO-ESI (Netherlands Organization for Applied Scientific Research-Embedded Systems Innovation)：応用科学研究機構組込みシステムイノベーション。TNOは科学技術分野における応用科学研究を行うことを目的としてオランダ議会によって1932年に設立された欧州では最大規模を誇る中立の総合受託試験研究機関。
- ※3 MISRA (The Motor Industry Software Reliability Association)：自動車メーカー、部品メーカー、研究者からなる欧州の自動車業界団体。
- ※4 Industrie4.0プラットフォーム運営委員会メンバ



写真1 CeBIT 会場入り口より。開場時間前から熱気に包まれていた。



写真2 ドイツテレコム社の展示。各工場のラインにおける稼働率の状況が一目で分かる。



写真3 HP 社は SAP 社と連携して、水量を自動制御できるインテリジェント・メンテナンス・システムを開発した。

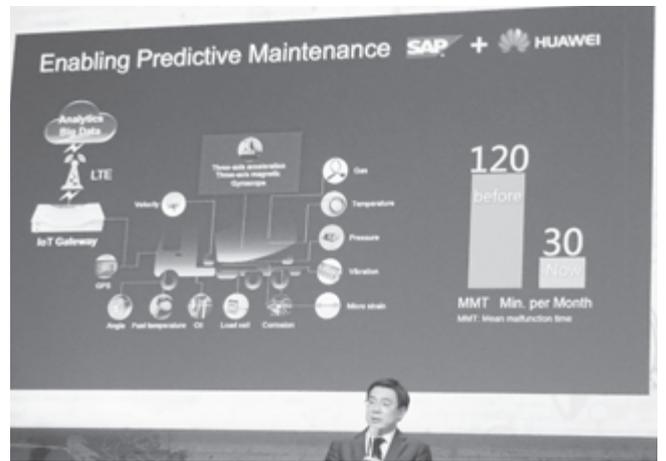


写真4 ファーウェイ社 ヤン氏の講演。中独の技術開発の連携による成功をアピールしていた。

フォンのアプリで制御された4台のクレーン型ロボットが、50cm角ほどのブロックに対してダイナミックかつ精密な加工を行っていた(写真5)。実用段階においては、スマートフォンアプリを用いて巨大なロボットを扱うには、高い信頼性や安全性が求められると考えられるが、そこに対しては触れられていなかった。

また、別のスタートアップ企業であるKIWI.KI社では、家のドアを開ける際、従来の鍵を使わず、スマートフォンアプリを鍵として用いるスマートホームシステムの展示を行っていた。利点として、従来のように鍵を探す必要がない、ひとつの「Ki(非接触式の小型チップ)」が複数のドアの鍵となる、スマートフォン経由で家族や友人のための鍵を登録できる、などが挙げられていた。

コンテストでの評価どおり、どのブースでもIoTで実現可能な画期的なサービスが展開されていた。ただし、



写真5 ROBOCHOPの展示。4台のクレーン型ロボットが、スマホアプリの制御で精密な加工を行う。

セキュリティや信頼性についてはどこも触れられておらず、実用化に向けて解決すべき課題は大きいと考えられる。

### ③ ドイツのソフトウェア及びものづくり教育への姿勢

ソフトウェアは無形物でありかつ専門的であるため、一般にその有用性や重要性が伝えづらいという側面がある。日本でもソフトウェアに興味を持ち、ものづくりの面白さを身をもって体験している若年層はいるが、展示会などでは広く効果的にアピールできていないように思われる。

CeBIT 会場の Fraunhofer<sup>\*5</sup> ブースでは、Roberta と呼ばれる子供向けのロボコン特設会場に大きなスペースが割かれており、盛況にイベントが開催されていた（写真6）。

日本でも ET（組込み総合技術展）や CEATEC JAPAN など、IT 関係の展示会は多くあるが、このように若年層の子供たちが楽しそうにソフトウェアやロボットと触れ合う機会はあまりないように思われる。ロボット技術自体はドイツより日本のほうが進んでいるかも知れないが、2020年の東京オリンピックに向けてIT人材が不足すると叫ばれている昨今、こういった取り組みや考え方に関しては、日本もドイツに学ぶところがあるのではないかと考える。



写真6 Fraunhofer ブースの Roberta 会場では、たくさんの子供たちがソフトウェアやロボットと触れ合っていた。

### 3. TNO-ESI との意見交換

組込みシステムでのソフトウェア分析と検証について、2014年3月に意見交換を行ったオランダ TNO-ESI を訪問した（写真7、8）。今回は IPA/SEC の障害原因診断 WG<sup>\*6</sup> の成果を説明して、障害原因の診断方式やモデルベースによる検査手法について昨年からの進捗などを Dr. Jan Tretmans（リサーチフェロー、ラド

バウト大 准教授）、Dr. Pierre van de Laar（リサーチフェロー）と下記の通り意見交換・議論した。

- IPA/SEC からモデルベースアプローチによる障害原因診断手法の確立について、ET2014 でのプレスリリースと IPA セミナーなどでの関連トピックを説明した。
- ESI からは Tretmans 氏より、モデルベース検査について説明を受けた。今回は、昨年からの進捗として現在注力しているツールに関して詳しい話を聞いた。
- van de Laar 氏からスペクトラムベースの障害箇所特定の手法について説明を受けた。テストケース実行時に通過した関数やモジュールに対して、合/否のカウンタでスペクトル的な配列を得、それを解析することで不具合が含まれそうな場所を特定するものである。

モデルベース検査ツールは形式記述からテストケースを自動生成するものであり、障害原因診断 WG で形式検証に利用しているモデル検査ツール SPIN や UPPAAL なども含め複数の記述法に対応しているため、検証について今後同様の環境に基づいた議論が見込める。また、スペクトラムベースの障害箇所特定手法は、障害原因診断 WG でも障害場所/原因候補の絞り込みについて検討を深めて行くので参考になると考える。今後も双方の活動について、定期的な情報交換や意見交換を行う関係を継続し、IPA のセミナーやシンポジウムなどでも IPA/SEC から要請があれば、対応を検討いただけることを確認した。

### 4. MISRA との意見交換

2014年11月に横浜で実施した打合せに続いて、今後の C++ 言語版のコーディング作法ガイド改訂における協力内容について調整を行い、MISRA の状況と計画について情報収集を行った。IPA/SEC と MISRA とはコーディング作法ガイドの改訂などについて定常的に意見交換や相互レビューを実施している。今回 MISRA ゼネラルマネージャの Dr. David Ward と、MISRA の活動状況及び、IPA/SEC の ESCR C++ 言語版の改訂や ESCR C 言語版の

#### 【脚注】

- \*5 Fraunhofer-Gesellschaft（フラウンホーファー研究機構）：ドイツ全土に67の研究所・研究ユニットを持つ欧州最大の応用研究機関。各研究所は科学の様々な応用を研究テーマとしている。
- \*6 障害原因診断 WG：製品・制御システムに障害事象が発生したときに、ソフトウェア面の原因を、モデルベースアプローチ、システムズエンジニアリングに基づき迅速かつ的確に指摘できるようにすることを目的として、IPA/SEC が設置した WG。



写真7 TNO-ESI エントランスにて。アイントホーフェン工科大のキャンパス内にあるが、昨年訪問時の場所から移転しており、広がっていた。

セキュアコーディングへの対応について、以下の通り意見交換した。

- MISRA からの活動状況紹介  
MISRA では 2014 年に C++ 言語版のコーディング作法ガイドの改訂作業のための WG を設置し、2015 年 1 月に WG 会合を開催し改訂作業を進めている。前年に策定した計画通り、旧版 (MISRA C++ : 2008) のエラー訂正から作業を始めた。
- ESCR C 言語版でのセキュアコーディングについての説明と議論  
IPA/SEC からはコーディング作法ガイド改訂 WG でのセキュアコーディングに対する取り組みと、その中で IPA/セキュリティセンターと連携して対応付けを行った ESCR のルールと CERT C ルール、CWE の対応表について説明し、コメントを要請した。MISRA では、セキュアコーディングについての活動は重要であると認識しており、具体的な活動の計画は策定されていないが、このような対応表は大事であるとの考えであった。

## 5. おわりに

CeBIT2015 の会場では、大企業、スタートアップ企業問わず、国を挙げて Industrie4.0 に取り組んでいくという熱気を肌で感じることができた。また、SAP 社と各企業との連携の展示が多く見られたように、情報産業と製造業の融合を考える際、やはり中心となるのはソフトウェア・エンジニアリングであるとあらためて感じた (Industrie4.0 の戦略提言を行っている Acatech<sup>\*7</sup> のカガーマン会長は、SAP 社の元 CEO である)。



写真8 TNO-ESI の Tretmans 准教授 (写真左) と、モデルベースアプローチによる障害原因診断手法について詳細な議論を交わした。

しかし一方で、広大な CeBIT 会場の中で、我が国の隣国である中国からは約 600 社の出展があった中、日本からの出展ブースがほとんど見られなかったのが印象的であり、心配に感じた (公式発表によると、日本からの出展は 17 社である)。

CeBIT2015 開催の直前にメルケル首相が来日した際、各所にて Industrie4.0 の重要性を説明したそうだが、各種新聞などで報道された記事では、隣国との歴史認識などの話題が多かったように記憶している。これまでのドイツと同様、スマート工場のノウハウが自社内に閉じられている日本においても、技術の標準化やルール作りなどに向けて検討していかなければ、世界での競争力を失ってしまうのではないかと感じると共に、ソフトウェア・エンジニアリングに知見のある IPA/SEC が貢献できる部分も多くあるのではないかと感じた。

TNO-ESI とは、要求仕様から論理、形式記述、モデルでのシミュレーションによって実システムの挙動の検証を行うことは同じ考え方であり、IPA/SEC のアプローチと補完的になっているとの議論ができた。今後、双方で連携できるテーマの提案に結びつくよう、関係の更なる強化を図りたい。

また、MISRA とは C++ 言語版のコーディング作法ガイド改訂の進捗に合わせた議論の場の設定とセキュアコーディングに対する連携の手掛かりを得ることができた。今後、コーディング作法の充実に向けた相互連携を更に進めていきたい。

### 【脚注】

\*7 Acatech (ドイツ工学アカデミー) : 政府・主要企業との協業により、Industrie4.0 実現のための勧告など、エンジニアリングの見地からの分析・助言を取りまとめている。

# ドイツ Steinbeis 訪問報告

IPA IT人材育成本部 イノベーション人材センター  
センター長

片岡 晃

2015年5月27日（水）から6月5日（金）にかけて、IPA IT人材育成本部 HRDイニシアティブセンター 津留専門委員と共にドイツの Steinbeis への訪問、EC主催の European conference 及び CEN<sup>※1</sup> 会議など各種会議に出席した。本レポートでは、ドイツ シュツットガルト市に本拠地のある欧州最大の技術移転（産学官連携サポート）機関である Steinbeis について報告したい。

## 1. 訪問の目的

今回の訪問の目的は主に以下の2点である。

- ① Steinbeis の活動、及び IPA の人材育成活動を相互に紹介し、とくに、未踏事業で創出された、人材、技術、アイデアなどについての産業界や研究分野での活用促進と産業界におけるアウトプットの拡大、またそのための組織体制構築に対して、アドバイスを受け、ヒントを得ること。
- ② 併設する Steinbeis 大学を訪問し、産学連携による人材育成に関して参考となる情報を収集すること。

## 2. Steinbeis の概要

Steinbeis はドイツ国内において、経済全体が科学的な知識を利用できるようにすることを目的として1971年に非営利財団として創立され、1998年には Steinbeis GmbH&Co. が設立されて完全民営化された。Steinbeis GmbH&Co. の傘下には、専門分野の異なる各種センター（日本法人も含まれる）が配置されている。現在、世界47カ国1,000カ所の拠点、6,000人以上の大学教授、研究者などのスタッフを有し、ビジネスニーズをスタッフにつなぎ、スタッフが専門知識を持って、コンサルティング、研究開発業務、技術移転、人材育成、起業支援などを有償で受託し対応している。このように、大学、研究機関の研究者が、特定企業のビジネスにおける「技術、マネジメントの実践的問題解決」を行う仕組みを持ち、グローバルに年間10,000社以上の企業に対し14,000以上のプロジェクトを実施している。各センターは、単独若しくは連携してビジネスを支援している。ドイツバーデン・ヴュルテンベルク州（州都 シュツットガルト市）の3大産業は、自動車、工作機械、ICTであるため、Steinbeis のプロジェクトの多くは、この領域で実施されている。

Steinbeis 側からは、未踏事業及び「一般社団法人未踏」

の仕組みは洗練されているとの感想があり、ベーシックな技術やアイデア、また優れた人材を社会と結びつけるポジショニングが、教育機関の研究と産業界を結びつける Steinbeis の活動と似ていることに大変興味を持っていただいた。また、Steinbeis では、ブリッジ役としてスタッフ（コンサルタント）が居る。一方、未踏では、ブリッジ役となるのは、PMでありアドバイザーである。これもよく似た構造である。

Steinbeis 側からは、「未踏事業」、及び「一般社団法人未踏」に、社会や産業界のニーズやリクワイアメントを取り入れる仕組みをより強化してはどうかとの提案があった。現在、未踏事業期間は、採択テーマの成果創出を通じた人材育成を主眼とした9カ月間の限定された取り組みであるため、未踏事業期間だけでなく、その前後のフェーズも含め、社会や産業界と接点を持つことも含めた総合的な視野での検討が必要であると考えている。具体的には、未踏期間においては、中間合宿などで、社会や産業界、また研究機関などのニーズを反映する形をより促進すると共に、「大学などにおけるアントレプレナー教育」や一般社団法人未踏の活動などへの反映を進めたい。

