

SEC[®]

journal

Software Engineering Center

20

巻頭言

浜口 友一 社団法人 情報サービス産業協会 会長

所長対談

これからの企業のIT活用の方向と CIOの役割を考える

石黒 不二代 ネットイヤーグループ株式会社 代表取締役社長 兼 CEO

技術解説

共通フレーム2007 第2版 解説

システム開発プロジェクトにおける意思決定実態調査
— 調査結果から導く「意思決定判断モデル」 —

活用例

ソフトウェア開発データ白書の効果的な活用法

SEC journal 論文賞受賞論文

流用開発を意識したソフトウェア実装品質向上手法

山口 有紀, 田中 尉氏, 市川 裕章

実証実験報告

ITプロジェクトを「見える化」するチェックシート

SEC成果活用事例紹介

オリンパスソフトウェアテクノロジー株式会社

ETSSシステムを独自に開発。部内の人材を「見える化」

株式会社日立情報制御ソリューションズ

事業と技術の特徴に合わせて3つのスキル標準を活用

SEC journal論文のまとめ方

アングル

経営者はアーキテクチャと形式手法を忘れてはいけない

組織紹介

一般社団法人組込みスキルマネジメント協会

九州地域組込みシステム協議会

Column

最後のチャンス!! 2010年代

IPA[®]

独立行政法人 情報処理推進機構

<http://www.ipa.go.jp/>



SEC journal No.20
Contents

- 1 巻頭言
浜口 友一 社団法人 情報サービス産業協会 会長
- 2 所長対談：石黒 不二代 ネットイヤーグループ株式会社 代表取締役社長 兼 CEO
これからの企業のIT活用の方向とCIOの役割を考える
- 6 技術解説
共通フレーム2007 第2版 解説
長谷部 武
- 13 システム開発プロジェクトにおける意思決定実態調査
—調査結果から導く「意思決定判断モデル」—
中村 宏美
- 20 活用例
ソフトウェア開発データ白書の効果的な活用法
小椋 隆
- 24 SEC journal論文賞 受賞論文
流用開発を意識したソフトウェア実装品質向上手法
山口 有紀
田中 尉氏
市川 裕章
- 32 実証実験報告
ITプロジェクトを「見える化」するチェックシート
樋口 登
- 36 SEC成果活用事例紹介
オリンパスソフトウェアテクノロジー株式会社
ETSSシステムを独自に開発。部内の人材を「見える化」
株式会社日立情報制御ソリューションズ
- 38 事業と技術の特徴に合わせて3つのスキル標準を活用
- 40 SEC journal論文のまとめ方
平山 雅之
- 46 アングル
経営者はアーキテクチャと形式手法を忘れてはいけない
藤枝 純教 高信頼性システム技術作業部会委員 グローバル情報社会研究所株式会社 代表取締役社長
- 50 組織紹介
一般社団法人組込みスキルマネジメント協会
—ETSSによるスキルマネジメントの体系化による人材育成を目指して—
大原 茂之 一般社団法人組込みスキルマネジメント協会 理事長
- 52 九州地域組込みシステム協議会
芦原 秀一 特定非営利活動法人九州組込みソフトウェアコンソーシアム 副理事長
(株式会社ネットワーク応用技術研究所 取締役)
犬塚 智彦 財団法人九州先端科学技術研究所 研究企画部
- 54 Column
最後のチャンス!! 2010年代
鶴保 征城 IPA顧問 学校法人・専門学校HAL東京 校長
- 55 BOOK REVIEW
- 56 編集後記
お知らせ (論文募集 / SEC journalバックナンバー)

ITサービスの信頼性向上を目指して



社団法人 情報サービス産業協会 会長

浜口 友一

ソフトウェア・エンジニアリング・センターの役割

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（IPA/SEC）は2004年に設立されて以来、品質の良いソフトウェアをいかに効率的に開発するか、という課題に一貫して挑戦してこられました。

エンタープライズ系と組込み系の両分野において、産官学の志ある技術者や研究者たちが集い、実践を通じて生み出された数々の知恵を体系化し、新たな英知を創出する活動が行われ、我々情報サービス事業者にとって価値の高い成果が多く生み出されてきました。

JISAの情報システムの信頼性向上への取り組み

2007年3月には、当協会（社団法人 情報サービス産業協会（JISA））とIPA/SECとの間で「JISA-IPA/SECの連携体制の構築について」を策定し、JISAに新設した特命委員会において、IPA/SECの支援を頂きながら、情報システムの信頼性向上に取り組んできました。

例えば、経済産業省が策定された情報システムの信頼性向上ガイドライン及び信頼性評価指標が「何をすべきか（What）」を示していることに対し、JISAの活動では、会員企業が実際の開発、保守、運用の現場で用いる信頼性確保の管理指標を集約・分析し、「何を測定し、どう管理・活用すればよいか（How）」「なぜそうするのか（Why）」を提示し

た「信頼性向上のベストプラクティスを実現する管理指標調査報告書」を取りまとめました。

加えて、契約書や仕様書においてあいまいになりがちな変更量や検収条件等について、ユーザ企業とベンダ企業の双方が合意しやすい定量的な管理指標を調査し、業界内外に提示する活動を行ってきました。

JISA-IPA/SECの連携強化

情報化の進展はとどまるところを知らず、情報システムに対するニーズが一層多様化・複雑化する中、情報サービス事業者にも経営の高度化やグローバル展開への対応が強く求められるようになっていきます。

また、インターネットの普及に加え、SOA（Service Oriented Architecture）、クラウド・コンピューティングといった新しい技術やアーキテクチャの出現に伴い、ユーザ企業のニーズは自社専用で開発した情報システムを保有して運用するという従来の考えから大きく変わりつつあり、情報サービス産業も従来のビジネスモデルからの転換に真剣に着手すべき時代に入ったと認識しています。

情報サービス産業協会は、経済・社会の情報インフラを担ってきた産業を代表する組織として、ITサービスを安定的かつ効率的に提供する使命を果たすべく、今後とも、IPA/SECとの連携を一層深めていく所存です。

これからの企業のIT活用の方向と CIOの役割を考える

ITの活用が業務の効率化から企業戦略の実現サポートへと変わってきている中、CIOの役割の重要性が増している。CIOにとって重要な能力や今後のIT活用策について、Webマーケティングの分野で豊富なノウハウと実績を持つネットイヤーグループ代表取締役社長 兼 CEOの石黒不二代氏に伺った。

ネットイヤーグループ株式会社 代表取締役社長 兼 CEO

石黒 不二代



SEC 所長

松田 晃一

松田 石黒さんは『日経ウーマン』誌ウーマン・オブ・ザ・イヤー 2009 の3部門のうちのリーダー部門で1位、総合2位を受賞されていますね。おめでとうございます。女性として経営の最前線でご活躍されて、これまで色々ご苦労があったことでしょうか。

石黒 よくそのような声をかけられます。でも、私には苦労という感覚はありませんでした。ずっと、自分がやりたいことをやればいいと考え行動してきましたし、また女性の社会進出は難しいと思ってしまっていたら、会社経営に踏み出してはいなかったと思います。

松田 今は男女雇用機会均等法がありますが、石黒さんが大学を卒業された1980年初頭はどのような状況でしたか。

石黒 私は経済学部を卒業したのですが、その時期は男女雇用機会均等法が無いどころか、そもそも女性に対する仕事のオファーが全く無い時代でした。在籍していた経済学部の学生は210人、そのうち女性は10人でしたが、大学の就職課への求人は男性のみで、学部には女性がいること自体が認識されていません。そこで、大学卒業後の1年間、就職をせずにアルバイトをし、チャンスをうかがっていました。卒業の翌年から、本当にごく一部の大企業が経済学部卒の女性の採用を開始し、私は地元のブラザー工業に中途で採用されたのです。

松田 そこではどんなキャリアを積まれたのですか。

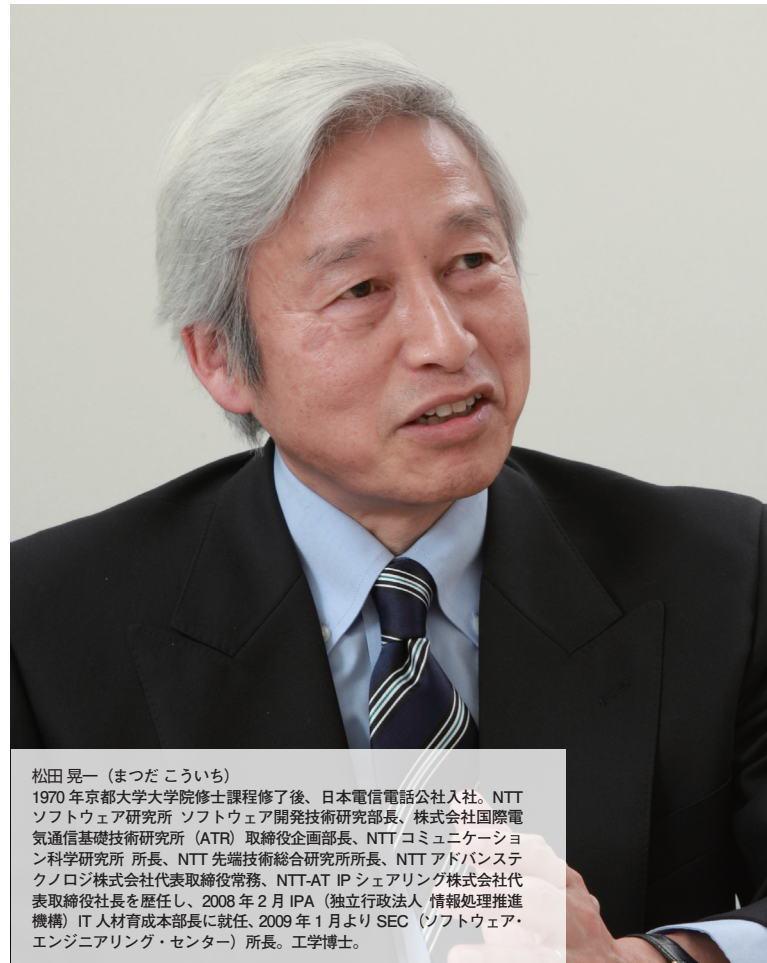
石黒 製品の90%が海外向けで、私は海外向けの商品企画と営業を兼ねたプロダクトマーケティングを担当し、ユーザーニーズをエンジニアへ説明し製品の仕様を決めるといった仕事に6年間取り組みました。

松田 そのような大事な仕事を任されて、女性にとっても、かなり働きやすい会社だったようですね。

石黒 海外向けの製品がほとんどなので、上司は「アメリカで10年仕事をしていました」という人ばかりでした。女性が仕事することに理解の高い会社だったことが私にとって大変にラッキーでした。同社の女性として、初めてヨーロッパに長期出張を命じられる等、いろいろな仕事を経験しました。

松田 アメリカへはいつ行かれたのですか。

石黒 ブラザー工業には6年間勤めた後、新たなキャリア獲得を目指し、ある外資系企業の新規部門のマネージャを



松田 晃一 (まつだ こういち)

1970年京都大学大学院修士課程修了後、日本電信電話公社入社。NTTソフトウェア研究所 ソフトウェア開発技術研究部長、株式会社国際電気通信基礎技術研究所 (ATR) 取締役企画部長、NTTコミュニケーション科学研究所 所長、NTT先端技術総合研究所所長、NTTアドバンステクノロジ株式会社代表取締役常務、NTT-AT IPシェアリング株式会社代表取締役社長を歴任し、2008年2月IPA (独立行政法人 情報処理推進機構) IT人材育成本部長に就任、2009年1月よりSEC (ソフトウェア・エンジニアリング・センター) 所長。工学博士。

5年間務めました。その間に結婚して子供が生まれたのですが、日本ではワーキングマザーとして生きていくのは難しいと考え、MBA取得に向けビジネススクールを受験し、第1志望のスタンフォード大学に合格し、アメリカへ渡ったのです。MBAを取得してからは、ハイテクに特化したコンサル会社をアメリカで立ち上げました。その会社の仕事をするうち、ネットイヤーグループをマネジメント・バイ・アウトという手法で独立させるため、私に資本及び経営参加してくれないかという話をいただいたのです。1999年、ネットイヤーグループを日本に設立し、2008年3月にはマザーズに上場させていただき、今日に至っています。

ものを作るための活動もマーケティングであり、宣伝や販売、営業等お客様と接点がある領域すべてでマーケティング活動が出来るのです。

松田 なるほど。かつてブラザー工業で経験された商品企画と、営業を兼ねたプロダクトマーケティングが原点にあるのですね。御社では Web マーケティングに取り組まれています。マーケティングにインターネットを活用する狙いはどんなところにあるのでしょうか。

石黒 テレビで宣伝する場合は、テレビ局が持つ枠を買いますが、テレビは他社が所有するメディアです。それに対して、インターネットは自社メディアであり、自由に使えるメディアです。お客様の持つインターネットという自社メディアを使って売上げを伸ばしたり、販売・宣伝活動をするための戦略・企画の立案と実際にそのための仕組みを作る制作業務、開発業務を行っています。

松田 つまり、マーケティング業務にインターネットを始めとしたITを活用して、企業の競争力を高めるための戦略立案を支援するということですね。

石黒 その通りです。

松田 石黒さんは、現在経済産業省の「産業構造審議会情報経済分科会」と「情報技術と経営戦略会議」の委員をされていますね。日本企業のIT活用をどのようにご覧になっていますか。

石黒 およそ20年くらい前のITシステムは、ホストコンピュータがあり、その上で基幹システムが動いていました。そのもっと前は磁気テープが主たる記憶媒体の時代でした。その頃のITは、まだ経営で利用されていたとは言い難く、90年代のクライアントサーバシステムの躍進以降、システムを利用することによる経営効率化へとITの活用目的が変わってきています。更に、コストを削減して経営の効率を向上させるだけでなく、もっとフロントのお客様向けにITを活用していく時代に入っています。経済産業省の「情報技術と経営戦略会議」では、インターネットで取得したデータを既存の顧客データベースや商品データベースと組み合わせることによって効率良くデータを解析し、その結果を基に営業がターゲット顧客をしっかりと掴んで訪問するというように、IT活用がバックエンドからフロントエンドに変わってきています、というお話しを申し上げます。

松田 ITの活用は、企業内のルーチン的オペレーションを機械化して効率化するということがスタートだったのですが、これからは新しいビジネスを作り出すことや、企業の競争力を高めることに活用していくべきだ、ということですね。

石黒 はい。まだその活用は、アメリカに比べて日本は不十分だと思います。

松田 そういう指摘はよく聞きます。日本企業のIT活用はバックオフィスの効率化が主体で、アメリカ企業は企業の競争力を高めること、企業の戦略を実現することにITを活用していると、よく耳にします。欧米企業のIT投資の半分は戦略的投資であるのに対して、日本企業では1/4程度で、3/4はシステムの維持運用等固定的支出だという



石黒不二代 (いしくろ ふじよ)
ネットイヤーグループ株式会社 代表取締役社長 兼 CEO。ブラザー工業、外資系企業を経て、スタンフォード大学にてMBA取得。シリコンバレーにてハイテク系コンサルティング会社を設立、日米間の技術移転等に従事。2000年よりネットイヤーグループ代表取締役として、大企業を中心に、事業の本質的な課題を解決するためウェブを中核に据えたマーケティングを支援し独自のブランドを確立。日経情報ストラテジー連載コラム「石黒不二代のCIOは眠れない」等著書や寄稿多数。経済産業省IT経営戦略会議委員に就任。URL: <http://www.netyear.net/>

ITの活用でホワイトカラーの生産性を向上させる

松田 ところで、ネットイヤーグループは、どのようなビジネスをされている会社なのですか。

石黒 ネットイヤーグループは法人のお客様を対象とした、インターネットを中心とする総合的なマーケティング支援企業です。マーケティングというと、日本ではテレビのコマーシャルのように宣伝に特化した活動と思われるがちですが、商品企画もマーケティングの範疇なのです。良い

調査があります。

石黒 私はビジネススクールで2年間学び、その後、アメリカ企業を相手にコンサルティングをしてきました。その経験を通して日米の企業を比較すると明らかに違いがあることに気づいたのです。日本企業の強さは製造部門の強さだと思います。製造部門の効率は、非常に高いものがあります。エンジニアも優秀です。アメリカ企業に比べて日本企業が劣っている職種はホワイトカラーだと思います。そしてホワイトカラー全員が劣っているというのではなく、優れた人が属的に知識や経験を所有していて、皆での共有がされていないことが問題なのです。

松田 なるほど。

石黒 アメリカのホワイトカラーはビジネスインテリジェンスツールをよく使っていて、ホワイトカラー全体の生産性を上げています。データの抽出は、ITの得意とするものです。もともとのデータは同じでも、抽出の仕方により、より有益な情報が得られます。日本企業の場合、勘のいい人が自分流にそれをしているのです。そこをIT化していけば、あらゆる業種業態にいる多くのホワイトカラーの仕事が効率化出来、日本企業はまだまだ成長するでしょう。また、既存業務の効率化の結果、ホワイトカラーは一段上の業務を行うことが出来るようになります。例えば銀行の窓口業務はほとんどが機械化され、インターネットを使って振込をすることが広がっています。そこで、銀行は、浮いた時間を使いコンシェルジュサービスや顧客の資産価値を上げるためのアドバイザーに力を入れようということになっています。

新しい技術の導入が必然的な米企業

松田 インターネットを使って業務をIT化することによって一段上の付加価値の高い仕事に移行していくことが、ホワイトカラーの生産性を上げることにつながるのですね。お話に出てきたビジネスインテリジェンスは、業務システムに蓄積された膨大なデータを、分析して企業の意思決定に活用する手法で、データマイニング等もそのための技術の1つですよね。このような価値のある情報の抜き出し方に関するノウハウは、アメリカでは相当に蓄積しているのでしょうか。

石黒 ケース・バイ・ケースの部分はありますが、基礎となるとはツールやプロダクトを使っています。日米の企業の大きな違いの1つは新しい技術を使うかどうかです。日本企業は新しい技術を使いたがらない傾向があります。アメリカ企業は新しい技術が出てくるととにかく使ってみるのです。日本では、やっとビジネスインテリジェンスツールを使ってみようという意見が出始めてきたのですが、アメリカ企業はもうみんな使っています。アメリカ企業には「これを使うと何か良くなるのではないか」という意識が強く、使うことへのためらいはほとんどないのですね。

松田 その意識の差はどこから出てくるのでしょうか。

石黒 おそらく、企業の人事制度の違いによるのではない

でしょうか。アメリカの企業の人事評価は業績連動型になっています。社員がもっとたくさん給料をもらうためには自分のパフォーマンスを改善しないといけない。そのため、ツールを使ったほうがよいというところへ、自然につながっていきます。

松田 新しい技術を取り入れて少しでも効率を良くすることが、自分の評価につながるという考えは日本には無いということですね。

石黒 そもそも、この会社の中でどれだけ長く生きていくかということではなく、自分自身の生産性を向上させるために何を使ったらよいのか、人を使ったらよいのか、技術を使ったらよいのか。これら使い方は、すべて自分の力になります。アメリカの社会は流動性が高いわけですが、そういう知識を持っていると他の職場でも使えるという、根本的な意識の違いがあるのではないかと思います。

松田 社員のそういう意識の違いが、新しい技術を取り入れる姿勢の違いになっているのですね。

CIOに求められるのは 基幹業務を熟知していること

松田 最近の新しい技術として日本でもクラウドコンピューティングが注目されています。

石黒 日本人は、今まで使っていたアプリケーションをクラウドにすると、コストが安くなるという、コスト的なものには強い関心があるのですが、日本企業の場合、セキュリティに対する意識が強く、クラウドに変えることに抵抗感がすごくありますね。それと、新しいものを使って失敗した場合を考えるカルチャーが、新規取り組みへのハードルになっていると思うのです。

松田 日本企業の中にITを含めて新しい技術をどんどん取り入れていくためにはCIOの役割が重要です。『日経情報ストラテジー』誌の連載でたくさんの日本企業のCIOと会われていますね。ご感想をお聞かせいただけますか。

石黒 日本を代表する企業のCIOと呼ばれる方とお話しています。これまでで40人くらいになります。大きい企業のITをすべて任されるCIOを置く会社は先進的なので、非常に勉強になります。

松田 アメリカのCIOは、24時間365日、経営とITを



つなげることばかり考えていますが、日本のCIOは他にも担当を持っていてCIOの仕事だけをしている人は少ないと聞いていますが、その辺りはどうでしょうか。

石黒 私がインタビューさせていただいているCIOの方々是一部上場企業の方がほとんどですが、他の業務を兼任されている方はあまり多くないですね。

松田 日本の場合、自社内で総務やシステムを担当してきた人がCIOになるケースが多いと伺っています。

石黒 私の認識ではアメリカ企業のCIOといえどもずっとITを担当してきた人ばかりというわけではありません。これまでたくさんの日本企業のCIOにお会いしましたが、文系の方が多く、文系と理系の比率は半々くらいです。その比率はアメリカのCIOも同じです。CIOとして大切なことは、その会社の基幹業務を経験していることだと思います。会社の実態をよく分かっていることと言ってもいいでしょう。

松田 なるほど。

石黒 会社ごとに一番大切な業務があるはずですが、具体的には、製造業であれば工場のオペレーションであり、流通であればフランチャイズです。そういう大切な業務をよく分かっていることが大切です。アメリカのCIOは、会社にとっての生命線を経験している人が多い。CIOの仕事の対象は技術ではなく、その上位概念の経営とITの橋渡し、もしくは経営です。会社を経営していくための一番上のコンセプトをどう作っていけばよいか分かる人、それがCIOではないでしょうか。CIOはあえて、技術系でコーディングが出来ますという人ではなくてもよいのです。

松田 同感です。その会社の代表的な基幹業務をよく知っているということは、CIOにとって重要です。

石黒 私流に言えば、経営とITの橋渡しの方向性が決められる能力があり、社内の人たちを動かせるということが大切なことだと思います。

アジャイルに通じる Webマーケティングの開発手法

松田 SECでは、システムを開発する側の立場で、ソフトウェアを中心としていかに効率良くものづくりを行うかについて盛んに議論をしてきました。最近では、企業戦略をITでどう実現するかという、開発の前段部分が大事だと考え、そちらのほうに活動を少しずつシフトしていこうとしているところですが、開発する側とCIOを中心とするユーザー側とがうまくコミュニケーションし、要求されるものを与えられた予算や時間内で作っていくためにはどうすればよいかを改めて考えないといけないと思っています。

石黒 大変正しい方向だと思います。ネットイヤーグループは、このケースでは開発する側の会社で、ユーザーが使うインターネットに関するシステムを中心としていますが、更に必要なバックオフィスのデータベースも開発し、提供しています。

松田 Webマーケティングのためのシステムを作られるわけですね。

石黒 そうです。

松田 システムが出来上がってから顧客との間でミスマッチが生じるということは起きませんか。

石黒 互いの同意がずれているときはミスマッチが起きますね。出来上がってから、これは違うということはありません。やはり、作るべきものの仕様をきちんと分かっている、きちんと契約しているかということが大切だと思います。最終的に何が出来るのかということをお客様にきちんと理解してもらう能力が必要ですね。

松田 SECも、顧客といかに上手なコミュニケーションを行い、意識合わせの精度を高めるかという部分に手を付け始めたところですが。

石黒 ネットイヤーグループの仕事は、3つの段階があります。1つ目は、ITだけではなくもっと上位のマーケティング戦略やCRM戦略を策定します。2つ目は、それをインターネットシステムの設計に落とし込みます。3つ目は、出来上がった設計を、ビジュアルを含めて技術に落とし込むというものです。3段階を、そのつなぎ部分を含めてすべて1社で行っています。実際に使うサイトをクリエイティブの人が描き、お客様から好評を得ても、実際のITシステムとして実現するにはハードルがある場合もあります。そこで、社内各担当者との意識合わせをするためにも、大人数のお客様との定例ミーティングを行っています。

松田 マーケティングという領域に関して上から下まで作って垂直統合的に提供する仕事ですね。

石黒 そうです。ネットイヤーグループは、以前はディレクター集団だったのです。コーディングやサイトを作ったあとのメンテナンスは外注していました。でも、今やインターネットは24時間365日、常に改善が求められるメディアになっています。そのため、メンテナンスの会社も持つ等、業務は上にも下にも拡大しています。

松田 お話をお伺いして、顧客が開発サイドと繰り返し議論して仕様を決めていくという、アジャイル開発に自然と辿りつかれたように思えます。

石黒 インターネットは、ユーザ参加型ですから基本的にアジャイルです。当社の顧客は法人で、法人の先には、エンドユーザがいます。エンドユーザが使ってみることでエラーが見つかるかもしれない。また使ってもらうことで改善点が見つかり、改良につながることもあります。一般ユーザが使い、その意見を聞きながら直していくというのはインターネットの基本精神と言えます。私どもは、この考えで仕事を進めています。それは、ソフトウェア開発におけるアジャイルという手法に近いのかもしれないですね。