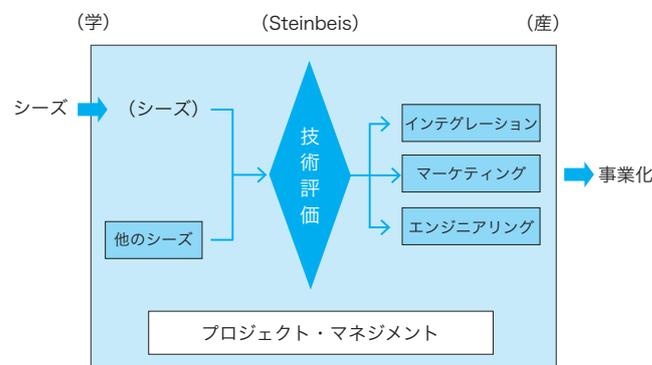


図1 Steinbeis の機能

【脚注】

※1 CEN : Comité Européen de Normalisation



写真 1 : Steinbeis 本部棟



写真 2 : Steinbeis foundation 創始者  
Ferdinand von Steinbeis の胸像と

### 3. Steinbeis 大学

Steinbeis 大学は、Steinbeis が併設する大学で 1998 年に設立され、ドイツの私立大学では最大規模である。本拠地はベルリンにある。Steinbeis 大学の設立趣旨は、国際的に著名な理工系大学の学位取得者を中心に（経済学などの学位取得者も含まれる）、一定期間以上の従業経験者に対して、企業のプロジェクトに参加させながらマネジメント（マーケティング、ビジネスプラン作成など）の理論教育と実践指導を行い、技術とマネジメントの両面に能力を有するプロジェクトリーダーを育成することにある。

Steinbeis 大学は、学生は習得した知識をすぐに自らの仕事に活用するという基本的な考え方の下、学部ではすべての学生がパートナー企業を持つことが条件となっている。学生は企業の社員である必要はないが、学費はパートナー企業が支払う仕組みである。

一方、修士課程では、大学が企業とのマッチングを行い、企業は専門技術者を 2 年間試用し、終了後に雇用するチャンスがある。これは知名度のない企業で、開発の意欲はあるが優秀で専門的な人材の雇用が簡単ではない企業にとってとくにメリットが大きい。学生はほとんどが社会人であるが、授業料、生活費は、その課題を与えた会社が負担し、学生の負担はない。学生が課題と共に学業をスタートし、在学中にその課題を解決して卒業するという、PBL<sup>※2</sup>を大きく発展させたプログラムとも言える。

日本においては、このような仕組みがほとんどなく、技術とマネジメントの両面に能力を有する人材育成の方法として、非常に参考になった。一方、ドイツは、歴史的なクラフトマンシップやマイスター制度に由来して、大学における研究成果を実社会で役に立てるという考え方や風土の上に、このような仕組みが時間をかけて構築

されており、日本では容易には適用できないことも事実である。

Steinbeis の活動について 15 年前までは、日本の大学に説明をしても理解を得られなかったが、2005、6 年くらいから、大学、とくに若い先生が産業界への貢献を意識するようになり、ゆるやかではあるが変化はしているとのコメントがあった。

### 4. おわりに

Steinbeis は、ベーシックな技術やアイデアを産業界と結びつけるための仕組みを有し、併設する大学では、企業との連携の下に人材育成を進めている。この領域に携わる者としては、ダイナミックな市場変化や技術革新に対応できる産学連携や人材育成の仕組みが具現化されていることに、彼我の差を実感した。更に、Steinbeis の考え方を聞くにつけ、いかに教育機関、そして自立した個人が、早いうちから産業界を意識して連携しているかが印象深く、この視点では米国と同じ、あるいは先行していると感じている。今後も継続的に情報交換、意見交換をしながら、日本の環境も考慮しつつ、グローバルな競争力を強化できる、新たな仕組みや活動のあり方を検討し、発展させていきたい。

また、EC 主催の European conference 及び CEN 会議など各種会議に参加して、EC が自ら戦略をたて、リーダーシップを発揮し、どのように産学官へ啓発していくのが垣間見えた。IoT や Industrie4.0 といった新テクノロジーを意識した具体的なテーマを進めており、大きくは Digital single market として欧州の産業革命以来の復権を狙うようにも感じられ、欧州の総力をあげた取り組みとして注目していきたいと思う。

#### 【脚注】

※ 2 PBL : Project-Based Learning

# Embedded Technology West 2015 (ETWest2015) 出展報告

SEC 企画グループ 主任  
荒川 明夫

IPA/SEC は、2015 年 6 月 10 日、11 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 2015 / 組込み総合技術展 関西 (ETWest2015)」に出展した。また、IPA セミナーを 2 日間 8 部構成で併催した。

## 1. 展示会概要

Embedded Technology 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 10 回目の開催となる。

## 2. 出展概要

IPA/SEC は、2007 年より事業成果の普及・啓発を目的として、本展示会に出展している。

本年は、IPA 展示ブースでのブースプレゼンを 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 13 回、別会場では IPA セミナーを 2 日間で 8 回と、計 47 回のセッションを実施した。



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

## 3. IPA 展示ブース

本出展では、ブースプレゼン・パネル展示・資料配布などを実施するブースに加え、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、「ソフトウェア高信頼化への取り組み」や「高信頼化を確保するための手法」など、SEC 事業についての紹介をはじめ、「アジャイル開発」や「共通フレーム」など、ソフトウェア開発の参考にさせていただきたい情報を紹介したほか、「情報処理技術者試験」、「i コンピテンシ ディクショナリ」といった、IT 人材を育成するための事業など、広く IPA 全体の取り組みについても紹介した。とくに、組込みシステム開発に深く関連する自動車や医療機器の情報セキュリティなどセキュリティに関する事業に関しては、来場者の関心が高かった。

また、IPA 展示ブースでは、各展示コーナーのほかに、関連資料の配布、ブースプレゼン (26 セッション / 2 日間)、SEC 先端技術入門ゼミを実施した。

## 4. SEC 先端技術入門ゼミ

IPA 展示ブースと隣接した SEC 先端技術入門ゼミブースにて、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初心者向けのチュートリアルを実施した。

本講義は、昨年秋に出展した ET2014 での新企画として実施し、今回 ETWest2015 では第二回目の試みとなる。2日間で 560 名と、多くの方に聴講いただいた。

セミナーやカンファレンスと異なり、受講者が直接、講師に質問を投げられるこの企画は、受講者だけでなく講師にも好評で、「学校の講義形式であったため、受講者の反応がよくわかった」や「受講者の理解度に応じて講義を展開していった」という意見が寄せられた。



## 5. IPA セミナー

展示会場に隣接するタワー C では、本年も 2 日間にわたり、IPA セミナーを 8 部構成で開催し、延べ 440 名の参加があった。



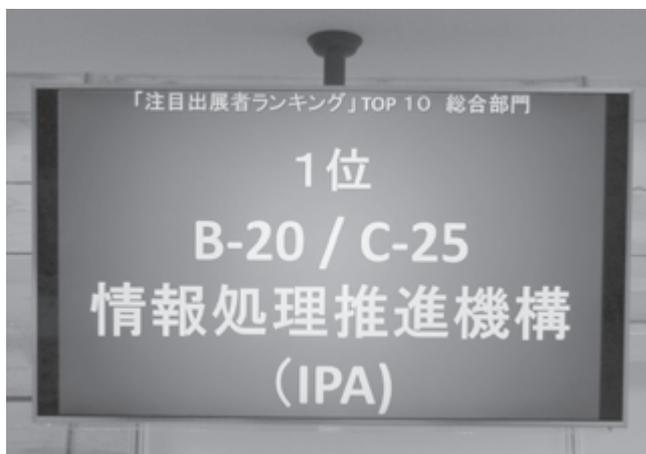
今回の IPA セミナーは、1 日目に「IoT 時代のソフトウェアエンジニアリング」や「定量データに基づく開発マネジメントで IoT 時代の信頼性・生産性向上を目指せ！」など、IoT 時代におけるソフトウェア開発の信頼性・生産性向上をテーマに講演したほか、システムにお

ける障害の再発防止・未然防止をテーマとした、「障害事例から学ぶ高信頼化へのアプローチ」、「大規模・複雑化した組込みシステムのための障害診断手法」について、紹介した。

2 日目は、「環境変化の適応から考えるサイバーセキュリティ対策」から「障害解析から再発防止・未然防止につなげる活動事例と教訓を共有する仕組み」、「ソフトウェア開発データ白書」などの取り組みを紹介し、昨年の ET2014 において記者発表を行った「モデルベースアプローチに基づく障害原因診断手法」について、講演した。

## 6. ETWest2015 を振り返って

今回の出展は、過去の出展の中で、最も多くの方に IPA 展示ブースに足をお運びいただいた。アンケートや展示会場で来場者の方から多くのご意見をいただいたので、次回以降の出展の参考とし、今後の IPA/SEC 事業活動に反映していきたい。例年どおりの出展方法ではなく、「SEC 先端技術入門ゼミ」のように、新たな企画にも挑戦し、来場者・参加者の理解度・満足度の向上に努めていく。



ETWest2015 IPA/SEC ウェブサイト

<http://www.ipa.go.jp/sec/events/20150610.html>

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

## 情報システムの事故データ

# 情報システムの障害状況 2015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

2015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合計21件で月平均3.5件となった。平均的な値に対しやや多い値である。システム更改作業直後の事故及び長期間認識されずに運用されてきたエラーが発覚した事故などが発生している。

## 1. はじめに

本稿では、2015年1月から6月までの2015年前半の半年間に報道された情報システムの障害状況をとりまとめて報告する。まず、次章で今期の概況について述べ、続く3, 4章では、今期の障害事例で多く見られたシステム更改作業直後の事故及び長期間認識されずに運用されてきたエラーが発覚した事故について述べる。

## 2. 2015年前半の概況

2015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21件となった。その全体は表1に示す通りであり、障害発生件数を月平均にすると3.5件/月となる。

今期の事例の中には、新しいシステムへの更改を行った直後に起こる初期不良による障害が6件報告されている(事例1501、1502、1506、1510、1517、1519)。また、以前からエラーが生じていたにもかかわらず、認知されずに



図1 情報システムの障害発生件数の推移

長期間放置されてきたエラーが偶然発見された事故が4件発生(事例1503、1508、1509、1512)している。この種の事故はこれまでも例はあるが、今期は比較的集中した。

また、これまでは対象外としてきた組込みシステム関連の事故の報道も増えてきており、障害一覧表の別表として記録した。今期には、家庭で使われていたテレビがある日突然電源のON、OFFを繰り返すという事故が起こった。テレビ受像機に内蔵されている組込みソフトウェアを更新するために、放送波の隙間を使ってブロードキャストされた更新データを、各受像機側が自分に必要なデータを取り込むことによってソフトウェアを更新する仕組みが作られている。今回は受像機側のデータ取り込み機能に不具合があり、本来取り込むべきでないデータを取り込んでしまった結果事故になった。正に、私たちの身近にIoTの時代(この事例はインターネット経由では無いが)が迫っていることを実感させる事故であった。この連載においては、組込みソフトウェアについては記録の対象外としてきた[松田2011]が、見直しの時期を迎えているのかも知れない。ただ、組込みソフトウェアの障害情報をどのような手段によって広く収集するかは大きな課題である。

## 3. システム更改を契機とする事故

今年7月にニューヨーク証券取引所の取引システムが障害を起こし、3時間半にわたって取引の全面停止を招いたが、その原因はソフトウェアの更新ミスであったと伝えられている。この例に限らず、新システムへの更改作業の直後は障害を起こしやすい。

新システムへの更改は、そのための作業時間を確保するために、しばしば年末・年始や連休など長期休業のタイミングを捉えて実施されるため、長期休み明け直後にシステムの事故が発生することはよくある。システム更改作業にあたっては周到な準備と慎重な作業によって障害発生を回

避することが必要である。

今期に発生した事例 1501 は、正月休みの間に更改した市役所の住民基本台帳システムが、休み明けの正月 5 日に運用を開始したところ、証明書発行に数時間を要する障害を起こしたものである。これは、同時に実行できる印刷命令の上限値の設定が過小であったため、上限値を超える大量の印刷命令が次々と待ち状態のまま滞留して、全体の処

理に大きな遅延をもたらした事故である。原因は、新システムにおいて利用されることになった印刷パッケージソフトの仕様の把握が不十分であったため、印刷命令の上限値の設定が不適切であったと伝えられている。

また、事例 1502 は事例 1501 と同じ更改システムで発生した障害であるが、旧システムから新システムへのデータの移行にミスがあった。本来移すべき旧システムのデー

表 1 2015 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				
1501	大阪市住民基本台帳システム	2015	1	5	9時00分	証明書発行システムで障害が発生。発行窓口を設置された約3,200端末のすべてで遅延が発生した。約3万5千件の書類発行が最大で2時間遅れた。	年明けから「基幹系システム統合基盤」が稼働したが、印刷サーバの設定ミス（同時に実行できる印刷命令数の設定ミス）により、処理遅延が発生した。昼休み時間帯に、設定を暫定的に変更したところ、遅延は解消した。設定ミスの原因は、帳票作成用パッケージソフトウェアの仕様を把握していなかったため。	設定ミス	<ul style="list-style-type: none"> <li>日経コンピュータ 電子版 (2015.1.8)</li> <li>日経コンピュータ (2015.2.19号)</li> </ul>
		2015	1	5	14時00分				
1502	大阪市住民基本台帳システム	2015	1	5		住民票の写し(世帯連記式)の592件で、日付を誤って記載していた。	原因は、1月5日から稼働した新システム「住民基本台帳システム」へのデータ移行時にミスがあったため。本来「個人の住定日」を記載すべき位置に、「世帯の住定日」が記載されていた。1月5日の業務終了後に修正処理を行ったため、1月6日以降の発行分には誤りはない。誤った住民票の写しを取得した人には個別に連絡を取り、正しいものを交付し直すことで対応した。	データ移行ミス	<ul style="list-style-type: none"> <li>日経コンピュータ 電子版 (2015.1.8)</li> <li>日経コンピュータ (2015.2.19号)</li> </ul>
		2015	1	5					
1503	住友生命 長期継続 配当金の 計算プログラム	2010				一部の保険契約で誤った配当金を支払っていた。629件で合計約1千万円分少なく払い、798件で合計1億5千万円分多く払っていた。	2010年に長期継続配当金の計算プログラムを変更した際に設計ミスがあった。従来のプログラムを2つに分割し、連携して計算する仕組みに変えたが、データ受け渡しに関する設計やテストが不十分だったため、誤計算が起きた。再発防止のため、開発ルールを見直す。誤計算は2010年から5年近く発生していたが、見過ごされていた。2014年の年末調整シーズンに、同社の担当者が偶然発見した。	ソフトウェア障害	<ul style="list-style-type: none"> <li>住生プレスリリース (2015.1.20)</li> <li>朝日新聞朝刊 (2015.1.21)</li> <li>日本経済新聞朝刊 (2015.1.21)</li> <li>日経コンピュータ 電子版 (2015.1.21)</li> </ul> ※障害発生日時は2010年であるが、影響が判明した日時に基づき掲載。
		2015	1	20					
1504	三菱UFJ信託銀行ATM	2015	1	30	10時20分	全国で、最大およそ160台のATMでの取り引きや、最大およそ124万枚のキャッシュカードを使った提携先の金融機関での取り引きが一時、できなくなった。	ホストシステムの障害に伴い、コンビニATMのE-netを除く提携金融機関との接続が不通となった。	不明	<ul style="list-style-type: none"> <li>NHK ニュース (2015.1.30)</li> <li>三菱UFJ信託銀行プレスリリース (2015.1.30)</li> </ul>
		2015	1	30	12時20分				
1505	スカイマーク 予約、搭乗手続き管理システム	2015	2	6	9時30分	予約や搭乗手続きを管理する社内のシステムにトラブルが発生した。搭乗時の自動チェックインや発券、ホームページでの予約ができなくなった。合計8便に15分以上、最大で1時間13分の遅れが出た。	電源の不具合で社内ネットワークがダウンした。詳しい原因は調査中。	ハードウェア障害	<ul style="list-style-type: none"> <li>NHK ニュース (2015.2.6)</li> <li>日本経済新聞夕刊 (2015.2.6)</li> </ul>
		2015	2	6	11時40分				
1506	徳島県牟岐町 防災サービス	2015	2	6	10時25分	震度5強以上の地震の際に届く県の防災サービス「すだちくんメール」が、同じ登録者に複数届いたり、地震発生から7時間後に届いたりした。2015年1月末の登録者は2万5,700人。	メールを配信するプログラムの設定ミスが原因。また、多くのメールを一斉送信しようとしたことで、サーバに過剰な負荷がかかり、メールの配信が遅延した。同サービスは2010年5月から運用されているが、実際に使用されたのは今回が初めて。	ソフトウェア障害	読売新聞電子版 (2015.2.9)
		2015	2	6					

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				
1507	埼玉県 富士見市 住民記録 /国民健康 保険システム	2015	2	19	8時30分	住民票発行や転入/転出などの手続きができなくなった。待機用のサーバを起動させ、証明書の発行業務に対応したが、25件の手続きが滞った。富士見市は、後日郵送したり、依頼者の自宅を訪問したりして対応した。	トラブル前日の2月18日に、同システムのデータベースに対してディスク容量を上回るデータ量の書き込み処理を実行したため、システムが停止状態となった。マンション住民の地番を修正した場合、関連情報として同じマンションに入居する他の住民のデータをログに記録する。今回、数百世帯の大規模マンションの修正があったため、ログの大きさがこれまでにないほど大きいものとなり、決められた容量(100ギガバイト)を超えることとなった。今回のケースは特殊なため、再発防止策としてディスク容量の変更はしないが、今後、ログの容量を削減する対策を行う。	データベース 障害	・日経コンピュータ (2015.5.14号)
		2015	2	19	11時00分				
1508	日新火災 海上保険 保険金支 払いシス テム	1999	10			システム上の不備で契約者から保険料を取り過ぎていたと発表した。規模は最大で、1,201人から約1億円の過剰徴収。	担当者が保険金支払いシステムに事故の種類を入力し忘れ、翌年以降、多く保険料を支払わなくてはならない等級にしてしまった。 1999年10月から2010年6月までに車の盗難や火災、落書きなどで保険金の支払いを受けた契約者が該当する。2014年12月、損保ジャパン日本興亜で同じ不備が分かり、調査していた。	入力 ミス (人的 ミス)	・朝日新聞電子版 (2015.2.26)  ※障害発生日時は 1999年であるが、 影響が判明した日 時に基づき掲載。
		2015	2	26					
1509	セゾン自 動車火災 保険 保険金支 払いシス テム	2001	7			上記1508と同様、担当者が保険金支払いシステムに事故の種類を入力し忘れ、翌年以降、多く保険料を支払わなくてはならない等級にしてしまった。 2001年7月から2010年1月までに車両保険の保険金の支払いを受けた契約者が該当する。 2014年12月、損保ジャパン日本興亜で同じ不備が分かり、調査していた。	入力 ミス (人的 ミス)	・朝日新聞電子版 (2015.2.26)  ※障害発生日時は 2001年であるが、 影響が判明した日 時に基づき掲載。	
		2015	2	26					
1510	東京消防 庁 司令管制 システム	2015	3	4	11時23分	午前11時23分からの9分間、東京23区内からの119番通報が受けられない状態になっていた。その間、人命に関わるような通報はなかった。	新たに導入されたシステムにプログラムミスがあった。異常が起きていないかチェックするために、装置に自動で接続する動作と、同じ装置に手動で接続する動作が重なったため、システムにエラーが起きていた。エラーが起きた場合は、自動で予備の装置に切り替わるはずだが、今回導入されたシステムそのものにプログラムのミスがあったため、切り替わらなかった。	ソフト ウェア 障害	・日本経済新聞電子 版(2015.3.4) ・日本経済新聞電子 版(2015.3.10) ・産経ニュース (2015.3.4) ・NHKニュース (2015.3.4) ・NHKニュース (2015.3.9) ・毎日新聞朝刊 (2015.3.10) ・日経コンピュータ (2015.7.9号)
		2015	3	4	11時32分				
1511	神奈川県 エリア 110番通 報(ジュ ピターテ レコム)	2014	11	12		神奈川県の一部地域の利用者が110番通報すると、4回に1回の割合で千葉県警鎌ヶ谷署につながっていた。誤接続は146回に及んだ。影響のあった世帯は42,617世帯。	2014年11月12日にIP電話サービスのサーバ内の設定を変更した際、誤って別データを登録したことによるもの。	設定 ミス	・朝日新聞電子版 (2015.4.3) ・ジュピターテレ コム社プレスリリ ース(2015.4.2)
		2015	3	27					
1512	ブルデン シャル生 命保険	2009				ブルデンシャル生命保険は2009年～14年の間に、誤った生命保険料の控除証明書を送ったり、証明書を発行していなかった契約者がいると発表した。対象は5,367人分の6,226契約分。	過去に旧日産生命保険と合併した際の、日産生命保険側のシステム不備が原因。	不明	・朝日新聞朝刊 (2015.3.28)  ※障害発生日時は 2009年であるが、 影響が判明した日 時に基づき掲載。
		2015	3	28					

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				
1513	JR グループ IC 乗車券	2015	4	3	14 時 20 分	みどりの窓口や自動券売機で、IC 乗車券による定期券の継続購入ができなくなった。影響があったのは東日本「SUICA」、東海「TOICA」、西日本「ICOCA」、北海道「Kitaca」。全国で 1,000 駅以上でトラブルが発生した。	影響人数、原因等調査中。当日は磁気定期券を発行し、後日切り替えるなどの対応を取った。なお、東海のトラブルのみ、発生時刻、回復時刻は 15:12 ~ 15:29。	不明	<ul style="list-style-type: none"> <li>朝日新聞朝刊 (2015.4.4)</li> <li>日本経済新聞朝刊 (2015.4.4)</li> <li>日経コンピュータ PLUS (2015.4.4)</li> </ul>
		2015	4	3	15 時 35 分				
1514	東京消防庁 司令管制システム (2 回目)	2015	4	14	17 時 00 分	4 時間にわたって着信してもすぐに切れてしまう状態となった。この時間帯の通報 712 件のうち、59 件で着信後すぐに電話が切れ、担当者が記録されている番号に折り返した。59 件のうち 8 件では再度通報があり、救急隊が出動した。	通報を受ける装置の設定ミスが原因。3 月 24 日に固定電話用の回線を一部廃止したにもかかわらず、システムの設定を変えず、廃止した回線に接続確認のためのデータを送り続けたため、バッファオーバーフローにより通報がつながりにくくなった。障害発生当時は、予備の受信装置に切り替えたところ、着信の不具合は解消された。当面の対策として、これまでに蓄積された制御メッセージをクリアする措置を 4 月 17 日に実施。今後、システム機能の改修や、障害発生時の対処策の改善などをはかる。	設定ミス	<ul style="list-style-type: none"> <li>朝日新聞朝刊 (2015.4.15)</li> <li>日本経済新聞夕刊 (2015.4.15)</li> <li>NHK ニュース (2015.4.15)</li> <li>日本経済新聞朝刊 (2015.4.22)</li> <li>朝日新聞電子版 (2015.4.22)</li> <li>日経コンピュータ 電子版 (2015.4.23)</li> <li>日経コンピュータ (2015.7.9 号)</li> </ul>
		2015	4	14	21 時 00 分				
1515	気象庁 ホームページ	2015	4	16	10 時 00 分	気象庁のホームページを閲覧することができなくなった。天気予報を行うシステムは正常で、気象観測などに影響はなかった。	障害が起きたのは外部に設けられたキャッシュサーバ。台風や地震が発生した際、アクセスが集中して負荷がかかるため、コンテンツをコピーし、外部のサーバから配信することで、負荷を分散している。このコピーの段階で不具合が生じていた。	ハードウェア障害	<ul style="list-style-type: none"> <li>朝日新聞夕刊 (2015.4.16)</li> <li>日本経済新聞夕刊 (2015.4.16)</li> </ul>
		2015	4	16	13 時 00 分				
1516	かんぽ生命 かんぽ総合情報システム	2015	5	1	13 時 35 分	全国約 2 万 100 の郵便局で新規契約の申し込みができなくなった。また、約 8,000 の郵便局で、一部の保険金や貸付金の返済ができなくなった。	郵便局の窓口にあるパソコンや営業担当者が持ち運ぶタブレット端末に職員がログインできなくなった。郵便局での窓口業務を終えた午後 4 時以降にシステム全体をログインし直したところ、復旧した。システムを統括するサーバに不具合があった。	ハードウェア障害	<ul style="list-style-type: none"> <li>日本経済新聞電子版 (2015.5.1)</li> <li>日本経済新聞朝刊 (2015.5.2)</li> <li>読売新聞朝刊 (2015.5.2)</li> <li>NHK ニュース (2015.5.1)</li> <li>かんぽ生命プレスリリース (2015.5.1)</li> </ul>
		2015	5	1	18 時 15 分				
1517	富士火災基幹情報システム	2015	5	12		5 月 12 日から断続的にシステム障害が発生し、保険契約の確認や事故対応などに影響が出た。5 月 15 日にシステムを全面停止し、5 月 18 日に復旧させた。この間は代替システムや手作業で業務を実施したが、保険証券の送付など一部の業務が遅延した。	5 月 11 日に実施したプログラムの変更が原因。プログラムに不具合があったため、夜間バッチが中断し、処理を積み残した。翌日以降、バッチ処理と並行してプログラムの修正を試みたもののうまくいかず、処理能力に余裕がなくなり、保険申込書の新規登録や事故の受付等のオンラインが断続的に停止することとなった。5 月 15 日にシステムを全面停止し復旧に専念、5 月 18 日に再稼働した。ただし、負荷制御のため、しばらくは代理店にアクセス制限を掛けながらの運用となった。	ソフトウェア障害	<ul style="list-style-type: none"> <li>富士火災プレスリリース (2015.5.15)</li> <li>日本経済新聞電子版 (2015.5.15)</li> <li>日経コンピュータ 電子版 (2015.5.15)</li> <li>日本経済新聞朝刊 (2015.5.16)</li> <li>日経コンピュータ 電子版 (2015.5.18)</li> </ul>
		2015	5	18	9 時 00 分				
1518	しんきん共同センター	2015	5	21	朝	東北と関東甲信越の計 50 の信用金庫で、現金自動出入機 (ATM) や窓口での入出金などができなくなった。障害が発生したのは、東北 6 県にある全 27 の信金のほか、群馬、埼玉、千葉、神奈川、長野、新潟、東京の計 23 の信金。	通常は午前 5 時頃に自動的に立ち上がる入出金などの管理システムが、この日は稼働しなかった。午前 5 時になっても前日の処理が終わらなかったことで、システム障害の発生に気づいた。運用プログラムの一部に不具合があったことは確認できたが、詳しい原因については調査中。	ソフトウェア障害	<ul style="list-style-type: none"> <li>朝日新聞夕刊 (2015.5.21)</li> <li>日本経済新聞夕刊 (2015.5.21)</li> <li>読売新聞電子版 (2015.5.21)</li> <li>毎日新聞電子版 (2015.5.21)</li> </ul>
		2015	5	21	10 時 49 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				
1519	ニトリネット	2015	6	17		6月17日から6月23日までサイトが閉鎖された。	商品の配達日を確認できるようにするなど、6月17日にサイトを刷新する予定だったが、プログラムの不具合でサイトを閉鎖した。閉鎖中は、正常に稼働していた楽天市場などに誘導して対応した。	ソフトウェア障害	<ul style="list-style-type: none"> <li>・ニトリプレスリリース (2015.6.17)</li> <li>・日本経済新聞朝刊 (2015.6.23)</li> <li>・日本経済新聞電子版 (2015.6.23)</li> </ul>
		2015	6	23					
1520	ピーチ・アビエーション予約・搭乗手続きシステム	2015	6	29	17時30分	29日の国内線では9便が欠航、8便が最大2時間ほど遅れ、2,665人に影響が出た。30日は、国内線16便がおよそ1時間半ほど遅れ、約2,200人に影響が出た。また、障害発生中、インターネット予約もできなかった。	原因究明中。	不明	<ul style="list-style-type: none"> <li>・朝日新聞電子版 (2015.6.29)</li> <li>・朝日新聞電子版 (2015.6.30)</li> <li>・NHK ニュース (2015.6.30)</li> <li>・日経コンピュータ電子版 (2015.7.2)</li> </ul>
		2015	6	30	12時00分				
1521	日本ケンタッキーフライドチキンクレジットカード決済システム	2015	1	29		クレジットカードを使用した一部の顧客に対し、利用代金を誤って請求した。 発生店舗数…290 店舗 発生件数…1,514 件 誤請求額…1,895,373 円 ※該当する顧客へは返金を行う。	クレジットカード決済 POS ターミナルが導入された店舗の一部において、搭載されたクレジットカード読み込み制御ソフト (ドライバー) に誤りがあった。ドライバーを改修し、十分な検証を行った上で、クレジット決済は7月下旬に復旧予定。	ソフトウェア障害	<ul style="list-style-type: none"> <li>・日本ケンタッキー・フライド・チキン株式会社プレスリリース (2015.6.30)</li> <li>・日本経済新聞夕刊 (2015.6.30)</li> <li>・日経コンピュータ電子版 (2015.7.2)</li> </ul> ※障害発生日時は2015年1月であるが、影響が判明した日時に基づき掲載。