松田 アジャイル手法は、ソフトウェア開発の1つの有力な手法としてSECでも注目をしています。アジャイルは、良いコミュニケーションを関係者で行うことが大事であり、突き詰めていくと、通常の業務の進め方と同じなのかもしれません。ユーザとベンダの両者が、良いコミュニケーションを通じてそれぞれの責任を果たすことが、良いシステム作りへの近道です。CIOの役割は、ますます大事になってきていると思います。

文：小林 秀雄 写真：越 昭三郎

共通フレーム2007 第2版 解説

SECエンタプライズ系プロジェクト 研究員
長谷部 武

IPA/SECは、2009年10月に「共通フレーム2007 第2版」を刊行した。
本稿では、第2版刊行の背景と狙い、主な改訂項目のポイントを解説する。
読者諸氏には、第2版とともに一読して頂ければ幸いです。

1. はじめに

SEC BOOKS「共通フレーム2007」を2年振りに改版し、その第2版を2009年10月1日に発行した。共通フレームとは、ソフトウェアの構想から開発、運用、保守、廃棄に至るまでのライフサイクルを通じて必要な作業項目、役割等を包括的に規定した共通の枠組みのことである。

図1に共通フレーム2007の位置づけを示す。共通フレーム2007は、日本のJIS X 0160-1996とJIS X 0160:2007の2つに基づきつつも、ISO/IEC 12207の翻訳規格であるJISでは定義されていない日本独自のプロセスを含み、またプロセスの解説や様々な補足情報等を含む書籍として利活用できる内容となっている。

表1に、各種観点から定量的に比較できる第1版（2007年10月1日発行）と第2版との差異を示す。

改訂の狙いは、大きく3つに分けられる。

- ・ガイダンス等の強化
- ・日本独自のプロセス内容の見直し
- ・原理原則17ヶ条の本質（3.4節参照）の訴求

第1点目は、第1版の第3部においてガイダンス内容の記述にバラツキがあるため、出来る限り、その内容の充実を図ることである。第2点目は、超上流と呼ばれる企画プ

表1 第1版と第2版との差異

【評価項目】	【第1版】	【第2版】
① プロセスの数	26	26 (うち名称変更: 1)
② アクティビティの数	123	123 (うち名称変更: 11)
③ タスクの数	456	472 (うち名称変更: 58、新規に追加: 16)
④ 図表の数	図: 56 表: 6	図: 60 (うち改訂: 41、削除: 2、新規作成: 7) 表: 7
⑤ 索引に掲載した用語数	448	244 (精選した数)
⑥ 書籍のページ数	320	344 (表紙、折込等を除く) (注) 320頁のうち改訂頁数: 283、88%相当。
⑦ 第3部で改訂した範囲	—	規定内容を改訂した範囲 (a) プロセスのうち、22個を改訂 (b) アクティビティのうち、15個を改訂 (c) タスクのうち、161個を改訂 (注) 新規追加のタスクはカウント対象外。

2. 改訂の背景と全体像

2.1 第1版改訂の背景

第1版発行後の1年間は、SEC主催セミナーや編著者各人による講演活動を通じてその普及・啓蒙に努めてきた。1年を経過後、編著者自らが認識した第1版の問題点を解決しようとの気運がプロセス共有化WG^{※1}内で湧き上がってきた。また第1版の読者から様々な質問や要望等がIPA/SECに届いていた。このような要請に応えるためにも、第1版の改訂に取り組むこととなった。その結果、出来上がったものが本稿で解説する第2版である。

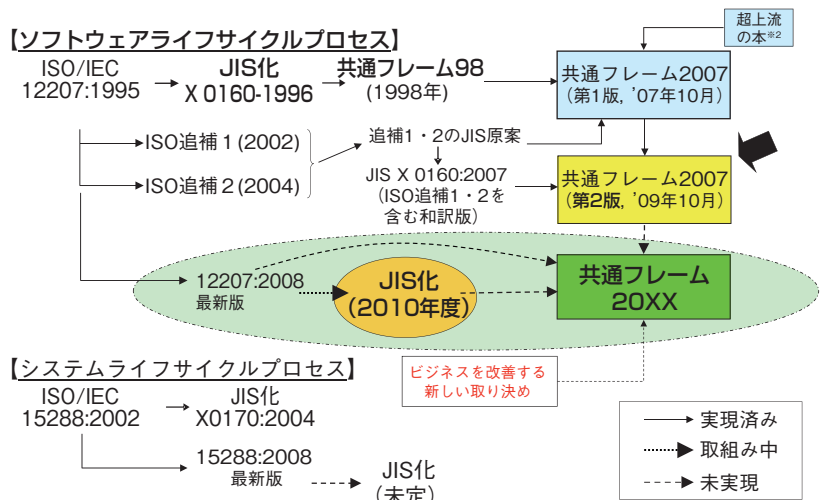


図1 共通フレーム2007 (第1版と第2版) の位置づけ

※1 第1版の編著者は、経済産業省の下で活動していた開発プロセス共有化部会 (委員) であったが、同部会が2008年4月からIPA/SECの下で活動することとなったため、第2版の編著者は、SECエンタプライズ系プロジェクト プロセス共有化WG (委員) となっている。
※2 SEC BOOKS「経営者が参画する要求品質の確保 (第2版)」を指す。

プロセス、要件定義プロセスを中心に各プロセスの内容の整合性を図ることである。第3点目は、SEC BOOKS「経営者が参画する要求品質の確保（第2版）」を踏まえ、その精神（スピリット）をメッセージとしてもっと明確にし、共通フレーム 2007 という書籍においても強くアピールしていくことにある。さらに、これら3点に加え、第1版における参照先の誤りや記述の重複等の訂正も行うことにある。

2.2 改訂の全体像

第1版から第2版へと、どのように改訂されたのかという全体像（改訂項目のすべて）は紙面の都合で省略するが、SEC ホームページ（<http://sec.ipa.go.jp/publish/index.html#ent>）から改訂情報（PDF ファイル）をダウンロードして知ることができる。

本稿では、筆者として特に重要と考えられる改訂項目を取り上げて解説する。

3. 重要な改訂ポイント

3.1 「ニーズ」と「要求」「要件」の概念整理と用語見直し

requirements という語の日本語訳は様々である。JIS Q 9001 では「要求事項」と訳され、JIS X 0160-1996 では「要求事項」又は「要求」と、JIS X 0160:2007 では「要件（規格が要求する事項）」又は「要求（利害関係者が要求する事項）」若しくは「要求事項（個別のものをいう場合）」と訳されている（JIS X 0160:2007 の p.53 参照）。

共通フレーム 2007（第1版）では、基本的には、用語が使われている文脈上で needs 又は wants の意味に近いところでは「要求」を、プロセスからのアウトプットとしては「要件」という用語を当てはめてきた。しかしながら、第1版を1年間読み込んできたプロセス共有化 WG

委員の中から、第1版（p.32）記載の説明文「システム化に対する要求と要件の使い分け」は少々分かりにくいという指摘が挙がった。これとは別に、国際標準審議の場でテーマに挙がっていた RE（Requirements Engineering）の規格化において、当初の米国案がシステム要件とソフトウェア要件の二つに限定されていたため、日本が超上流の重要性の観点から、経営層や業務層からの一連の流れを具体化すべきであると主張し、かつ 2008 年 5 月から needs / wants と requirements との位置づけを明確にしたモデルを提案した結果、これらの提案が受け入れられつつあった（図 2 参照）。これは、RE 規格案を担当する ISO/IEC SC7/WG7 の日本代表を兼ねるプロセス共有化 WG（村上憲稔主査）が、当該標準化委員としてプロセス共有化 WG 有志を国際標準審議の場に参画させ、規格のコエディタとして橋本恵二氏（東京国際大学）を推薦し国内審議を重ね、日米共同提案へと持ち込んだ成果である。このような一連の諸活動を通じて、そのモデル検討の成果を「ニーズ」及び「要求」「要件」の概念整理に反映することとなった。

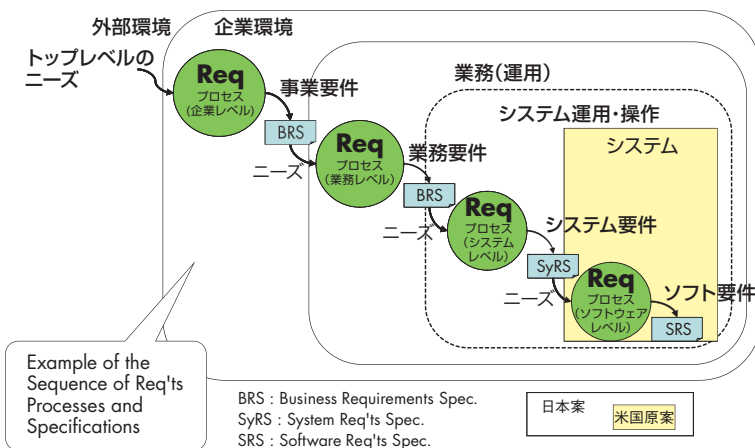
このような経緯の下で、プロセス共有化 WG として最終的に合意した事項を以下に示す。

- ①「要求」「要件」は、ともに requirements の訳語である。
- ②「要求」を一般的な訳語とする。ただし、「要件」は要求プロセスのアウトプットに対して、特に「固まった文書」を強調したいときに使う。
- ③「要求」「要件」以前のものを「ニーズ (needs)」という。期待 (expectation)、要望 (desire/wants)、思い (wish)、夢 (dream)、意図 (intension) 等は、ニーズの類義語である。

図 2 の見方であるが、「Req プロセス（要求プロセス）」へのインプットとして「ニーズ」があり、そのアウトプットとして「要件」がある。要件が階層構造になっていることにも留意されたい（事業要件⇨業務要件⇨システム要件⇨ソフトウェア要件である）。また、ある要件が後続のプロセスのインプットとなるときには、ニーズとなっていることも読み取って頂きたい。

なお、「要求プロセス」という名称は、固有のプロセス名称を示したのではなく、一般的な説明として「要求分析及び／又は要件定義」の機能をもつプロセスを示すものとして表現している。

この概念整理に基づいて、第2版では、共通フレーム全般にわたって「要求」「要件」「要求事項」「要求仕様」等の表示箇所を見直して適切な表現に変更している。また、「要求」「要件」の使い分けに関する説明文も書き直している（第2版 p.34 参照）。



ISO/IEC JTC1/SC7/WG7国内小委員会2009.10提供

図2 「ニーズ」及び「要求」「要件」の概念整理

1.4.3 システム化計画の立案

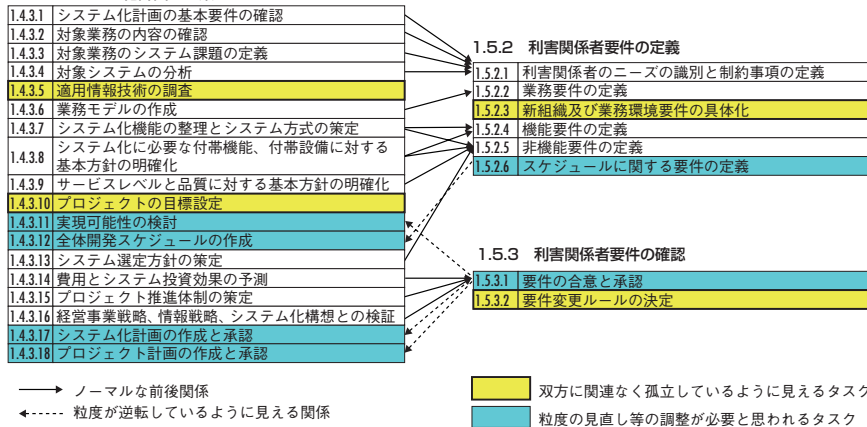


図3 企画及び要件定義のタスクの関係性と粒度の見直し

3.2 企画プロセス及び要件定義プロセスの整備

本来、企画プロセスの次に要件定義プロセスを行うよう、双方のプロセスに含まれるタスクの内容が記述されているはずである。しかしながら、例えば「スケジュールに関する要件」の記述レベル（粒度）を比較してみたところ、第1版では要件定義プロセスよりも企画プロセスの方で細かく記述されていることが分かった。そのため、このようなタスク内容の記述レベルの逆転関係や相互関係等を見直すことになった。