タとは異なった別のデータが新システムのデータとして移行されたために、間違ったデータが印字された住民票の写しが出力されてしまった事例である。

システム更改時の移行についてはトラブルが多いため、この連載においても以前に留意点について指摘していた [大高 2013]。その中で、旧システムからの移行設計として、旧システムと新システムのデータ項目の対応付けや整合を取るための変換について注意喚起をした。今回の事例では、新旧システムでデータベースの構成が大きく変更され、一対一の単純なデータ移行ではなかったために、データ移行のためのプログラムのロジックに不備があって事故につながった。本事例には該当しないが、新旧システム間でのデータ移行に際しては、長期運用によって蓄積された不要データの削除などのデータクレンジングも併せて実施されることもあり、これらを含めたデータ移行設計や移行の単位、移行データ量の見積りと移行に必要な処理時間の推定など、実施計画を準備することが重要である。

#### 4. 長期間のエラー放置

事例 1503 は、業務プログラムの一つを変更した際に、設計ミスが発生し、テストでもその誤りが発見されずそのまま運用が続けられた例である。すなわち、3章で述べたシステムの更改による事故の一例であるが、その誤りが5年近く見過ごされたまま運用されていたことが偶然発見されるという、やや特異な事例である。この事例では、旧業

務プログラムを2つに分解して、連携して処理する仕組みにプログラム構成を変更したと伝えられており、いわゆるリグレッションテストを十分に実施していれば発見できた誤りではないか、と思われる。システムの更改の際は、その変更によって予想外の影響が現れていないかどうかを確認するテストも重要である。このためには、テスト項目と共に、そのためのテストプログラムやテストデータの蓄積・整備を十分に行い、リグレッションテストを効率的に、できれば自動的に実施できる体制を確立することが望ましい。

事例 1512 も同じく業務処理の誤りであるが、会社合併が行われる前の一方の会社のシステムの不備が原因と伝えられている。誤り発見のきっかけや原因の詳細は不明であるが、処理誤りは2009年までさかのぼり、最近まで誤りに気付かずに長期間運用されてきた事例である。

事例 1508 および事例 1509 は担当者のデータ入力ミスが原因であるが、処理誤りの発生は事例 1508 では1999年まで、事例 1509 では2001年までそれぞれさかのぼる可能性があると言われている。同業他社でこの種の誤りが発見され、それをきっかけに自社システムを点検したことによって発見された。ほかでの失敗の経験を活かして、誤りを発見できた好例である。

不具合によってシステムがダウンするなどの現象が発生すれば、誤りは検知され修正されるが、これらの事例のように一見システムは正常に運転されているにもかかわらず得られた結果が誤っているケースでは、その発見・検証には業務の観点からの検証が必要である。

別表 組込みシステム関連の事故の報道

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	ゆりかもめエレベータ (竹芝駅)	2015	2	15	13時30分	ホーム階から改札階までの高さ6mを、40分かけてゆっくり降り続けるトラブルがあった。(通常は20～30秒で移動する距離)	原因不明	不明	・日本経済新聞夕刊 (2015.2.16)
		2015	2	15	14時10分				
2	首都高速道路 ETC システム	2015	2	16	13時00分	首都高速道路は、火災に伴う停電で一部の入口のETC機器に障害が発生し、不適切な課金をした可能性があるとして発表した。	火災による停電の影響で、通過が記録されず、出口の記録だけが残るため、最大料金(普通車930円)が課金されたり、逆に課金されなかった可能性がある。コールセンターで申告を受け付け、確認できれば記録を修正する。対象となったICは中央環状線外回り四つ木、船堀橋、同内回り清新町、平井大橋、四つ木。	停電	・朝日新聞朝刊 (2015.2.17)
3	三菱電機テレビ	2015	3	29	0時00分	視聴中または録画中に、電源がON/OFFを繰り返す不具合が発生した。対象製品は、2010年から2015年3月までに製造された120機種、約168万台。	原因は、対象製品のソフトウェアの不具合によるもの。他社の製品向けの更新情報を、誤って取り込んでしまう不具合があった。暫定処置として、データ配信の内容を変更してもらえよう依頼。本格対処として、自社のテレビのソフトウェアを更新した。	ソフトウェア障害	<ul style="list-style-type: none"> <li>三菱電機プレスリリース (2015.3.29)</li> <li>三菱電機プレスリリース (2015.4.9)</li> <li>朝日新聞朝刊 (2015.3.30)</li> <li>朝日新聞朝刊 (2015.4.10)</li> <li>日本経済新聞朝刊 (2015.3.30)</li> <li>日本経済新聞朝刊 (2015.4.10)</li> <li>日経産業新聞朝刊 (2015.4.10)</li> <li>日経コンピュータ (2015.4.30号)</li> </ul>
		2015	3	29	12時00分				
4	救急車・消防車の赤色灯	2003	1		救急車や消防車の赤色灯が点かないトラブルがあった。対象は2003年1月～09年3月に製造されたトヨタの救急車464台、日産の救急車171台、日産車をクラタが改造した消防車12台。事故や到着遅れはなかった。	サイレンと赤色灯を連動させる同じメーカーの装置にプログラムミスがあった。サイレンの音量を変えるモードにすると、赤色灯が点かなくなる可能性がある。	ソフトウェア障害	<ul style="list-style-type: none"> <li>朝日新聞朝刊 (2015.4.10)</li> </ul> ※障害発生日時は2003年であるが、影響が判明した日時に基づき掲載。	

## 5. むすび

2015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。4章で紹介したように、他のシステムの失敗事例を他山の石として誤りの検出に成功したように、事件事例の中から開発・運用にあたって参考にすべき多くの教訓を汲み取ることができる。今後も、これらの経験を社会の共通の財産として共有し、少しでも事故を防ぎ、安心・安全なIT社会に向けて地道な努力を続けていく必要がある。

IPA/SECでは様々な事故の原因や対策について多方面から考察を行い、業界横断的に利用可能な要素を抽出し「見える化」する活動の成果として、2014年5月に「情報処理システム高信頼化教訓集」[SEC1 2014][SEC2 2014]、2015年3月にはその改訂版として「情報処理システム高信頼化教訓集(2014年度版)」[SEC3 2015][SEC4 2015]を公表した。IPA/SECのこれまでの活動で蓄積されたソフトウェア・エンジニアリングに関する検討成果と関係付けて教訓

が整理されている。今後は、この活動を継続し、新たな教訓を更に追加すると共に、得られた教訓を関係者で広く共有し、活用を促す活動を推進していく予定である。システム障害の再発や影響拡大を防ぐために、経験者や関連事業者の方々に、この事業への積極的な参画と協力をぜひお願いしたい。

### 【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 2010年データ, SEC journal No.26, Vol.7, No.3, pp.102-104, 2011年10月
- [大高 2013] 大高浩・松田晃一：情報システムの障害状況 2012年後半データ, SEC journal No.32, Vol.9, No.1, pp.37-41, 2013年5月
- [SEC1 2014] 独立行政法人 情報処理推進機構 SEC：情報処理システム高信頼化教訓集 (ITサービス編), 2014年5月
- [SEC2 2014] 独立行政法人 情報処理推進機構 SEC：情報処理システム高信頼化教訓集 (製品・制御システム編), 2014年5月
- [SEC3 2015] 独立行政法人 情報処理推進機構 SEC：情報処理システム高信頼化教訓集 (2014年度版) (ITサービス編), 2015年3月
- [SEC4 2015] 独立行政法人 情報処理推進機構 SEC：情報処理システム高信頼化教訓集 (2014年度版) (製品・制御システム編), 2015年3月

# SWEBOK V3.0 日本語訳版<sup>※1</sup>の 連続紹介 —3の3

新谷 IT コンサルティング

新谷 勝利

## 1. はじめに

SWEBOK V3.0 連続紹介の3の1では以下について紹介した。

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

連続紹介の3の2では、SWEBOK V3.0になって新しく紹介された知識領域（以下の章名の右端に\*が付いていないもの）をそれらの全体像を見る形で説明した。以下に示すのはSWEBOK V3.0の章立てである。

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

最終回の今回は、SWEBOK 2004年版から存在していた章（章名の右端に\*のあるもの）を紹介する。前回同様この紹介にあたり、引用をこころよく承諾していただいた原文の発行元のIEEE及び日本語訳の発行元のオーム社に感謝する。

## 2. 第1章 ソフトウェア要求

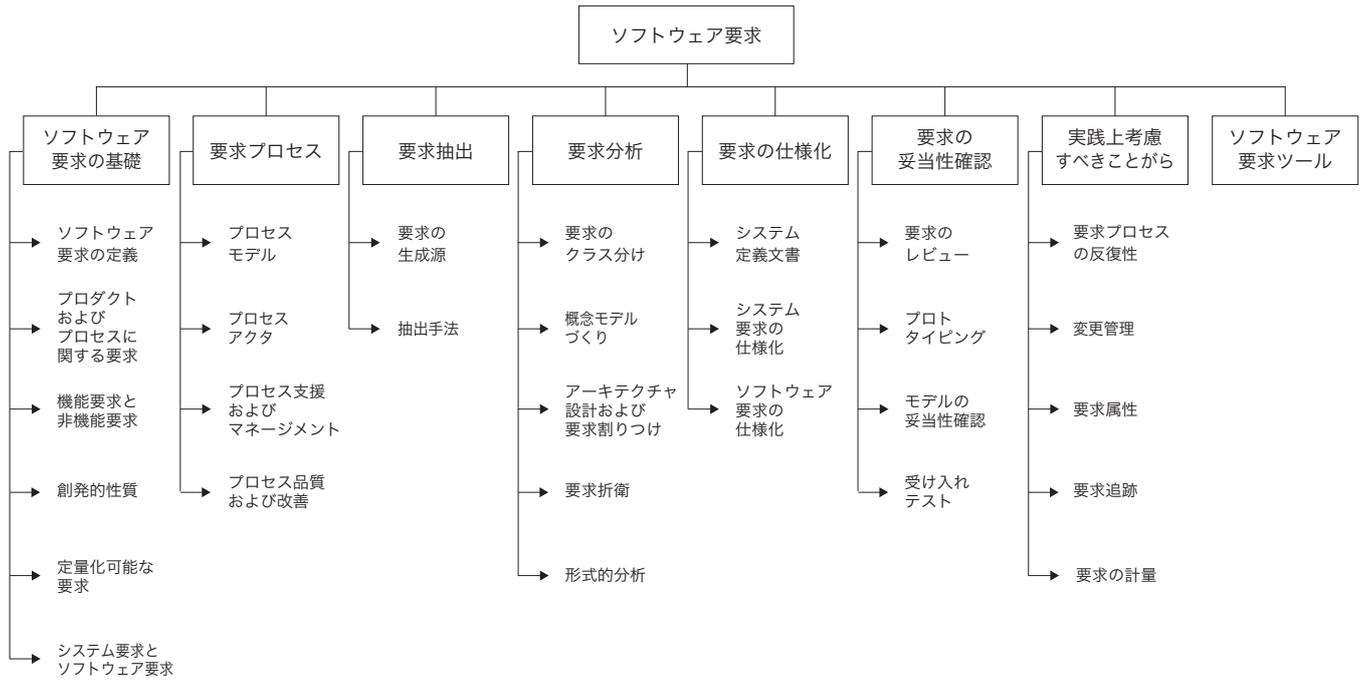
ソフトウェア要求ということであると、SWEBOK V3.0同様に日本語で提供されているものに、要求工学知識体系REBOK<sup>※5</sup>がある。SWEBOK V3.0の方が発行年が新しい分だけ引用文献に新しいものが追加されており、カバーされている範囲も広いものもある。REBOKは参照文献が多く、日本語翻訳を含む多くの書籍が紹介されている。ただし、SWEBOKはトピックスごとにそれがどの引用の何処を参照すればよいかのトレースができるようになっているが、REBOKは何が引用されているかのみでの紹介にとどまっているし、多くの参照文献は何処がどのように引用されているかのトレースは難しい。ページ数で明らかであるが具体的な技法の活用の仕方に関してはREBOKが詳細なので、先ずSWEBOKにて要求に関する全体を鳥瞰し、REBOKで具体的に学習してゆくという方法があるように思う。

## 3. 第2章 ソフトウェア設計

日本におけるソフトウェア開発の問題点として、SECの調査<sup>※6</sup>において、発注者及び受注者共に開発プロセスの上流に問題があることは認識されている。2章において紹介している、JISAにおけるソフトウェア要求に関連する基礎知識の体系化の動きや、後で述べるソフトウェアテスト

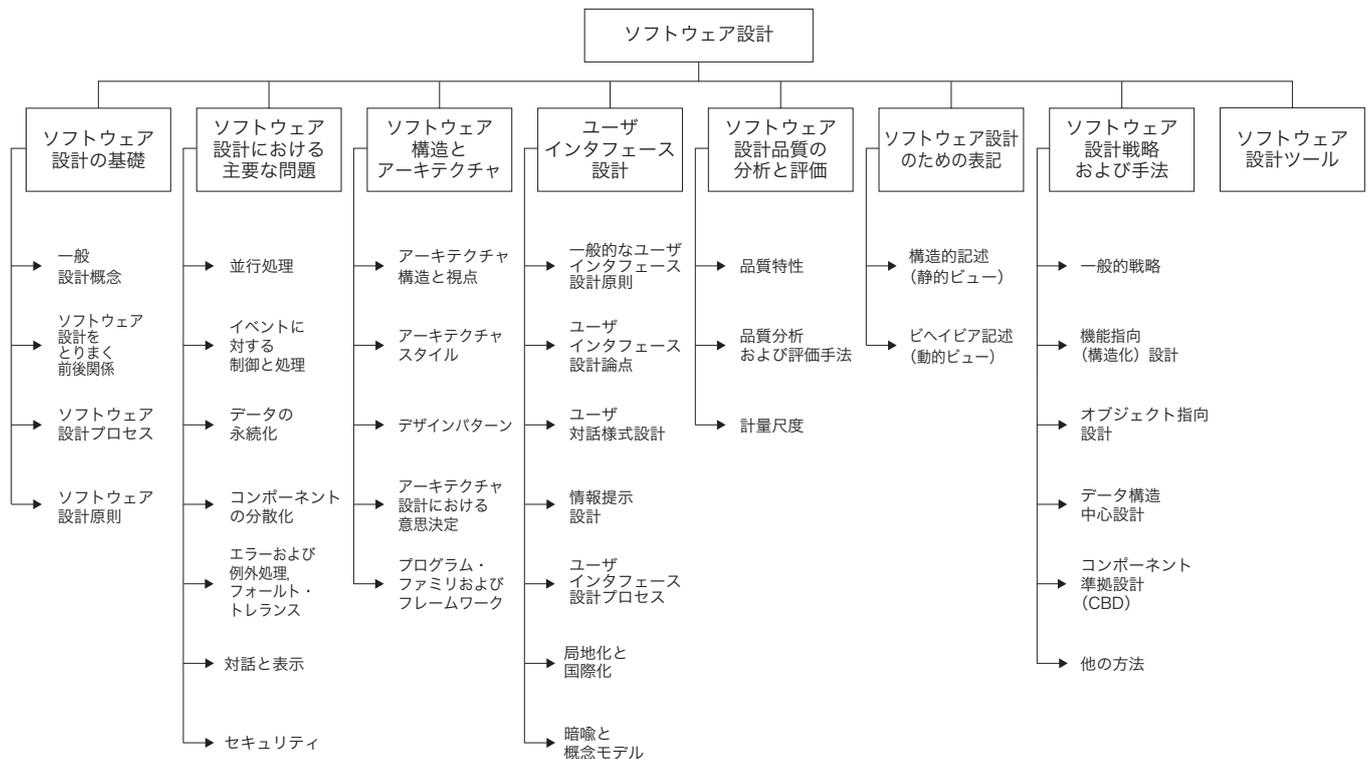
### 【脚注】

- ※1 松本吉弘訳、ソフトウェアエンジニアリング基礎知識体系—SWEBOK V3.0—、オーム社、2014年11月25日、ISBN978-4-274-50521-8
- ※2 Software Engineering, Report on a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th Oct., 1968、<http://homepages.cs.ncl.ac.uk/brian.randell/NATO/>
- ※3 [http://cisas.unipd.it/didactics/STS\\_school/Software\\_development/Trial\\_Version1\\_00\\_SWEBOK\\_Guide.pdf](http://cisas.unipd.it/didactics/STS_school/Software_development/Trial_Version1_00_SWEBOK_Guide.pdf)
- ※4 松本吉弘訳、ソフトウェアエンジニアリング基礎知識体系—SWEBOK 2004—、オーム社、2005年6月、ISBN 4-274-50029-2
- ※5 要求工学知識体系 REBOK 第1版、(一)情報サービス産業協会 REBOK 企画WG編、近代科学社、2011年6月、ISBN978-4-7649-0404-0
- ※6 「2012年度ソフトウェア産業の実態把握に関する調査」、2013年4月、<https://www.ipa.go.jp/sec/softwareengineering/reports/20130426.html>



ングに関する動きはあるものの、ソフトウェア設計に関しては組織的な具体的な動きが見えない。SWEBOK V3.0における参照資料も新しいものではないが、ユーザインターフェース設計への関心の高まり、アーキテクチャの重視など SEC における先進的技術適用事例の調査においても SWEBOK V3.0 の新知識領域と類似の傾向がある。今まで、設計手法は種々提案されてきており、ソフトウェア設計に

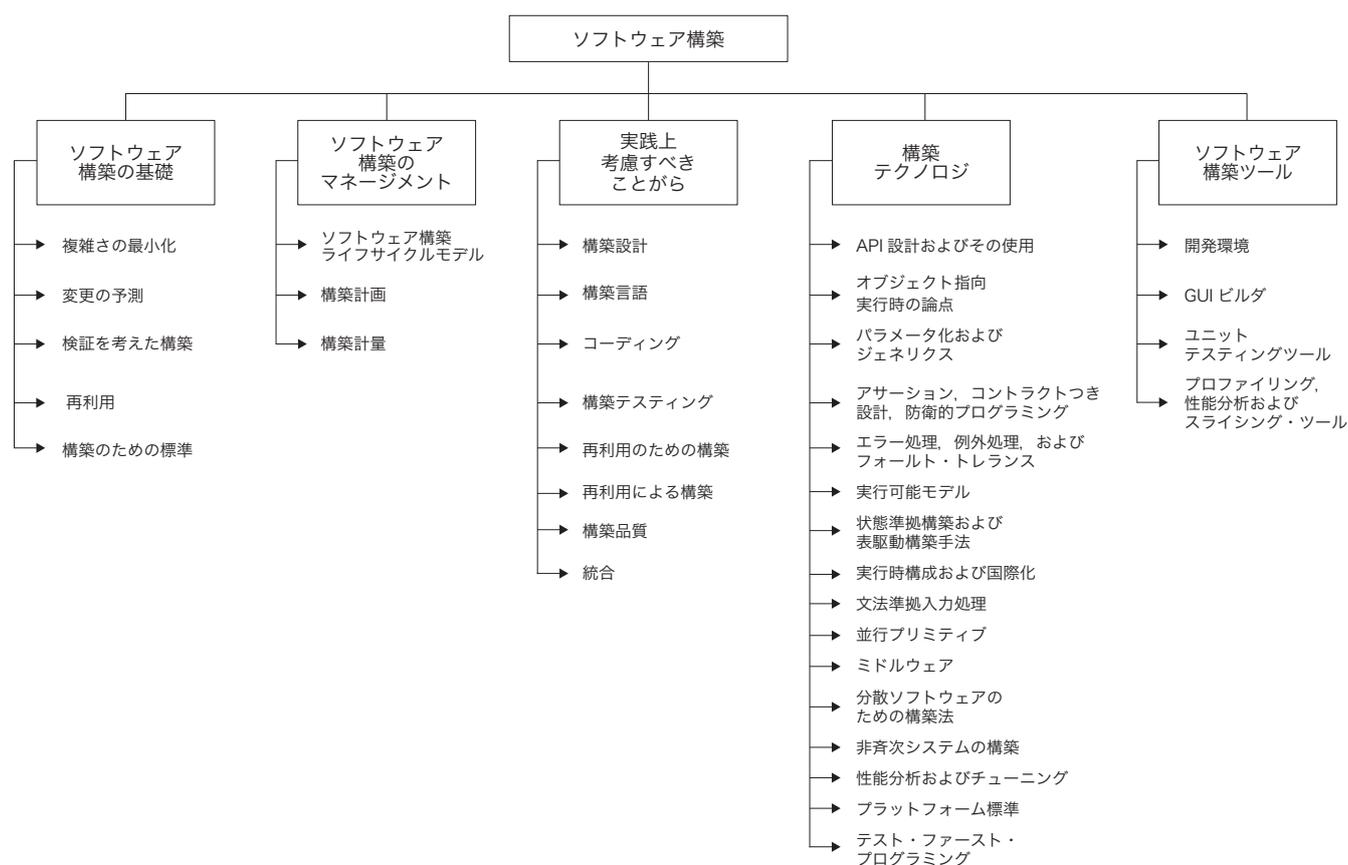
関しては、要求の実現のための記述の観点、及び、開発プロセスのポストモータムからソフトウェア設計の段階で何がどのように考慮されていればよかったかの観点から、今一度設計手法に関して新しい動きがあってもよいのではないか。アジャイル、DevOps、超高速開発、納入の無い開発、派生開発、SPLなどを支援する方法論として新設計方法論が出現することが期待される。



## 4. 第3章 ソフトウェア構築

当知識領域は、提供されるツールの有無と「構築テクノロジー」という副知識領域に関する経験の有無が大きく影響しているものと思われる。最近脚光を浴びているものにコーディング規約がある。実行されるものは構築されたコードであり、コードそのものがアセットとして認識され、再度再利用に目が向いているということである。プログラミング言語の利用に当たり一定の経験に基づくルール（コーディング規約と呼称）を付加することにより、「類似の誤り」

を再度発生させないという試みであり、ツールによるコーディング規約の強制化が良い影響を与えていると思われる。適用分野が限定されているのでSWEBOKには取り上げられていないが、MISRA（Motor Industry Software Reliability Association）が開発したC言語のためのソフトウェア設計標準規格は今や自動車業界のみならず組込みソフトウェア業界において安全性と可搬性（移植性）と信頼性を確保することに寄与しているようである。即ち、ソフトウェア構築にはルールを持ち込むことが容易であり、これは結果としてツールによるソフトウェア構築支援に寄与している。



## 5. 第4章 ソフトウェアテスト

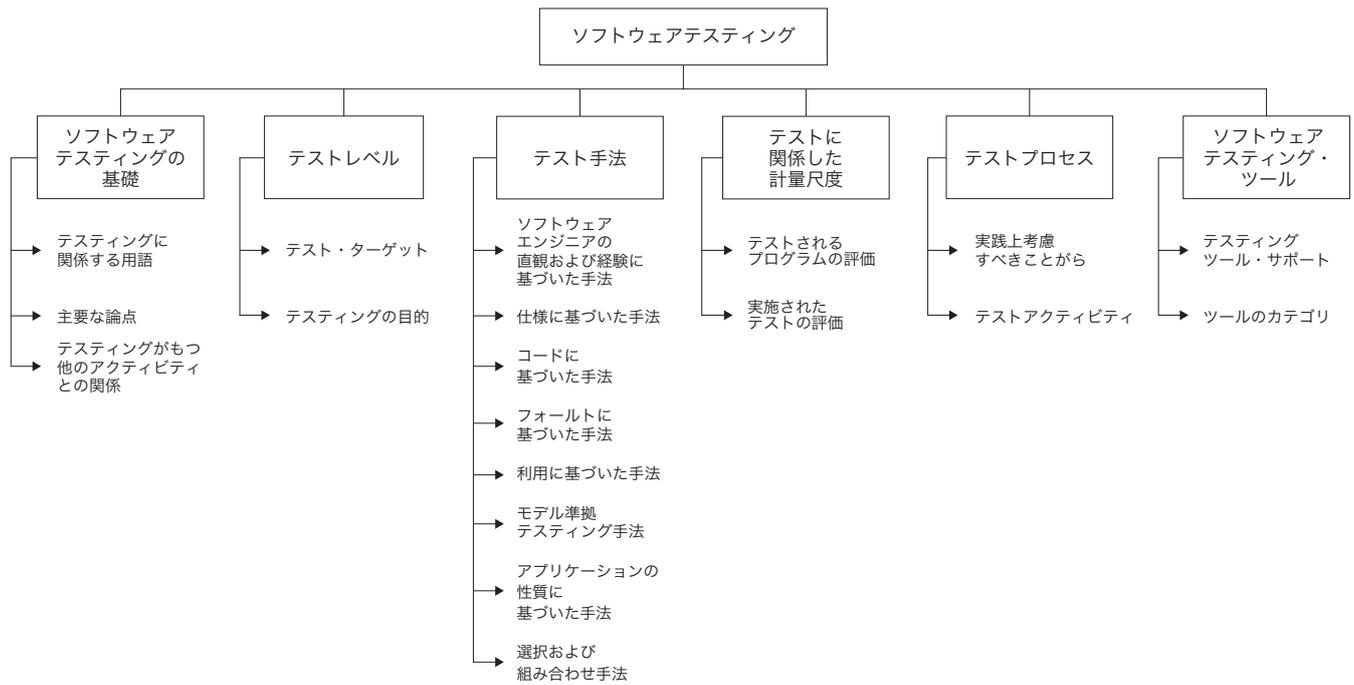
ソフトウェアエンジニアが最も多く参加している外部セミナーあるいはシンポジウムは、テストにかかわるものとプロセス評価・改善にかかわるものであるように見える。ここ数年 JaSST<sup>\*7</sup> のシンポジウムが全国各地で実施されている。なぜか、ソフトウェアテストに関する技術が体系化されることには時間がかかっており、参考文献にある ISO/IEC/IEEE 29119<sup>\*8</sup> はようやく 2013 年に正式に国際規格になった。ISO/IEC 33063<sup>\*9</sup> というソフトウェアテストの実施プロセスを評価する規約も国際規格として発行される最終段階にあり、今後ソフトウェアテストがソフトウェア開発プロセスの中で方法論として高く評価されるようになることが期待される。ソフトウェアテ

スティングの重要性を再認識させたのはマイクロソフト社におけるデイリービルドへのプロセス改善であり、それを可能にした体制及び方法論の変革であろう。

詳しくはマイケル・クスmanoなどによる Microsoft Secrets<sup>\*10</sup> を参照のこと。ソフトウェアテストの技術論の観点からは、参考文献にも記述されている Wipro 社、Cisco 社、IBM 社における知見をまとめたものが実践的なものである。