図3で示すとおり、記述レベルの逆転しているタスク間の見直し（破線の矢印）や、孤立しているタスク（太枠で示すタスク）等を見直した。個々のタスクでどのように改訂したかは省略するが、タスクの記述を追加したり変更したり、またタスクを新たに追加することも行っている。さらにガイダンス内容の強化も行った。このような企画プロセス及び要件定義プロセスのタスク内容の記述レベルの見直しを行った第2版の内容は、バランスの取れたものとなっている。

【見方】太字：追記された「呼出される主プロセス」を示す

1.7.8.1 業務運用の評価
 「また ～ 場合には、問題解決プロセス、保守プロセス、企画プロセス、**要件定義プロセス**又は開発プロセスに引き継ぐことができる。」

2.9 ユーザビリティ（使用性向上）プロセス
 「ユーザビリティ（使用性向上）プロセスは、～ を含む。このプロセスは、ソフトウェア又はシステムの**企画**、**要件定義**、開発、運用及び保守の全期間に渡って ～ 」

「開発者は ～ を管理する。ユーザビリティの専門家は、ユーザビリティアクティビティ及びユーザビリティアクティビティからの結果と**企画プロセス**、**要件定義プロセス**、開発プロセス、運用プロセス、**保守プロセス**及び支援ライフサイクルプロセスの ～ を融合させる。」

図4 呼出される主プロセスの追加例

3.3 プロセス同士の呼出し関係の見直し

共通フレーム 2007 では、JIS X 0160 に対して、企画プロセス、要件定義プロセスを独自に追加しているが（共通フレーム 98 でも同様）、第1版ではそれらのプロセスを後から追加したことによる不整合を見逃したままであった。具体的には、主プロセス又は支援プロセスの各プロセスから、他のプロセスを呼出している箇所が幾つかあるが、それらの箇所において、企画プロセス及び／又は要件定義プロセスの呼出し漏れがあり、それらを修正す

ることとした（図4参照）。

さらに、企画プロセス、要件定義プロセス以外のすべてのプロセスについても、相互の呼出し関係をチェックした（マトリクスを作成し、どのプロセスがどのプロセスを呼出しているのかを1つずつ確認）。その結果に基づき、主プロセスにおいて、文書化や構成管理、検証及び共同レビューといった各プロセスも必要な箇所と呼出されるように追記した（詳細は省略）。特に、後述する共同レビュープロセスについては重要なプロセスとして、随所で呼出されるようにした。

3.4 原理原則 17 ヶ条を第3部のガイダンスに追記

原理原則 17 ヶ条とは、SEC BOOKS「経営者が参画する要求品質の確保（第2版）」の付録として掲載されている「超上流から攻めるIT化の原理原則 17 ヶ条」を指す。これは、超上流（企画プロセス及び要件定義プロセス）を発注者及び受注者の双方が上手く進めるための重要なポイントを短い言葉でまとめたものである。条文ごとに「基本的な考え方」と「行動規範」が示されており、実務で活用できる有用なものである。

この原理原則 17 ヶ条は、第1版では全く触れられていなかったが、第2版では、共通フレームの本体である第3部において、プロセスごとに関係が深い17ヶ条をガイダンスで紹介することにした（図5参照）。読者に17ヶ条を作業上の留意点又は行動規範として意識してもらうことにより、実務において良い状況、良い結果をもたらすことを願ってのものである。

【例】1.5 要件定義プロセスの「ガイダンス」において

1.
2.
3. 「要件定義プロセス」については、「超上流から攻めるIT化の原理原則17ヶ条」における以下の原理原則を参照するとよい。
 原理原則[1] ユーザとベンダの想いは相反する
 原理原則[8] システム化の方針・狙いの周知徹底が成功の鍵となる
 原理原則[9] 要件定義は発注者の責任である
 原理原則[10] 要件定義書はバイブルであり、事あらばここへ立ち返るもの

図5 プロセスに関連する原理原則 17 ヶ条の埋め込み

3.5 「共同レビューの実施」タスクの追加

3.3項で説明したように、プロセス同士の呼出し関係を一から見直した。その中で、共同レビュープロセスの呼出し不足が目立ったので随所に追記した(図6参照)。例えば、要件定義プロセスの中には1つも共同レビュープロセスを実行するように明記されていなかったが、それではおかしとの指摘が挙がり、図6に示すタスク項番 1.5.1.2の(d)項として追加したのである。

この図6に示す改訂情報は、読者諸氏の開発標準等の整備を図る上で参考にして頂きたい。なお、蛇足ながら、JIS X 0160-1996 から引き継いでいる規定内容においては、開発プロセスのところで、くどいくらいに共同レビューのプロセスの呼出しが入っているという指摘があったことを

付記しておく(例えばソフトウェア詳細設計ではなくてもよいのではないかという意見もあった)。

3.6 V&V適用場面の解説図の追加

V&V (Verification and Validation) は、システム開発において共同レビューとともに不可欠なプロセスである。共通フレーム 2007 (第2版) では、今まで曖昧であったV&Vの適用場面を、図として提供している(図7参照)。この図は、システム開発に係る人々(経営層を含むユーザ及びベンダの方々)に対して、例えば妥当性確認はどのフェーズで行うのか、行うときには何と何を確認対象とするのか等を認識してもらい、ソフトウェアを含むシステムの品質向上を図るために作成したものである(第2版 pp.255-256 の解説を参照)。

図において妥当性確認は、V字の左側(上から下っていく線)に集中していることに注意されたい。システム(ソフトウェア)が設計工程を終えて実装されてしまってから行うのでは遅いのである。例外として、運用テストのときに、システム(ソフトウェア)が利害関係者要件を満たしているかどうかを確認するのは、運用に入る前の「最終的な」要求品質の確保という意味合いで行う。

共通フレーム 2007 では、具体的な妥当性確認の手段を規定してはいないが(幾つか例示はされている)、図7を見れば分かるように、利害関係者要件がシステム化構想・計画を満たす

【記号の意味】 項番：タスク番号、○：JIS X 0160に明記済、★：第1版で追記、●：第2版で追記	
○ 1.1.4.1 (取得プロセス内)	● 1.3.4.1 (契約の変更管理プロセス内)
○ 1.2.6.2 (供給プロセス内)	● 1.4.1.2 (d) (企画プロセス内)
★ 1.6.2.3 (開発プロセスのシステム要件定義)	● 1.5.1.2 (d) (要件定義プロセス内)
○ 1.6.3.5 (システム方式設計)	● 1.6.9.4 (開発Pのソフトウェア適格性確認テスト内)
○ 1.6.4.3 (ソフトウェア要件定義)	● 1.6.11.3 (システム適格性確認テスト内)
○ 1.6.5.7 (ソフトウェア方式設計)	● 1.7.1.11 (運用プロセス内)
○ 1.6.6.8 (ソフトウェア詳細設計)	● 1.7.2.3 ()
○ 1.6.8.6 (ソフトウェア結合)	● 1.7.3.2 ()
★ 1.6.10.6 (システム結合)	● 1.7.7.1 ()
○ 1.8.4.1 (保守プロセス内)	● 1.7.8.1 ()
○ 3.5.1.2 (c) (資産管理プロセス内)	● 3.1.2.2 (管理プロセス内)
○ 3.5.1.3 ()	● 3.1.5.3 ()
○ 3.5.2.3 ()	● 3.2.1.3 (環境整備プロセス内)
○ 3.6.2.3 (再利用施策管理プロセス内)	● 3.2.2.2 ()
○ 3.6.4.4 ()	● 3.3.1.2 (改善プロセス内)
○ 3.7.3.5 (ドメイン技術プロセス内)	● 3.4.1.2 (人的資源プロセス内)
○ 3.7.4.4 ()	● 3.4.2.2 ()
	● 3.4.3.2 ()
	● 3.4.4.2 ()
	● 3.4.6.2 ()
	● 5.4.2 (修整プロセス内)
○：14箇所、★：3箇所	●：21箇所

図6 「共同レビューの実施」タスクの追記

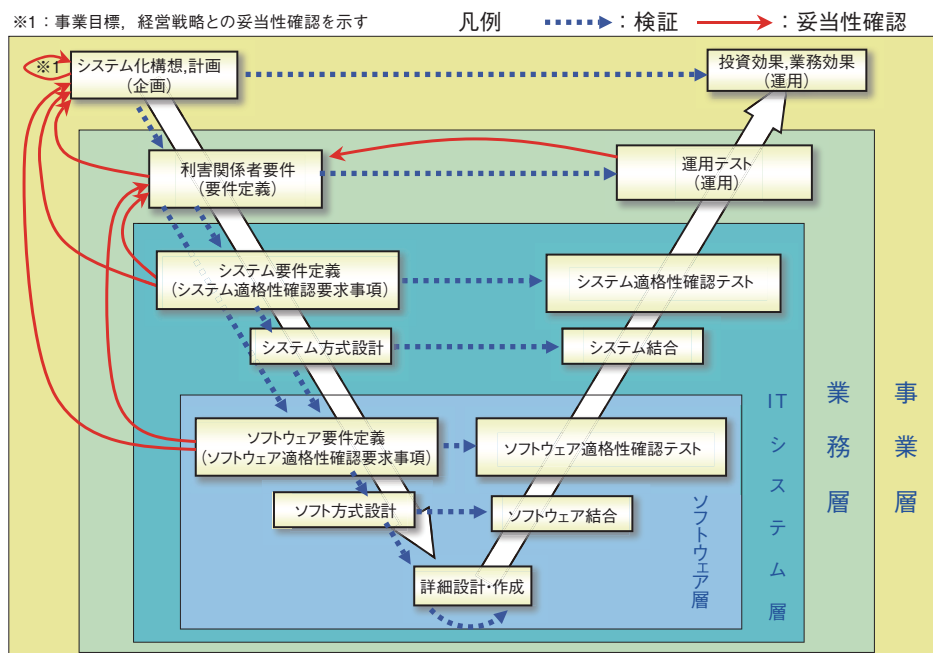


図7 V&Vの適用場面を示す図の追加

か否かの妥当性確認においては、矢印が通っている層の関係者（事業層では経営者、業務層では業務部門の責任者・担当者）の参画を暗に求めている。

さらに、図7において、検証を示す破線の矢印、妥当性確認を示す実線の矢印が、実は、トレーサビリティの確保をも示していることに注意されたい。例えばISO9001の審査のときに、この図を参考にして、作成したドキュメント同士の関係を、トレーサビリティの視点から説明すると有用であろう。

3.7 保守プロセスに係る規定とプロセス連関図の追加

企画プロセス、要件定義プロセスを日本独自に後から追加したことによる影響は、保守プロセスの規定内容にまで及んでいた。共通フレーム 2007では、ソフトウェアのみならず、ソフトウェアを含むシステムも規定範囲に含むため、保守プロセスから呼出されるプロセスは、開発プロセスだけではなく、企画プロセス、要件定義プロセス、運用プロセスもその対象となるのである。つまり、システム上に発生した問題を是正する上で、必ずしもソフトウェアだけが修正の対象となるわけではない。ときには、業務フロー

の変更が必要となるかもしれない。またシステム化構想の見直しにまで発展することも考えられる。となると、問題の影響分析の後に呼出すプロセスとしては、例えば業務フローの変更のときには要件定義プロセスにまで、システム化構想の見直しときには企画プロセスにまでそれぞれさかのぼって取り組む必要があるということである。

このようなことを第2版で明らかにし、図8に示すように規定内容を一部追加した。

また、第2版の第5部では図9に基づき、問題解決レベルや影響範囲の大きさにより、保守プロセスから呼出されるプロセスの範囲が異なることを、5つの事例を用いて解説している。

【第2版の p.151】（保守プロセスのタスクにおいて1文を追加）
「1.8.3.2 修正の実施
…」
また、修正対象によっては、企画プロセス（1.4参照）、要件定義プロセス（1.5参照）、運用プロセス（1.7参照）を利用してもよい。」
第4部の「2.2 主プロセスから主プロセスの呼出し方とその関係」において、この規定部分を引用している。