### 【脚注】

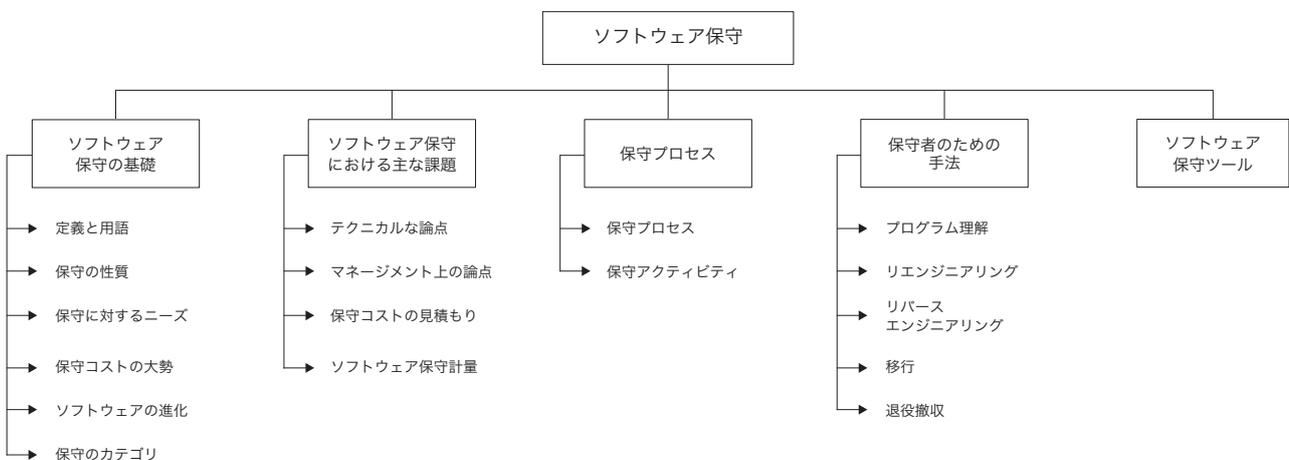
- \*7 <http://www.jasst.jp>、ソフトウェアテストシンポジウム
- \*8 <http://www.softwaretestingstandard.org>
- \*9 <http://www.softwaretestingstandard.org/33063.php>
- \*10 「マイクロソフト シークレット—勝ち続ける驚異の経営（上・下）」として日本経済新聞社より発行



## 6. 第5章 ソフトウェア保守

ソフトウェアテストは、開発プロセスのワークロード及びコスト配布の観点から、ライフサイクルの全ワークロード及びコストの半分以上を占めている。以上からの経験、知見が蓄積され ISO から国際規格として発行されるレベルにまでになっている。ところが、ソフトウェア保守という運用段階から使用停止するまでにおけるワークロード及びコスト配布は全ソフトウェアライフサイクルのマジョリティを占めているにもかかわらず、その知識領域としてはまだまだ認識が低くみえることが危惧される。とくに組込みソフトウェア業界においては、出荷後に機能追加が行われることにより製品及びソフトウェアのライフサイクルが長くできることを設計段階から考慮しようとい

う XDDP<sup>\*11</sup> という方法論が最近多くのエンジニアの関心を引きつけているが、まだ国際化にまでいたっておらず SWEBOK の参照文献に上がってきていない。今後英語での海外シンポジウムなどでの発表がなされることにより、ソフトウェア保守を考察したソフトウェア開発がグローバルに注目を浴び、推進されるものと思われる。ソフトウェアライフサイクルプロセスである ISO/IEC/IEEE 12207 及び 15288 においては、まだソフトウェア保守に関して十分な記述がなされてはおらず、参照資料においても ISO 文書としてはまだ十分開発が進んでいるとはいえない。ソフトウェア開発が本格化して数十年経ち、アセットとして保有されているソフトウェア量が増えるにつれ、スクラッチから開発するというより保守開発の量が増えることが予想され、今後この知識領域は発展するものと思われる。



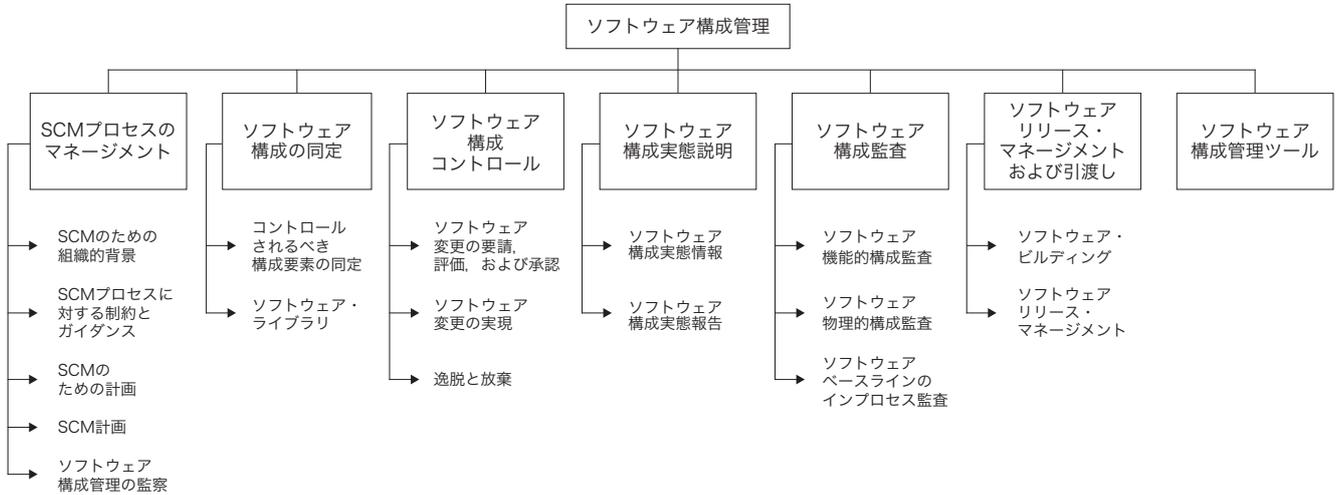
### 【脚注】

\*11 XDDP : eXtreme Derivative Development Process、派生開発に特化したアプローチ  
[http://affordd.jp/about\\_xddp5.shtml](http://affordd.jp/about_xddp5.shtml)、AFFORDD、派生開発推進協議会

## 7. 第6章 ソフトウェア構成管理

ソフトウェアがメインフレームでなされていた時代には、ライブラリが維持され、チェクイン、チェックアウトが

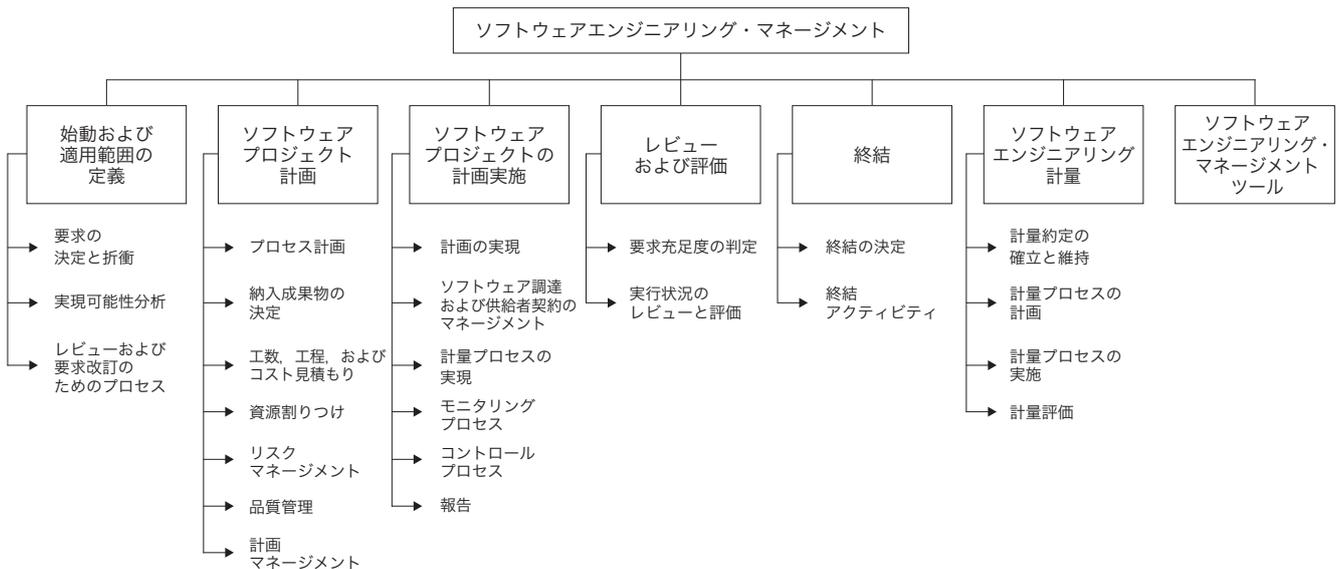
構成管理されるということは日常的に行われており、参照文献に記述されている IEEE 規格が技術的背景として維持されていた。技術的には確立しており、むしろ、ソフトウェア開発管理の観点からの推進が必要であろうし、それを容易にするツールが必要となる。



## 8. 第7章 ソフトウェアエンジニアリング・マネージメント

ソフトウェア開発はエンジニアリングか、という質問に関してはそうであると言いたいのが実態はエンジニアリングになりきれていないのではないであろうか？ 当知識領域において PMBOK が引用されていることに留意する必要がある。始動し、計画し、実施し、評価し、そしてプロジェクトを終えるというのはプロジェクト管理の知識領域であり、

プロジェクト成否のベンチマーキング<sup>\*12</sup>において、ソフトウェアのサイズに依存するが、失敗が多いということはエンジニアリングになりきれていないということであろう。Standish 報告書によれば、過去 10 年近いベンチマーキング数値におけるプロジェクト成功の比率は 10% 程度しか向上していない。SEC はその 2004 年の創設以来「定量的ソフトウェア開発」を主張しており、ソフトウェア開発データ白書を毎年のように発行している。定量的開発そのものへの関心及び取り組みが進んできたということは観測できるが、まだ十分ではないことが窺える。



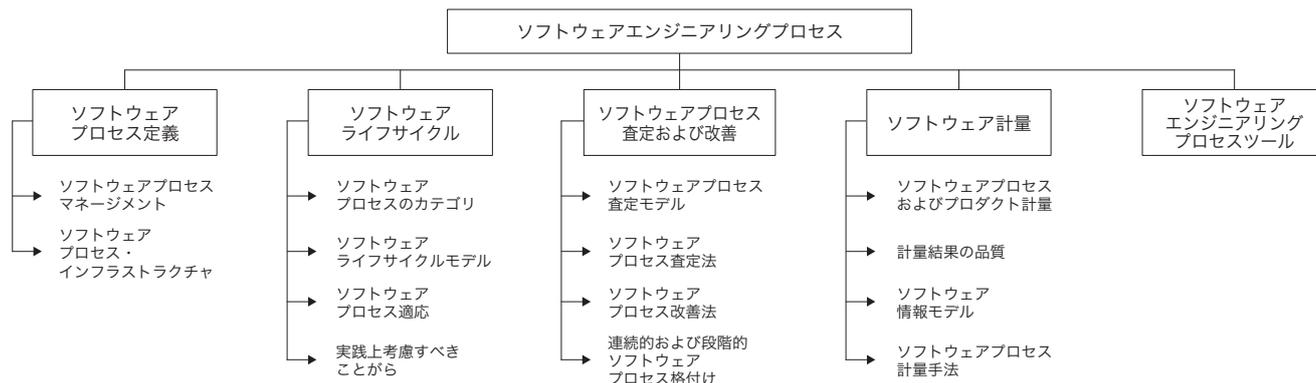
【脚注】

\*12 <http://www.standishgroup.com>、Standish グループ

## 9. 第8章 ソフトウェアエンジニアリングプロセス

SECの2004年の創設以来継続しているものに、かつ残念なことに関係する作業部会は共に数年前に終了しているが、共通フレームワークのISOとの同期及び日本独自のプロセスの補強とプロセス改善の推進がある。前者は、共通フレーム<sup>\*13</sup>を開発・維持し、後者はSPEAK-IPA版<sup>\*14</sup>を開発・維持してきた。前者のベースとなるものはISO/IEC/IEEE 12207であり、後者はISO/IEC 15504である。国際

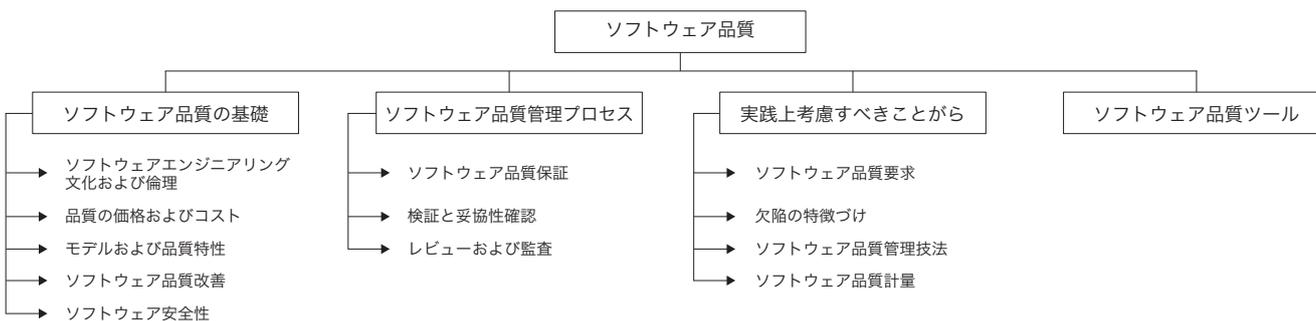
規格の開発にあたり、12207は2016年版(予定)が開発されつつあり、15504はISO/IEC 33Kシリーズと名称も変わり、そのコア規格が今年になりIS文書として発行されている。SPEAK-IPAがベースとするものも2016年版(予定)が開発されつつある。更に、IoTの時代になり、ソフトウェアのみからシステムの時代に移行しつつあり、対応する国際規格はISO/IEC/IEEE 15288となっており、つい最近国際規格の最新版が発行された。開発プロセスは関連するステークホルダが共通の言語を話すように定義されており、グローバルにビジネスが進めば進むほど最新のプロセスに準拠する必要がある。ここ数年の動きに注目する必要がある。



## 10. 第10章 ソフトウェア品質

ソフトウェアがその社会的位置を高めている現在、品質には留意が必要であろう。SECジャーナルに「情報システムの障害状況」が連載されている。これは、メディアに発表されたものを分析しているものであるが、ソフトウェア品質が十分高く問題はない、という認識であってはならないということを示しているように考える。とくにSWEBOK

V3.0の「ソフトウェア品質の基礎」における諸トピックスは今一度考察される必要があるだろう。この中でも「品質の価値及びコスト」については、従来ともすれば開発者の観点から品質を考察していたが、これからは、「顧客への引渡し後に発見されたソフトウェア問題に対処するアクティビティに付帯するコストを含む」に視点が必要であろう。前述のStandish報告は「価値」に関する必要であることを主張している。



## 11. 終わりに

3回の連載を通して、SWEBOK V3.0の技術的内容そのものは本体を読んでいただくことにして、ソフトウェア・エンジニアリングがカバーしている知識領域、副知識領域及びトピックスについては、オリジナルの発行者であるIEEE及びその訳本の発行者であるオーム社の好意により、すべ

ての知識領域について構造図を掲載した。知識領域を讀んでいただくに際して、留意事項を追記した。この連載が読書のみなさんがSWEBOK V3.0の日本語訳を讀む動機になったのであれば幸いです。

### 【脚注】

\*13 <https://www.ipa.go.jp/sec/publish/tn12-006.html>  
SEC BOOKS 共通フレーム 2013

\*14 [http://www.ipa.go.jp/sec/reports/20130326\\_2.html](http://www.ipa.go.jp/sec/reports/20130326_2.html)、SPEAK-IPA版

# 日本ファンクションポイントユーザ会 活動紹介



日本ファンクションポイントユーザ会 会長  
株式会社 NTT データ 技術開発本部 部長

藤貫 美佐

## 1 はじめに

日本ファンクションポイントユーザ会は「ソフトウェア  
定量化の推進を通じてソフトウェア市場の発展に貢献する」  
ことをミッションとして活動している。今年で設立 22 年と  
なり、FP 法（ファンクションポイント法）を始めとしたソ  
フトウェアメトリクスの普及推進に尽力してきた。

今年に入って、その成果を象徴する出来事が起きた。こ  
の 4 月から「政府情報システムの整備及び管理に関する標  
準ガイドライン」において、原則すべてのシステムでファン  
クションポイントを用いた見積りを行うよう規定された  
のである。本件の背景には、IPA/SEC（独立行政法人情報処  
理推進機構 / ソフトウェア高信頼化センター）様などが発  
行するファンクションポイントによるベンチマークの存在  
も大きく寄与しており、引き続き連携し市場の発展に貢献  
していきたいと考えている。

ソフトウェアメトリクスは非常に重要な分野であり、受  
発注の適正化並びに品質・生産性向上のためには、必要不  
可欠な要素である。また、見積りや品質・生産性管理にお  
いてユーザ企業とベンダ企業が共通認識を持つためには、  
同じメトリクスを使う事が重要である。そのために、我々  
はソフトウェアメトリクスの計測方法、ソフトウェアメト  
リクスの活用方法、業界での普及展開の 3 つを大きな柱と  
して活動している。

弊会の会員は約 150 会員であり、うち 9 割が法人会員、  
1 割が大学関係者等の個人会員である。法人会員はベンダ  
企業だけでなく、ユーザ企業も加入いただいております。研究  
会活動などでは異なる立場から意見を交換する場がよく見  
られる。このような活動の積み重ねから業界を変革してい  
く流れを作っていきたいと考えている。

## 2 組織の設立経緯と概要

ソフトウェアの規模計測のスタンダードとして FP 法を推

進するため 1986 年に IFPUG（International Function Point  
Users Group）が米国に設置された。

本会は日本における FP 法推進の拠点として、1994 年に  
IFPUG 日本会員連絡協議会として発足し、1996 年に日本  
ファンクションポイントユーザ会（JFPUG。IFPUG の正式  
な日本支部）となった。

本会の目的は会員相互の情報交換と親睦を図り、FP 法の  
理解やその利用法、更にはソフトウェアの定量化手法の検討  
を通し、国内への普及・展開を推進することである。

具体的な活動としては、定例会合にてソフトウェアメト  
リクス分野の著名人による講演や会員企業における FP 法  
の導入事例、大学や研究機関からの先端的情報、IFPUG や  
ISBSG の最新動向の紹介などを行っており、ほかでは得ら  
れない貴重な情報が得られる。

また、IFPUG が主催する FP 法のスペシャリストを認定す  
る CFPS（Certified Function Point Specialist）試験を実施し  
ている。

これらの活動を推進する組織として、表 1 に示す各種作  
業部会が設置されており、同じ課題を持つ会員が情報交換  
を行いながら課題の解決を目指して活動している。

## 3 FP 法とソフトウェア開発 のベンチマーク

ファンクションポイントはユーザに提供する入出力など  
の機能を定量化した値（機能規模）であり、基本的にはソ  
フトウェアの複雑さや品質に対する要件を反映しない値と  
して設計されている。

それ故、ファンクションポイントによる生産性のベンチ  
マークでは適用技術や業種、プロジェクトの特性による工  
数、欠陥数などへの影響が明らかになる点に特徴がある。

プロジェクトの定量的管理及び見積りなどの応用では、  
工数、コスト、欠陥数に対する影響を反映するため、日本においてファンクションポイントのベンチマー

表1 JFPUG 作業部会

計測技術委員会 (CPC)	FP 計測ルールの解説・補完
	計測上の疑問への回答 (カウンティングクリニック、会員用 Q&A)
教育検討委員会	ファンクションポイント計測コース (基礎編 / 演習編)、メトリクス活用コースの実施、研修資料の作成、指導
CFPS 試験実施委員会	CFPS (IFPUG 認定ファンクションポイントスペシャリスト) の情報交換、CFPS 国内試験開催、試験問題の翻訳
FP 活用研究会	FP 活用の課題の対策や定量データを利用する活動改善のための検討
	FP ベースのソフトウェア開発データの収集と結果分析
ビジネス活用研究会	ソフトウェアメトリクスのビジネス活用に関する研究
会員サービス委員会	FP 法及びソフトウェアメトリクス関連技術の普及と活動インフラの整備
	機関誌『JFPUG News』の発行と、JFPUG ホームページの企画・運営
国際委員会	FP 法及びソフトウェアメトリクス関連団体との連携・交流、情報交換
	国際諸団体 (IFPUG、ISBSG、COSMIC など) の最新動向の情報提供
JIS 原案作成委員会	FP 法及びプロジェクトベンチマーキングに関する JIS (日本工業規格) の原案作成
普及推進委員会	FP 法及びソフトウェアメトリクス関連技術の普及推進
COSMIC 検討作業部会	COSMIC 法の評価・普及 (組込み系、リアルタイム / 制御系)
	事例への適用、講習会の実施
CPM 翻訳作業部会	CPM (Counting Practices Manual) をはじめとする IFPUG 出版物の日本語翻訳と JFPUG 会員への提供

クを提供している IPA/SEC、一般財団法人経済調査会、一般社団法人日本情報システム・ユーザ協会の役割は大きく、本会の活動とも密接な関係がある。

## 4 活動の状況

### 4.1. 定例会合

JFPUG では、定期的に会員のみが参加できる会合を年 4 回開催している。

会合では、ソフトウェアメトリクス分野の著名人による講演、会員企業による FP 法の導入事例のほか、IFPUG や ISBSG の海外動向など、ソフトウェアメトリクスに関する最新情報の収集ができる。

また、会合後に開催している意見交換会 (立食パーティ) では、講演者や会員相互の意見や情報の交換も可能である。

参考) 2014 年度会合での各種講演

#### □ 2014 年 4 月総会

- ・「未来を作る新たなテクノロジーと DevOps 最前線」  
日本アイ・ビー・エム株式会社 榎原 彰 氏
- ・「ファンクションポイントで要件定義を制す」  
株式会社 NTT データ 藤貫 美佐 氏

#### □ 2014 年 7 月会合

- ・「アジャイルプロセスの見積り  
～ 変更分を意識した見積り事例の紹介 ～」  
株式会社日立ソリューションズ 英 繁雄 氏
- ・特別座談会  
「今求められる「超リスク管理」とは」  
株式会社日立製作所 初田 賢司 氏  
日本アイ・ビー・エム株式会社 榎原 彰 氏  
株式会社日経 BP 池上 俊也 氏

#### □ 2014 年 10 月会合

10 月会合は、特別イベントとして ISBSG の IT Confidence 2014 と共同開催した。各国からの発表と共に、以下の基調講演を実施。

- ・「エンタープライズソフトウェアの生産性に影響を与える要因の分析について」  
東海大学 古山 恒夫 氏
- ・「Why Can't People Estimate: Estimation Bias and Strategic Mis-Estimation」  
CEO of Galorath Incorporated D. Galorath

#### □ 2015 年 1 月会合

- ・「これが日本のトップアーキテクト！  
～持つべき五つの心構え～」  
株式会社日経 BP 島津 忠承 氏
- ・事例発表「第三者見積チェック」のご紹介  
日本ユニシス株式会社 大塚 和隆 氏

### 4.2. 研究推進活動

JFPUG はソフトウェアの規模計測を推進する団体であるが、計測値は現場及び経営者を支援する指標として活用された時にその真価を発揮する。そのため、規模計測値の活用方法も重要な課題であり、本会では以下に示す 3 つの活動を行っている。

- ・ FP 活用研究会
- ・ ビジネス活用研究会
- ・ 活用推進活動 (会員向け、非会員向け)

#### ① FP 活用研究会

FP 活用の課題に対する解決策を探求し、定量データを利用する様々な活動の改善につなげることを目的としている。検討したノウハウを形式知化し、JFPUG の情報資産とし

て蓄積していくことで活動成果とし、活動成果については、適宜会員と共有するよう推進している。

IT プロジェクトパフォーマンスベンチマーキングのフレームを軸に、総合的品質管理などの組織的FP活用と、見積りや品質管理などのプロジェクトにおけるFP活用の切り口から、課題や原因、対策などを検討している。

ユーザ企業、ベンダ企業の枠を超え、双方の視点の違いの認識や、社内の取り組み、その考え方などを共有しながら、課題解決の糸口を検討している。

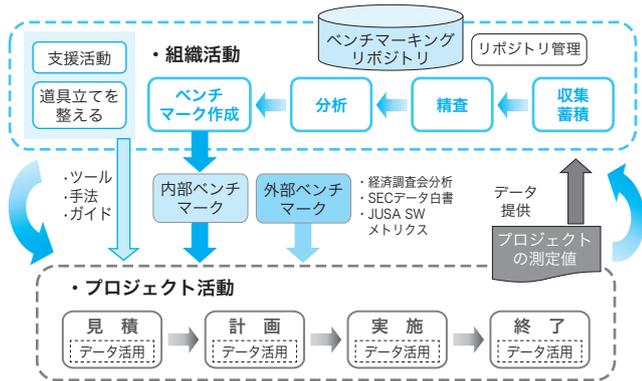


図1 FP活用研究会活動

## ② ビジネス活用研究会

ソフトウェアの価格評価や受発注の適正化を研究することを目的としている。

大目標を「ソフトウェア開発取引の指標の確立」に置き、以下のステップで活動を進行中である。

- ・ソフトウェア取引の定義と分類  
システムの利用分野、委託契約形態、開発タイプといった軸で分類し、それぞれの特性を研究する。
- ・価格指標の策定  
上記分類に対応して、価格の決定要因を検討し、有効な価格指標のアイデアを研究する。
- ・データの収集とベンチマーク  
上記の分類、価格指標に基づいて、主に会員企業からデータを収集し、ベンチマークを行う。

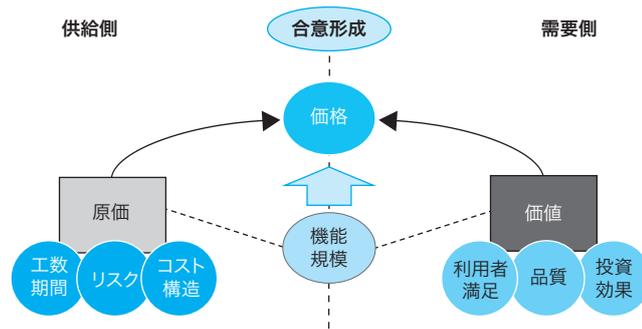


図2 価格相場形成のモデル

価格の決定要因には、発注側の価値と受注側の原価という二つの側面があり、ユーザ企業、ITベンダ、コンサルタント、調査統計会社など幅広く、立場を問わない意見交流を行っている。

## ③ 活用推進活動（会員向け、非会員向け）

FP活用研究会のメンバを中心に JFPUG 会員や一般の非会員に向けて、以下の活動を行っている。

- ・会員向けオープンワークショップ
- ・IPA との共催イベントの開催
- ・IT コーディネーター協会主催カンファレンスへの出席

研究会メンバ以外の多くの方々と意見交換を行うことで新たな課題やその対策のための取り組み、工夫などを共有している。



図3 ワークショップ活動風景

## 4.3. 教育・試験

会員へのFP計測法の普及のために1997年にスタートし、以後18年間の間に240回開催、受講者は延べ7,000名（2014年12月時点）となっている。現在のコースは、以下の3種類である。

- ・ファンクションポイント計測コース（基礎編）  
ファンクションポイントの成り立ちや概念、計測方法、活用方法などを説明する講義形式のセミナー。
- ・ファンクションポイント計測コース（演習編）  
計測方法をより深く理解するため、仮想システムの事例を元に計測の演習を行う。
- ・メトリクス活用コース  
ソフトウェアメトリクス活用の方法論や事例を元にしたケーススタディなど、講師が実体験を交えながら対話的に進める。今年度は「実践アナリティクス」「工数見積におけるFP活用」の2テーマを予定している。