図8 保守プロセスにおいて追加した規定内容

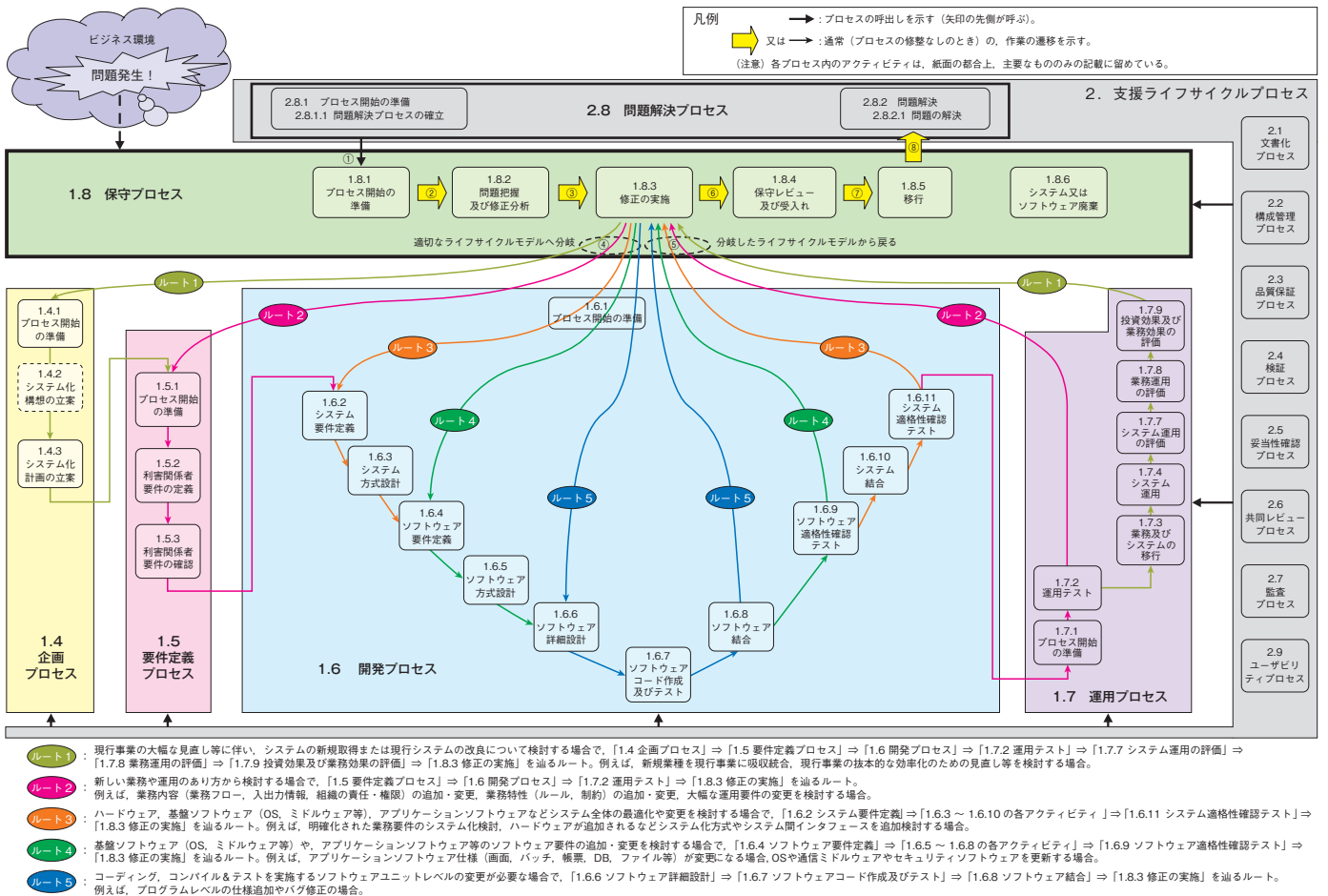


図9 保守を中心とした問題解決に関するプロセス連関図

図9が小さくて分かりづらいという読者の方は、第2版巻末折込みの挿入図(図5-6)そのものを見て頂きたい。同図において、まずルート1(保守プロセスから企画→要件定義→開発→運用→保守に戻るルート)としては、例えば2つの会社が合併してITシステムの一部を統合せよという経営方針が出されたときに実行される類のケースである。次のルート2(保守プロセスから要件定義→開発→運用→保守に戻るルート)としては、例えば人事や会計業務等の業務範囲の中で法規制事項の改正に伴う業務の見直し、ITシステムの機能変更に及ぶようなケースである。次いでルート3は、例えば人間系の業務には影響しないが、ITシステムの性能向上のためにハードウェアの更新及びソフトウェアのバージョンアップを行うようなケースである。ルート4は、ソフトウェアレベルに留まる問題のケースで、例えば単純にアプリケーション機能の追加や改良を行うようなケースである。最後のルート5は、最も影響範囲が小さいケースであり、例えばソフトウェアのバグが発生したときに、そのバグ修正のために実行されるのが典型的な例である。

読者諸氏には、第2版巻末折込みの図5-6を参考にして、開発標準又は保守標準の内容強化に取り組むことをお奨めする。

3.8 4つの保守タイプの定義を更新

ユーザとベンダとの間でトラブルに陥り易いのが、保守の範囲である。どこからどこまでを保守契約の範囲とするのかを、双方が事前に明確にし合意しておかないと、後々

になって保守範囲の解釈の違いによるトラブルが起こり易い。そこで役に立つのが、共通フレーム2007第2版のp.298で記述されている「4つの保守タイプ」である(図10参照)。

第2版では、最新のソフトウェア保守の国際標準ISO/IEC 14764:2006(日本ではJIS X 0161:2008が利用し易い)に基づいて、第1版に記述されていた保守タイプの定義を見直した。ただし、「完全化保守」の定義だけは、JIS X 0161:2008ではなく、旧規格であるJIS X 0161:2002による定義を採用している。その理由は、JIS X 0161:2008では「完全化保守」と「予防保守」の定義が似通ったものとなっており、非常に分かりにくく、利用者も解釈に迷うのではないかと判断し、違いが分かるJIS X 0161:2002の方の定義を採用することにしたものである。

ユーザもベンダも、図10に示す4つの保守タイプ(JISによる定義文を採用)を参考にして、お互いに保守契約の範囲について合意していくことをお奨めする。

4. 共通フレームを利活用する上での留意点

4.1 テーラリングプロセスの適用

共通フレームの特徴の1つは、テーラリングプロセスである。テーラリング(修整)とは、共通フレームをそのまま適用するのではなく、組織(企業)やプロジェクトの特性(例えば開発モデル)に合わせて、共通フレームで規定されているプロセス/アクティビティ/タスクを取捨選択したり、繰り返し実行できるように、又は複数をつまみ括って実行できるように組み替えたりする作業をいう。こ

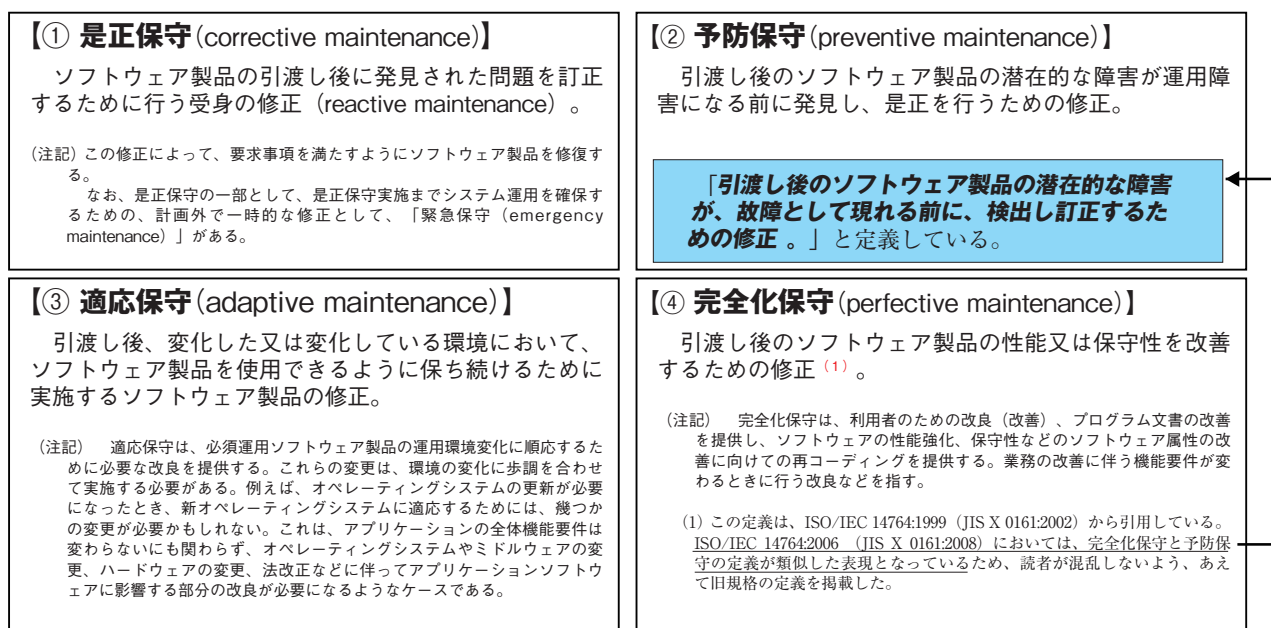


図10 4つの保守タイプ

のテーラリングプロセスが共通フレームに含まれていることによって、共通フレームの適用が非常に柔軟なものになる。

なお、共通フレームを適用する上では、以下の点に留意して頂きたい（第2版 pp.268-269を参照）。

- ①「共通フレームにある作業項目をすべてそのまま実行しなければならない」ということではない。
- ②「共通フレームにある作業項目」を、組織やプロジェクトが妥当と判断した場合には、省略してもよい。組織（企業）標準やプロジェクト標準に加えなくてもよいということ。
- ③「共通フレームに書いていないこと」を、組織（企業）標準やプロジェクト標準に追加してもよい。

4.2 タスクの実施順序

共通フレームの特徴の1つとして、「工程、時間、手法、ライフサイクルモデルからの独立性」というものがある。この意味は、共通フレームの第3部に記述されているアクティビティ及びタスクが並んでいる順番とおりに必ずしも実行するとは限らないということを示す。しかしながら、開発プロセスのアクティビティ及びタスクが並んでいる順序をみていくと、基本的には通常の新規開発の流れに沿っていることから、共通フレームとしては、第3部に記述されているタスクの順序とおりに実行するのが基本であると推察できる。

だが、1つだけ例外がある。それは、プロセスの最初のアクティビティとして定義されている「プロセス開始の準備」である。「プロセス開始の準備」の中の各タスクは、必ずしもそれらの並ぶ順番とおりに行う必要はなく、例えばプロセス開始に先立って複数のタスクを並行して実行してもよいのである。例えば、開発プロセスでの「プロセス開始の準備」とは、プロジェクトの特性に応じて最も相応しい開発モデルの選択や手法、ツール類の具体化など開発にあたっての段取りを行うことを意図している。その顕著な例を示そう。

運用プロセスの「プロセス開始の準備」アクティビティには、11つのタスクが含まれている。これらのタスクを項番順に実行することは、むしろ有害であるといえてよい。例えば、タスク「1.7.1.8 業務運用に係る作業手順の確立」は、要件定義プロセスの終了後に実行を開始し、開発プロセスと並行して進展させるのが妥当だと考えられる。他にも、タスク項番1.7.1.3、1.7.1.6、1.7.1.9も1.7.1.8と同様、要件定義プロセスの後に開発プロセスと並行して行うのがよいタスクである。この点に注意しないと、とんでもないことになる。ちなみに、この点については、第2版のp.35（①開発プロセス中の「業務詳細設計」の分散化）及びpp.276-277（2. 共通フレーム2007における業務主体作業の考え方）に記述されているので確認してみることをお

奨めする。

5. おわりに

本稿で解説した共通フレーム2007第2版は、1995年発行のISO/IEC 12207とその追補1(2002年)及び追補2(2004年)に基づいている。これらの規格は、各国で最も普及しているものである。一方、共通フレームのベースとなっている国際標準の動向をみると、システムライフサイクルプロセス（ISO/IEC 15288）が2002年に発行されたものの、両者間の粒度の違いや整合性が一部課題となっていた。そこで暫定的なハーモナイズ案として、ISO/IEC 12207:2008とISO/IEC 15288:2008が発行されている（その12207のJIS化作業が現在、財団法人日本規格協会INSTACにおいて進行中である。図1参照）。また国際標準審議の場では、ISO/IEC 12207、ISO/IEC 15288、関連規格との本格的なハーモナイズもさらに検討されているため、それを睨みながら、新たなビジネスを改善する取り組みをも加えた「新たな共通フレーム20XX」を必要とする時期が来るであろう。

最後に、産業界の取引の適正化を進める有効な手段として、共通フレーム94、共通フレーム98から連続と続いってきた日本独自の知恵、知見等が多く含まれている共通フレーム2007は、日本の情報サービス産業にとっての宝であり、ユーザとベンダとの間の共通な物差しになっていることを、改めて強調しておきたい。また、要件定義プロセスを先んじて導入・強化し、効果を上げている組織が増えていることから、読者諸氏におかれましても、共通フレームに基づくライフサイクルプロセスの適用（益々の横展開、深堀り）に取り組み、組織の一層の発展に寄与して頂くことを願ってやまない。

謝辞

本稿の執筆に際して、共通フレーム2007第2版の開発に携わったIPA/SECエンタプライズ系プロジェクトビジネスプロセス改善領域 プロセス共有化WG、国際標準化に携わっているISO/IEC SC7/WG7国内委員会から必要な情報を頂きました。御礼を申し上げます。

参考文献

- [1] IPA/SEC：共通フレーム2007第2版，オーム社，2009年
- [2] IPA/SEC：共通フレーム2007，オーム社，2007年
- [3] IPA/SEC：経営者が参画する要求品質の確保～超上流から攻めるIT化の勘どころ～第2版，オーム社，2006年

システム開発プロジェクトにおける 意思決定実態調査

—調査結果から導く「意思決定判断モデル」—

SECエンタプライズ系プロジェクト 研究員

中村 宏美

システム開発プロジェクトには多種多様な意思決定が存在する。個々の意思決定場面でもまた、複数の判断が行われている。更に個々の判断では、情報システムが利用される業務上の性質や利害関係といった要素が考慮される。考慮する要素が増えることで判断が複雑になり、意思決定を難しくする。

SECでは、企業における合理的な意思決定方法、及び意思決定時に考慮する要素についての実態調査を行った。本稿は、この調査結果の概要を述べると共に、そこから導かれた実践的な意思決定プロセス^{*1}(=意思決定判断モデル)を紹介するものである。

1. はじめに

1.1 価値と意思決定

情報システム開発には、様々なステークホルダー^{*2}がかかわっている。情報システムに対する、各ステークホルダーの要求やニーズ、期待する振る舞い等、意思決定の際に考慮する要素について、ここでは「価値」と定義する。価値は利害関係を持ってぶつかり合い、調整を要する。この調整が合意形成である。システム開発プロジェクトでは、合意形成を含む判断を繰り返し、最適な結論を導き出すことになるが、これがプロジェクトの意思決定である。

ここで気を付けたいのは、ステークホルダーはあくまで協調関係にある、ということである。意思決定は、互いの価値の共通認識と最適化を目指すものであり、ステークホルダーの対立結果、すなわち勝ち負けを決めるものではない。関係者双方が協力し合うアクティビティである。

1.2 意思決定プロセスの仮説

例えば、システム化企画段階で、「投資効果」という価値を考慮するとする。投資効果は「収益性」というメトリクスで測定するとして、実際はどこまで精緻な予測が出来るだろうか。価値の事前評価（以後、価値評価）は、意思決定の根拠として不十分な場合がある。そのような場合は、何をもって「根拠を充足」あるいは「根拠を代替」し、最適な意思決定としているのか。また、実態として、企業はどのような価値評価を利用しているのか。

以上のことから、システム開発プロジェクトでは、価値評価と意思決定プロセスの組み合わせにより、満足度の高い意思決定が行われている、という仮説を立てた。そして、この仮説を基に、

- ・どのような「価値評価」を行っているか
- ・どのように最適な判断を行っているか（意思決定プロセス）

について、企業事例を収集することを決めた。次章では調査の内容について記述する。

2. 調査内容

本調査は、企業の生の声を収集することに重きを置いた。独自の価値評価と意思決定プロセスを執っている企業（ユーザ企業10社、ベンダ企業6社）を対象に、アンケートを配布した。アンケートはプロジェクトの概要把握としての位置付けであり、具体的な意思決定内容については、アンケート結果を基に、意思決定単位で全件ヒアリングを行った。

2.1 調査項目

(1) 意思決定内容

プロジェクトで遭遇した意思決定局面での、具体的判断内容についての調査である。ヒアリングの冒頭でプロジェクト予算執行の流れを確認し、意思決定プロセスの全容を捉えたいと考えた。判断結果に対する満足度と、意思決定

※1 意思決定プロセス：JIS X 0170では、システムライフサイクルの意思決定プロセスについて、以下のように記述されている。

- ・選択肢が存在する場合にプロジェクト行動に最も有利な進路を選定すること
- ・最適化された成果を達成するために、いかなる発生源であっても、システムライフサイクルの間に遭遇した決定の依頼に応えること
- ・代替案が分析され、行動方針が選定され指示されること
- ・決定および論理的根拠は将来の意思決定を支えるために記録すること

※2 ステークホルダー（=利害関係者）：システムに、権利、持ち分又は請求権をもっている関係者。又は、関係者のニーズ及び期待に合う特性をシステムが有することに、権利、持ち分又は請求権をもっている関係者。（JIS X 0170）

に複数の判断がある場合はそれらの実行順序（プロセス）と重要度順について、ヒアリング内容に含めた。

システム開発プロジェクトで考えられる意思決定局面について、「SEC 価値指向マネジメント WG IT-VDM/VOM 概要版」^{※3}で、価値を考える局面として定義されている価値局面を参考に、洗い出しを行った。更に、「SEC 共通フレーム 2007 第2版」[共通フレーム 2007]及び、「経営者が参画する要求品質の確保」[SEC2006]に記述されているソフトウェアライフサイクルプロセスの定義を活用し、意思決定の判断場面を意思決定プロファイルとして表1のように整理した。以降、同プロファイルにより、事例を整理した。

表1 意思決定プロファイル

局面	意思決定
システム化 企画関連	情報システム導入判断
	情報システム受注判断
	予算枠の決定
プロジェクト 計画関連	予算額（実行予算）の設定
	カットオーバー時期の設定
	開発タイプの選定
	開発体制の決定
	プロジェクト計画の妥当性判断
	プロジェクト計画の変更
見積り関連	開発要件（要求内容）の決定
	見積り金額の決定
契約関連	契約方式の選定
	サービスレベルの合意
	契約金額の決定
要求管理/ 要件定義関連	機能要件の選定
	要求変更の受入れ可否
開発関連	完全内製/外注開発の判断
	外注先選定
	オフショア活用の要否
	開発プロセスの選定
	開発技術の選定
	リリース判断

(2) 価値評価項目

価値評価項目とは、意思決定の際に考慮した価値と価値評価方法（手法とメトリクス）を指す。これらが誰にとっての価値であるかを、ステークホルダー単位に見るフレームとして、価値マトリクス^{※4}（表2）を採用した。ステークホルダーは、「経営層」「業務部門」「情報システム部門」「システム開発部門」の4つにあらかじめ分類し、ステークホルダーの関係詳細は事例ごとに確認した。

(3) 制約条件

制約条件は価値の1つである。調整が効かない上、判断に影響を与える重要な要素として、価値と切り離してヒアリングした。どのような項目が制約条件として考慮されているか、傾向を見た。

2.2 各社事例

意思決定事例の整理方法として、IPOダイアグラム^{※5}を採用した。本章では、2社の事例を紹介する。

(1) 事例A（ユーザ企業）

旅行業A社の事例を図1に示す。

(2) 事例B（ベンダ企業）

金融系システム開発ベンダB社の事例を図2に示す。

表2 価値マトリクス

ステークホルダー	価値 /リスク	予測 (メトリクス)	意思決定	評価
利用ドメイン（業務部門）				
経営ドメイン（経営層）				
インテグレーションドメイン （情報システム部門）				
開発ドメイン（システム 開発部門）				

※3 SEC 価値指向マネジメント WG IT-VDM/VOM 概要版, <http://sec.ipa.go.jp/reports/20090415.html>
(VDM: Value Domain Model, VOM: Value Oriented Management)

※4 価値マトリクス: SEC 価値指向マネジメント WG 「IT-VDM/VOM 概要版」にて公開されている価値評価フレームワーク。ステークホルダーがどのような価値を持っているか、また、その価値を基にどのような意思決定を行うかを整理する。

※5 IPOダイアグラム: システムの機能を入力 (Input) - 処理 (Process) - 出力 (Output) の図式で表現する際に使用される構成図。ここでは、意思決定のプロセス単位に入力 (Input) = 価値、制約条件、処理 (Process) = 判断内容、出力 (Output) = 判断結果として使用している。

【旅行業A社の事例】

概要

- 旅行事業会社の店舗における会計システム再構築プロジェクトの事例。
- 個人旅行の精算業務、団体旅行の仮勘定、提携販売店との精算業務、管理会計等の機能を持つ。

意思決定のポイント

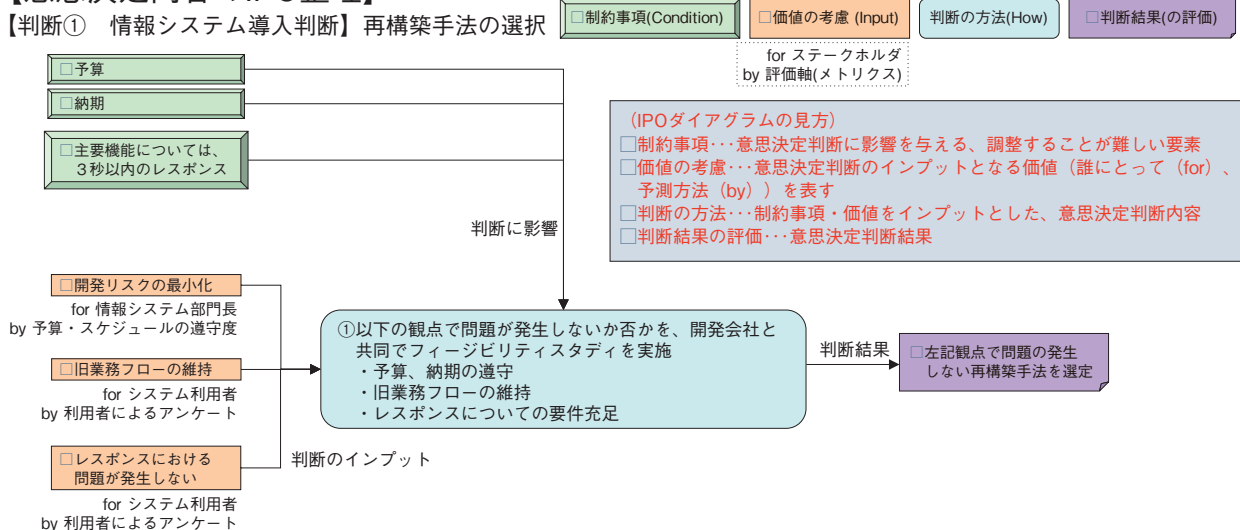
- 再構築内容の判断（制約事項（費用・納期等）を考慮した結果、サーバ統合のみとした）。
- パワーユーザとの機能要件の合意形成。
- リリース判断（パイロット稼働を行った後に、順次全国展開を実施）。

特記事項

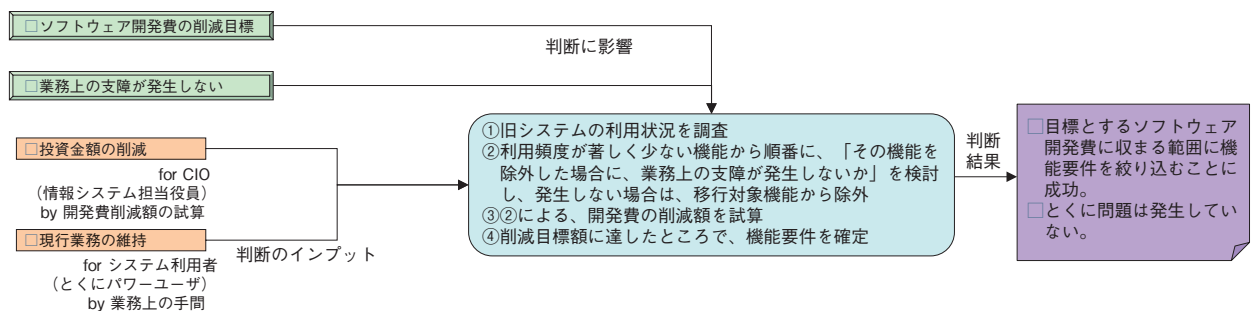
- サーバ統合のみ、という意思決定結果について、技術的な裏付けを取るために、フィジビリティスタディを実施。主要機能のレスポンスについて確認。
- 必要機能絞り込みのため、現行機能を1カ月間監視し、現行機能の6割しか利用していないことが分かった。この事実が「旧業務フロー維持」を求めるパワーユーザへの説明責任を果たした。
- 過去、類似プロジェクトの一斉展開で失敗した経験が、順次展開の判断をスムーズにした。