講師陣にはFP計測やメトリクス導入推進における先進的な経験者をそろえており、会員企業の計測担当者が、フェイ

ス・ツー・フェイスで質問ができる場として活用されている。

普及に向けたもう一つの活動は、FP 計測スペシャリスト試験（CFPS）の国内開催である。

CFPS（Certified Function Point Specialist）は、FP 法の推進母体である IFPUG が認定する国際資格であり、合格には3時間150問に対し、各章9割以上の正答という計測精度が要求される。かつ3年毎に更新が必要であり、これを維持する者の計測技術の高さを裏付ける。JFPUG は、2001年度から日本語によるCFPS試験を12回開催し、延べ73名の合格者を輩出している。

公的機関でのFP活用が進む中、今後注目される資格である。

また、CFPSを中心として、ファンクションポイントの計測を正しく行うための活動も行われている。具体的には様々な局面においてファンクションポイントの適用指針をまと



図4 教育研修風景

めたガイドラインや、会員の直面している疑問や悩みに答える場の提供を行っている。

## 5 おわりに

JFPUG のミッションにある通り、ソフトウェア市場の発展に貢献すべく様々な活動を行ってきたが、まだ道半ばであると感じている。

その一つが業界への情報発信が不十分なことである。そのため、今まで会員内に閉じていた情報発信の幅を広げ、非会員向けのオープンセミナーなどを開催し多くの方に活動内容を知ってもらう機会を設けることにした。また、今までは首都圏を中心とした活動であったが、今後は地方における普及推進活動にも注力していきたい。興味のある方は是非参加いただきたい。

加えて強化していきたいのが受発注の適正化に向けた相場観の醸成である。システムが提供する機能を定量化するFP法をベースに価格の相場観が形成されれば、流した汗ではなく、価値で価格を判断することが可能となる。そしてこのことは業界の発展のためには必要不可欠な事項だと考えている。

価値のある成果を会員の方々と作り上げていきたいと考えているので、是非皆さんにも参画していただきたい。

年会費 法人会員3万円、個人会員1万円  
<http://www.jFPug.gr.jp/app-def/S-102/wp/>

### ファンクションポイントとは？

ファンクションポイントとは、ソフトウェアの機能規模尺度である。

あらゆる製品や資源には、その大きさを表す単位があり、取引のおおよその指標となっている。土地や建物であれば坪数、自動車なら排気量、テレビなら画面サイズといった単位があり、その規模に応じて価格相場が形成されている。

ところがソフトウェアについては、その非可視性から規模の捉え方が難しく、人月やソースコード行数といった単位が指標として使われてきた。

人月は受託した業務に要するリソースの量であり、成果物の大きさを表すものではない。いわば発注元に対して必要経費を請求するようなものである。ソースコード行数は、ソフトウェア開発技術が高度化し、フレームワークなどが普及した結果、昨今では測定自体が困難となっている。

その点、ファンクションポイントは、ソフトウェアで扱う①情報や、画面や帳票を通じた②情報の出し入れを解析し、ルールに基づいて計測するため、技術要素に左右されない客観的な数値を得ることができる。また多少の訓練を積めば、専門家ではない利用者でも計測が可能である。

また、こうした尺度については、メートル法のように世の中に広く普及し、認知されていることが重要である。

FP法は米国のIFPUGを中心に世界各国の支部や関連団体が推進し、国際標準化機構（ISO）の規格ともなっている。現在のところ、最も普及したソフトウェア規模指標であると言える。

我が国においてもFP法の知名度は高いが、実際には、見積時の参考値や、生産性や品質などの社内指標として使われる例が多い。

しかし、先般、公共入札時の提示要件のひとつとして推奨されるなど、これからは取引指標としての普及も期待されるところである。

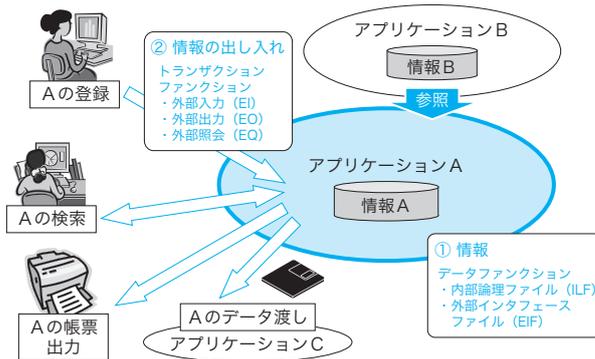


図5 ファンクションポイントの計測方法

# イノベーション —自分の頭で考え、行動する—

IPA 顧問、学校法人・専門学校 HAL 東京 校長  
鶴保 征城

オーストリアの経済学者ヨーゼフ・シュンペーターは1912年に、著書『経済発展の理論』の中で、「経済活動において旧方式から飛躍して新方式を導入する」ことを、「イノベーション」と名付けている。あるいは、「生産手段の新結合」とも呼んでいるように、技術に限らず、生産方法、材料調達手段、販路、組織のあり方など生産手段の組み合わせを意図している。

更に、ピーター・ドラッカーも、「イノベーションは天才によるひらめきではない (Innovation is not flash of genius.)」と言っている。

日本語では「技術革新」と訳されることが多いが、これは誤解を招く表現だ。かつての日本の製造業が世界を席巻した名残もあって、「アジアメーカに首位の座を奪われているが、技術は負けていない」とか、「企業内研究者の技術力が足りないから、ヒット商品の開発が進まない」などの意見が聞かれる。

これらの言葉に共通しているのは、技術そのものを鍛え世界のトップになれば、それだけで製造業というビジネスの競争力を強化することができる、という考え方だ。

一方、2015年7月に来日したGEのイメルト CEO は、「GEはメーカなのか IT 企業なのか」と聞かれて、「接続産業企業 (connected industrial company) だ。リアルとデジタルの交差点に立ち、デジタル化と同時に、製造業を更に進化させ、新たな時代で勝利する。正に偉大な産業企業へと変革している最中だ」と答えている。

GEはITやインターネット技術のインパクトを理解しつつ、単に商品やサービスの高度化を目指すのではなく、企業やビジネスそのものを変革、イノベートしようとしている。

イノベーションは製造業の技術や製品だけでなく、サービス業においても非常に重要だ。例としては、QB

HOUSE がわかりやすい。駅中やショッピングモールでよく目にする散髪屋だが、「10分の身だしなみ」というキャッチフレーズの通り、10分1000円で整髪を行う。QB HOUSEは髪を切るという単一サービスに徹しており、シャンプーや顔そりは提供しないから、スタッフはヘアカットのみに集中することができる。

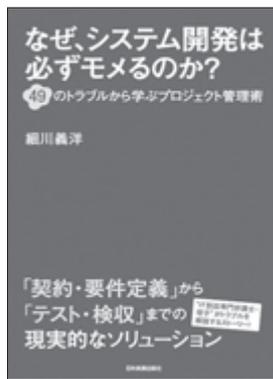
創業者は、以前高級理容室を利用した際、髪を切るだけで半日もかかることに疑問を感じ、理容室に本当に求められているサービスに絞って、現在のビジネスモデルを構築した。

QB HOUSEの例は非常に分かりやすいのだが、同じようにユニクロも全く新しい業態というわけではなく、散髪屋や洋服屋のビジネスモデルを刷新したということだ。世の中には短時間で前と同じ髪型にして欲しいという人もいれば、服を選んでいるときに店員に話しかけられたくないという人もいる。そのニーズを掴めるかどうかだ。

こうした逆転の発想を思いつき事業化することは、今後あらゆる企業で必要となるのだが、ここで重要なのが企業体質だ。

企業によっては、部長も課長も上を見て仕事をし、自ら考えていないことが多い。上から指示された仕事を鵜呑みにしてひたすらその範囲でやり遂げる、ということにほとんどすべてのエネルギーを使っている。

計画作成に時間を使うだけでなく、それをどういうビジョン、戦略で実行するのか、どうやって具体的に現場のアクションに落とし、経営改革を進めるのかについて計画作成に倍する時間を使わなければならない。要は、「自分の頭で考えて、積極的に発言し、行動する」力を身につけるとのことだが、これがこれからの厳しいビジネス環境に打ち勝つために欠かせない。



## なぜ、システム開発は必ずモメるのか？

49のトラブルから学ぶ  
プロジェクト管理術

細川義洋 著

ISBN: 978-4534051158

日本実業出版社刊

A5判・278頁

定価 2,000円 (税抜)

2013年9月刊

## 失敗を繰り返さないように学習しましょう

本書はソフトウェア開発現場での経験を有し、かつ、東京地方裁判所、東京高等裁判所のIT専門委員として多くの裁判事例にかかわってきた著者の経験に基づいて書かれている。

いろんな統計があり、そのコンテキストと共に語らなければ正確に現状を表すことはできないかと思うが、中規模サイズのソフトウェア開発プロジェクトの約7割が失敗するという産業界は異常であり、それにかかわるエンジニア及び管理者は本当にプロフェッショナルな仕事をしているのかと考える必要がある。更に、その失敗に対していかなる社会的、経済的判断が下されるのかも考える必要があろう。本書の読み方には種々あると思うが、今一度トラブルの予防策を考え、同様の失敗を繰り返さないように学習するには格好のものとする。

「はじめに」において、「オーダーメイドのシステム開発契約では、ベンダのみでシステムを完成させることはできない」という裁判官の実際の発言から始まる本書は、ともしれば堅くなりがちな裁判所における想定やりとりを軽妙な会話の中に取り入れることにより読みやすいものになっている。ただ、語り口が自分の環境と合わない読者は違和感を覚えるかもしれない。

開発ライフサイクルに従い、各ステージで押さえておくことが期待される事項についてのチェックリストが準備されており、これは参考になるであろう。また、本書の各所で裁判所がどう判断したかという判例が紹介されており、プロジェクト管理上の留意点として非常に参考になるであろう。

(新谷 勝利)

## システム科学技術で日本を再興しよう！

かつてNHKで日本の電子技術を謳歌した電子立国日本の特集やプロジェクトXが家庭や職場で話題にされた。その後、国際競争力は大きく低下した。今、日本再興に向けた成長戦略が、実施・展開されている。

本書は、「ものづくり敗戦『匠の呪縛』が日本を衰退させる」の著者、東大名誉教授木村英紀氏編著による技術思考の解説書であり、技術立国ニッポンの復活の鍵として“システム科学技術”の重要性が主張されている。

本書の要旨は主に下記である。

- ・ システム技術の事例紹介：電力系統システム、ボーイング777、KOMTRAX、ハイブリッド車、ロボット技術、交通チケットシステムなどの事例を通じたシステム技術の解説。
- ・ システム科学・技術とシステム構築：システム構築の強力なツール

であるITとシステム化は別物であり、システム化の能力とソフトウェアの開発力の双方がうまく機能することで時代のドライビング・フォースになることの解説。

- ・ 技術政策と日本の課題：巨人IBMの変身を例に挙げ、“ものづくり基本法”が、要素技術に注力しすぎていることや、システム技術をベースにした技術社会システムの構築の必要性。

文豪、夏目漱石が追及したテーマは“則天去私（天二則り、私ヲ去ル）”と言われている。世界では、インダストリー4.0などの動きが活発である。戦後経済を支えたものづくりへの思い込み（“私”）を捨て去り、時代が求める“技術思考”に脱皮すべき時期に来ている。本書はそのような思いを抱かせる“技術思考”への入門書でもある。

(久保 忠伴)



## 世界を動かす技術思考 要素からシステムへ

木村英紀 編著

ISBN: 978-4062579186

講談社（ブルーバックス）刊

新書判・206頁

定価 860円 (税抜)

2015年5月20日刊

## 編集後記

今回の「所長対談」は株式会社デンソー 技監 村山様にお話を伺いました。クルマ×IT で皆様は何を思い浮かべますか。コネクテッドカー、自動運転、カーナビ、UBER、CarPlay や OAA など、車という身近な存在だけに、様々な技術やサービスがあげられると思います。これらの中にはIoT や CPS の進展により初めて実現できるようになったものもあります。しかし、既に CPS は車の中やパワートレイン系といった、更に小さな世界では以前から行われてきたことが村山様のお話から理解できました。「つながる世界」を考えていく上で、車載システムは先行事例として、そこから学ぶことが多くあると感じました。

本誌が皆様のお手元に届くころから IPA/SEC のメンバは各種イベントで動き回っています。CEATEC JAPAN 2015 (10/7-10 幕張メッセ)、「ソフトウェア品質向上のためのコーディング技法と標準」セミナー (10/14 名古屋国際会議場)、IPA シンポジウム 2015 (10/21 東京ミッドタウン)、Embedded Technology 2015 / IoT Technology 2015 (11/18-20 パシフィコ横浜) など、皆様もぜひ会場へお越しください。 (編集長)

## 編集部より

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

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

## SEC journal 編集委員会

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



元気に泳ぐ金魚

(撮影 : k.hasegawa)

SEC journal 第 11 巻第 2 号 (通算 45 号) 2015 年 9 月 1 日発行

© 独立行政法人情報処理推進機構 2015

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

※本誌は「著作権法」によって、著作権等の権利が保護されている著作物です。

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

# SEC journal 論文募集

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

## 論文テーマ

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

## 論文分野

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

## 応募要項

締切り：1月・4月・7月・11月 各月末日

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

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

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

## SEC journal 論文賞

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

## ITパスポート試験のご案内

### ー ビジネスにITを活用する すべての社会人のための「国家試験」ー

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

## iパスのメリット

### ITを活用した業務効率化とビジネス拡大に！

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

### 情報セキュリティ対策・コンプライアンス強化に！

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

### 経営全般に関する知識など幅広い知識がバランスよく習得できる！

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

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

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

# IPA Better Life with IT

SEC Journal No.42  
第 11 卷第 2 号 (通卷 45 号)  
2015 年 9 月 1 日発行

© 独立行政法人情報処理推進機構

ISSN 1349-8622