【意思決定内容のIPO整理】

【判断① 情報システム導入判断】再構築手法の選択



【判断② 機能要件の選定】機能要件の選定（絞り込み）



【判断③リリース判断】リリース判断

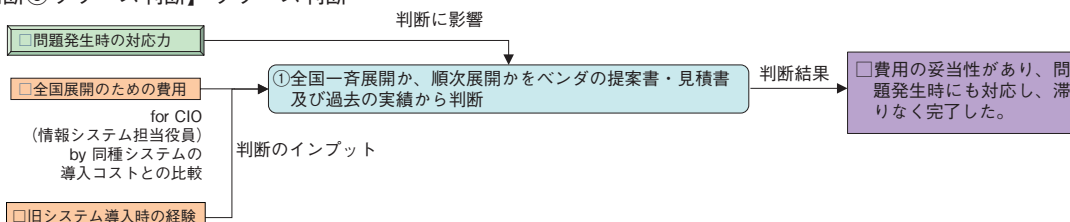


図1 事例A (ユーザ企業)

【金融系システム開発ベンダB社の事例】

- 概要
 - ・現行バックアップセンターの機能・サービスレベルの見直しを実施した事例。背景には、同業他社及び一般企業の動向がある。
 - ・外部ベンダへの一括外注(SI契約)とベンダとの協業(一部を内製化)を検討し協業体制での開発を選択、また、外注開発の範囲を決定した。
- 意思決定のポイント
 - ①一括外注/ベンダ協業(一部内製化)は、以下を総合的に評価し、ベンダ協業を選択した。
 - ・組織方針(「原則として内製化」) ・開発リスク ・開発委託費用
 - ②外注範囲の決定では、以下を評価した。
 - ・自社の技術力(自部門が保有する技術力) ・当該期間に捻出可能なワークロード(部門が保有するマン・パワー)
- 特記事項
 - ・新技術を採用する場合、外注による技術支援を検討項目に含め、開発リスクをコントロールする。 ・外注範囲は開発委託費に影響する。

【意思決定内容のIPO整理】

【判断① 完全内製/外注開発の判断】

一括外注/一部内製の判断

- 制約事項(Condition)
- 価値の考慮(Input) for ステークホルダー by 評価軸(メトリクス)
- 判断の方法(How)
- 判断結果(の評価)

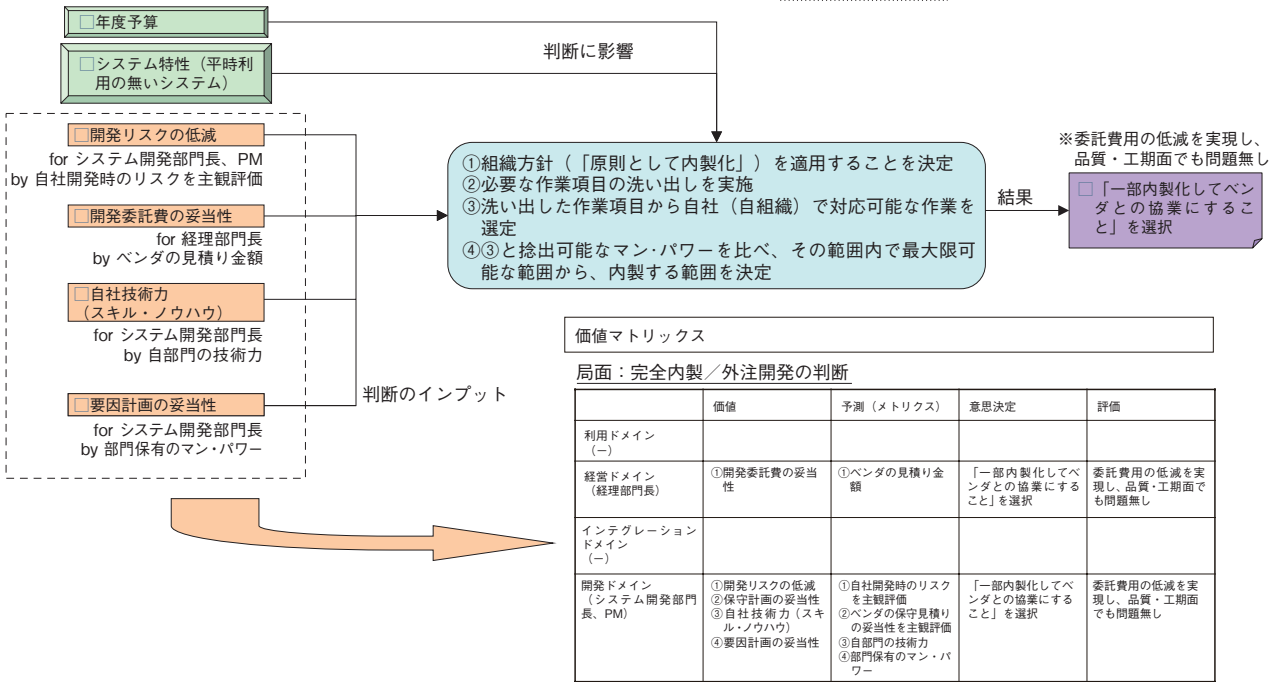


図2 事例B(ベンダ企業)

2.3 ヒアリング結果の考察

(1) 意思決定内容

まず、収集事例を意思決定プロファイルに従い整理した(表3)。事前にリストアップした意思決定は、システム開発プロセスの超上流に相当するものが多数を占めており、その内容に準ずる形で、企業事例が集まった。各企業の事例のうち、プロジェクト内で最も重要な意思決定を選択してもらったところ(表4)、こちらも開発関連に比べて、システム化企画関連やプロジェクト計画関連が重要視される内容となり、結果として超上流の考え方と符合した。

表3 意思決定事例一覧

意思決定	システム化企画関連	プロジェクト計画関連	見積り関連	契約関連	開発管理要件定義関連	開発関連	リリース管理
事例数	5	2	2	1	2	2	4
企業	金融・保険業A社	5				4	1
ITベンダ企業	ITベンダ(金融・保険業)B社	4	1	1			1
	ITベンダC社	3	1				2
	ITベンダ(金融・保険業)D社	5	1	1	1	1	1
	製造業E社	3	1	1			1
	ITベンダF社	3	1	1			2
ユーザ企業	金融・保険業G社	2		1			1
	情報通信業H社	4		2		1	
	製造業I社	4	2	2	1		
	金融・保険業J社	2	1	2			1
	製造業K社	3	1	1	2		
	情報サービス業L社	4	1	2	1		
	建設業M社	4	1	1	1		1
	旅行業N社	4	1	1		1	1
金融・保険業O社	2	1				1	

表4 企業が重要と回答した意思決定局面

意思決定局面	システム化企画関連	プロジェクト計画関連	見積り関連	契約関連	要求管理／要件定義関連	開発関連
プロジェクト数(全17)	5	6	3	0	1	2

表5 価値評価項目(ステークホルダー別)

ドメイン	ステークホルダー	判断時に考慮する価値	評価方法(マトリクス)	
経営ドメイン	社長、会長	投資効果 現状のリスクとサービスレベルアップによる効果	収益性 経営者の主観	
	経営層	管理情報の精度向上 業務の優先順位	管理データの正確性及び有効性 主要業務の列挙による相対的判断	
	経営執行会議	低コスト コスト削減	ベンジ提案の見積り金額 人員削減減額の見積り	
	情報システム部門担当役員	現状のリスクとサービスレベルアップによる効果	経営者の主観	
	リスク管理部門担当役員	現状のリスクとサービスレベルアップによる効果	経営者の主観	
	経理部門長	開発委託費用	年度予算額、見積り金額	
	ユーザ	利便性の向上 保守性の向上	見積り方法の選定 見積り金額の決定	
	業務部門の担当者	対象とする商品及び機能の充実 要件充足度(投資対効果)	充足度 ITアーキテクチャ選定会議での協議	
	<p>企業から収集した価値情報の生データ(同表は一例) 調査報告書では、これらに加え、体系的に整理した価値一覧を定義している</p>			
	利用ドメイン	業務内容の精度向上	業務データの正確性	業務データの正確性
業務品質の均一化		PJステコミでの協議	PJステコミでの協議	
収益機会の拡大		状況の可視化	状況の可視化	
現有人数での業務遂行		管理情報の信頼性	管理情報の信頼性	
業務内容の精度向上		業務量、システム依存度から影響度を主観判断	業務量、システム依存度から影響度を主観判断	
メインセンター機能停止時の影響範囲		システムを前提としない代替案の有無	システム依存度から代替可能性を主観判断	
システムを前提としない代替案の有無		業務量、システム依存度から期待されるレベルを主観判断	業務量、システム依存度から期待されるレベルを主観判断	
利用者視点でのサービスレベルアップ度合い		業務の標準化	PJステコミでの協議	
業務の標準化		不要業務の削減	作業所へのヒアリング	
不要業務の削減		利便性の向上	作業所へのヒアリング	
利便性の向上		管理の高度化	利用者の定性評価	
管理の高度化		投資効果の最大化	主計部へのヒアリング	
投資効果の最大化		主計部長	主計部へのヒアリング	
主計部長		制度改正への対応	ベンジ提案の実現する業務内容	
内部統制推進部		要求の充足性(含致性)	製品のデモ	
代理店	要求の充足性	営業現場の負荷(研修実施、テスト・アンケートの配布・回収)の軽減		
営業現場の負荷(研修実施、テスト・アンケートの配布・回収)の軽減	スケジュール通りのサービスイン	代理店からの意見聴取		
スケジュール通りのサービスイン	多くの改良要望の実現	要求の根本性と汎用性		
多くの改良要望の実現	顧客フォローの健全性	要求の根本性と汎用性		
顧客フォローの健全性	画面・帳票の整合性	必須対応性		
画面・帳票の整合性	外部連携変化(法制度改定等)への対応	重要度		
外部連携変化(法制度改定等)への対応	変更要求の受け入れ	対応しない場合の業務への影響度		
変更要求の受け入れ	スケジュール通りのサービスイン	説明会の実施状況、マニュアル・ガイド類の整備状況		
スケジュール通りのサービスイン	リリース後に業務面で問題が発生しないこと			
リリース後に業務面で問題が発生しないこと				

(2) 価値評価項目

意思決定の入力情報となる価値について、全体の傾向を見るために集約した(表5)。価値評価方法としては、コストにかかわる価値は定量的評価指標を使用することが多い。その他、とくに定性的な価値については、ステークホルダーの主観が大きく影響していることが分かった。

(3) 制約条件

制約条件の集約結果(表6)から、傾向としてはQCDの観点での項目が多く、同業他社の動向や、内規等の組織特有のしがらみも見られた。制約条件は他の価値を制限する性質を持つものと捉えられる一方で、意思決定の際、とくに複数の選択肢より1つを選ぶ場面において、選択判断の決定要因となるケースもあった。

表6 制約条件(調査報告書より一部抜粋)

分類	意思決定・合意形成	制約条件	制約条件の分類		
システム化企画関連	情報システム導入判断	組織内調整と納得性	内規		
		操作性	Q		
		コスト	C		
		行政改革(三位一体改革による歳入減少)	制度		
		ITがバネラス実現の必要性	制度		
		情報システム企画書テンプレート	内規		
		情報システム調達ガイドラインテンプレート	内規		
		業界の慣行(期間5年が一般的)	習慣		
		技術動向の変化	環境		
		業務の変化	環境		
プロジェクト計画関連	情報システム受注判断	<事例からは、未収集>	-		
	予算種の決定	民営化に伴う運用多様化の将来ビジョン	環境		
		納期	D		
		今期IT予算	C		
	RFP承認	<事例からは、未収集>	-		
	プロジェクト計画関連	予算額(実行予算)の設定	今期IT予算	C	
		カットオーバー時期の設定	<事例からは、未収集>	-	
		開発タイプの選定	短納期(3ヵ月)	D	
		開発体制の決定	短納期(3ヵ月)	D	
			稼働可能工数	C	
見積り関連		開発要件(要求内容)の決定	<事例からは、未収集>	-	
		見積り方法の選定	<事例からは、未収集>	-	
		見積り金額の決定	今期IT予算	C	
			短納期(3ヵ月)	D	
		契約関連	契約方式の選定	パートナーシップ契約についての内規(共同出資会社との5千万以上の案件)	内規
	サービスレベルの合意		参考ガイドライン(情報システムに係る政府調達へのSLA導入ガイドライン)	制度	
			IT依存度の高まり [調達管理の適正化]の動向(電子政府の取り組み)	制度	
	契約金額の決定		<事例からは、未収集>	-	
	要求管理／要件定義関連		機能要件の選定	[分かりやすくシンプルに]	内規
				サービスイン期日	D
			保守も含めた予算上限	C	
			短納期(3ヵ月)	D	
要求変更の受け入れ可否			アプライオナテストに間に合うこと	D	
			予算上限	C	
開発関連		完全内製、外注開発	<事例からは、未収集>	-	
		外注先選定	予算内収めること	C	
			平成21年6月の継続教育開始までの稼働がリミット	D	
			既存システム(なし)	環境	
		コスト	C		
		データセンターは国内が望ましい	内規		
	オフショア活用の要否	<事例からは、未収集>	-		
	開発プロセスの選定	<事例からは、未収集>	-		
	開発技術の選定	今期IT予算	C		
	リリース判断	説明会が完了していること	D		
	必要なマニュアルガイドの整備が完了していること	D			
	進捗遅延や未実施タスクが無いこと	D			
	所定の品質基準に到達していること	Q			
	必要なユーザIDが発行されていること	D			
	必要な帳票類の準備が完了していること	D			
プロジェクト計画の変更	<事例からは、未収集>	-			

(4) ヒアリング総括

意思決定をスムーズに行う決め手として、ステークホルダー間での事前の根回しや、プレゼンテーション等、コミュニケーションのスキルに依存する部分があることは、実態調査においても多く聞かれた。このような必ずしも価値だけが判断の決定要因とならない実態の中にも、価値評価を合理的な判断方法として利用する企業の工夫を抽出し、次章にてモデル化する。

3. 意思決定判断モデルの作成

収集した企業事例を基に、表1の意思決定プロファイルの中から、企業の回答数が多かった局面の意思決定判断モデル(12個)を試作した。本モデルでは、判断の結論を

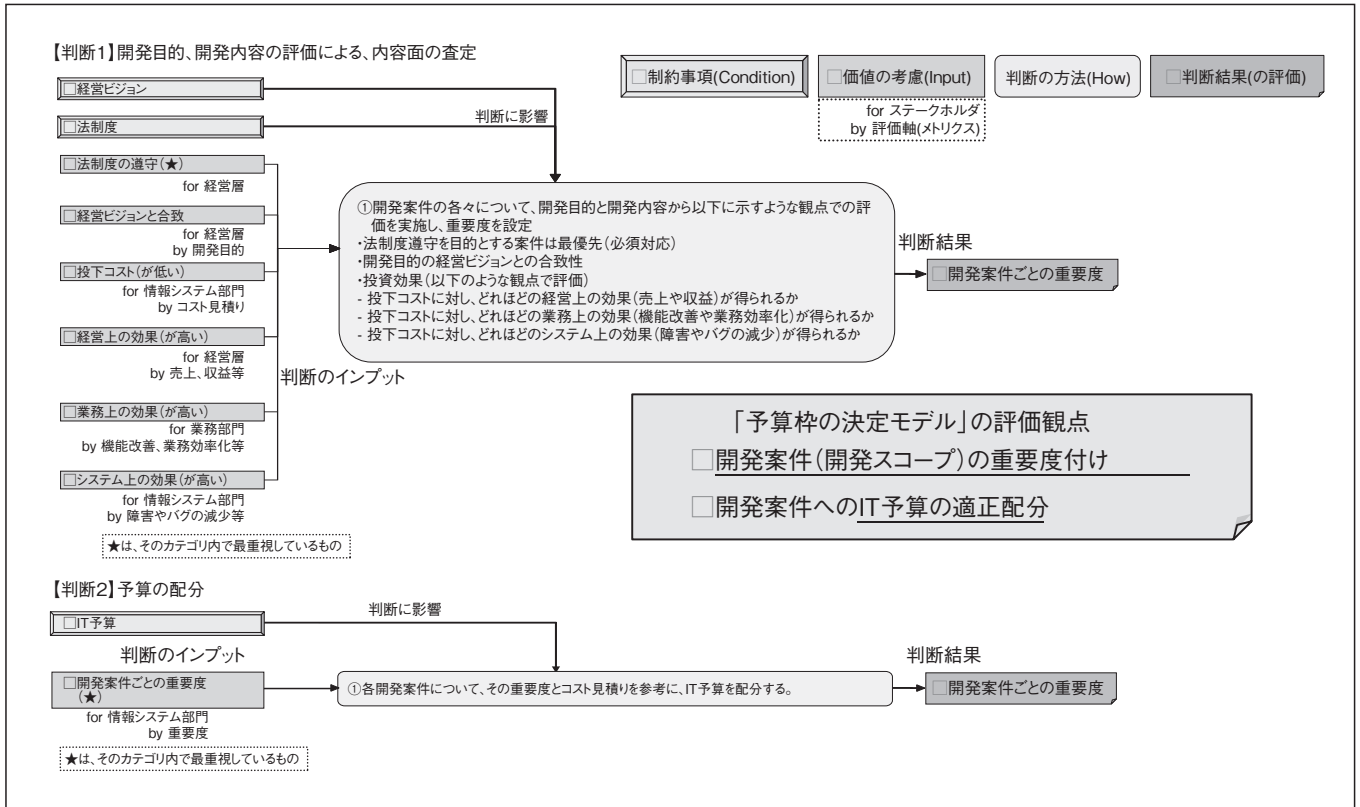


図3 システム化企画関連—予算枠の決定モデル

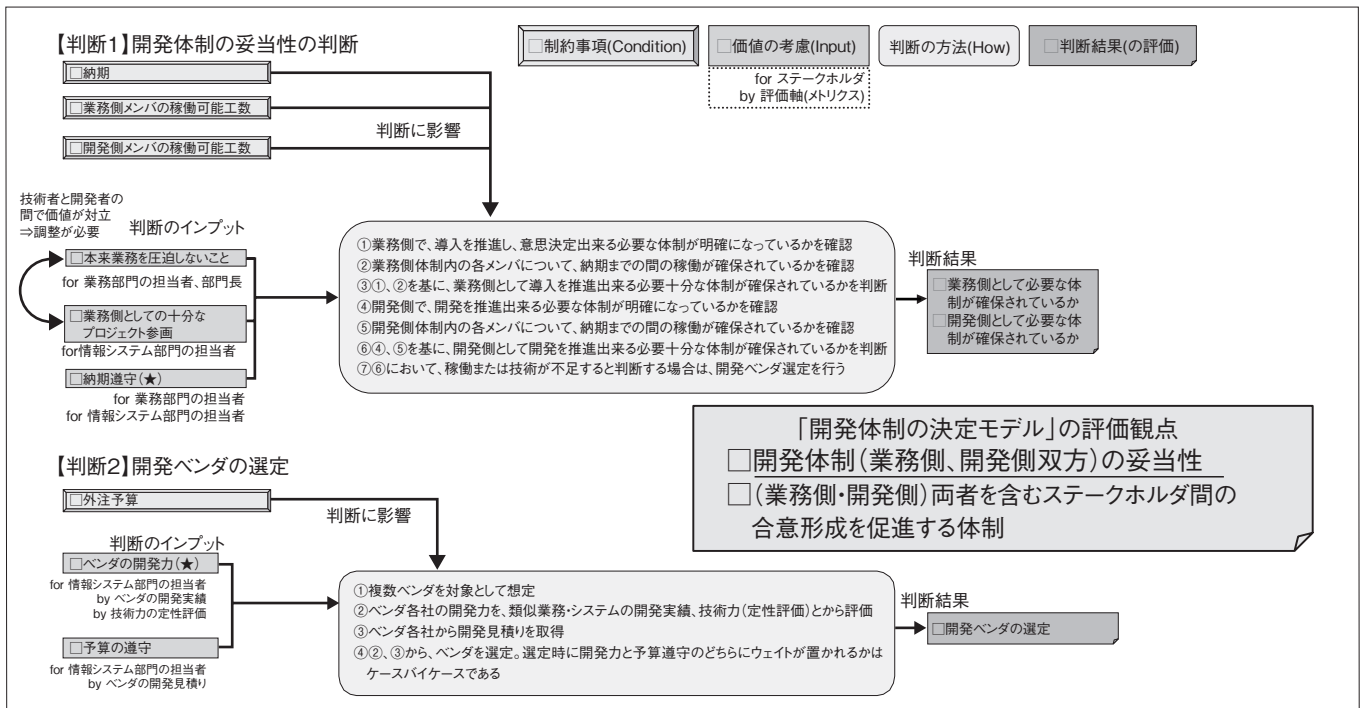


図4 プロジェクト計画関連—開発体制の決定モデル

導くものではなく、判断の際に考慮すべき事項を明示し、意思決定を助成するためのものである。これらのモデルのうち、「予算枠の決定」と「開発体制の決定」について紹介する。

3.1 システム化企画関連－予算枠の決定モデル (図3)

複数の開発案件候補がある中で、各案件の実施目的と内容から重要度を評価し、重要度の高いものから順に予算を配分するモデルである。重要度を評価する際、法制度改正への対応等、実施が必須である案件とそれ以外とを考慮している。そして、企業事例に多く見られた、次の2つの観点で判断する内容となっている。

- ・複数の開発案件（投資案件）の各々について、重要度を設定（重要度は、開発目的と内容（経営ビジョンとの合致性）、投資効果等がベース）
⇒開発案件（開発スコープ）の重要度付け
- ・各案件の重要度と、ベンダの参考見積りことから、全社の開発予算を案件に配分
⇒開発案件へのIT予算の適正配分

3.2 プロジェクト計画関連－開発体制の決定モデル (図4)

開発体制（業務側、開発側双方）の妥当性を評価し、必要に応じて外注先の選定を行うモデルである。ステークホルダー間の調整や合意形成の仕組み（コミュニケーションプラン）と、外注先の開発力を評価することで、開発体制を判断するモデルである。

調査では、開発体制の決定において、

- ・業務側の体制
- ・開発側の体制
- ・両者を含むステークホルダー間の合意形成を促進する体制

を確保する企業努力が見られた。作成したモデルは、これらを充足するロジックとなっている。

4. おわりに

本調査の報告書^{※6}では、価値評価について、GQM^{※7}や

AHP^{※8}といった戦略的な分析手法を紹介している。これらの手法によって、情報システムにかかわる価値評価の精度向上が望める。とくに、価値評価を初めて検討する企業にとって、調査報告書にあるAHPの活用例は参考になるので、ぜひ、参照いただきたい。なお、今回試作した意思決定判断モデルについては、次の局面での活用が可能である。

- ・システム企画局面（(例) システム導入判断等）
- ・要求管理・要件定義局面（(例) 機能要件選定等）
- ・見積り・契約局面（(例) サービスレベルの合意等）
- ・開発局面（(例) リリース判断等）

意思決定判断モデルは、実践的な意思決定のフレームを目指して提案しているものの、その判断ロジックはあくまで企業事例をベースとしている。つまりは、今回提案したモデルには、事例提供企業の文化や、業界の風潮が少なからず作用している可能性がある。しかし、他社の事例や標準化されたモデルは結局のところ、自社にとっての一例に過ぎず、重要なのは「いかに自分達の身丈に合った内容に落とすか」である。本調査が提案する意思決定判断モデルが、組織に合った意思決定プロセスを構築する際の参考となり、また、ステークホルダー間の合意形成に貢献することを期待する。

参考文献

- [SEC2006] IPA/SEC：SEC BOOKS 経営者が参画する要求品質の確保～超上流から攻めるIT化の勘どころ～ 第2版，オーム社，2006
- [共通フレーム2007] IPA/SEC：共通フレーム2007 第2版～経営者、業務部門が参画するシステム開発および取引のために～，オーム社，2009

※6 本調査の報告書：IPA「情報システム導入時の価値評価と合意形成に関する調査」報告書，<http://sec.ipa.go.jp/press/index.html>

※7 GQM：Goal, Question Metric. ソフトウェア工学におけるゴール指向の分析アプローチ。目標（Goal）定義と、それに対する課題（Question）、そして具体的なメトリクス（Metrics）につなげる部分をサポートする。目標に対してその実現度合いや実現のために必要なコントロールを可能とするためのメトリクスを洗い出す手法。

※8 AHP：Analytic Hierarchy Process, 階層分析法。複数の代替案があるときに、目的に最も合致した案を選択するための方法。評価基準を設定し、各案の評価値を決める。複数評価基準がある場合は、評価基準間で優先順位（重み付け）を付け、それを各評価値と掛け合わせ、総合的に評価値を算出する。

ソフトウェア開発データ白書の効果的な活用法

SECエンタプライズ系プロジェクト

研究員

小椋 隆

SECではソフトウェア開発プロセスの定量化及び標準化とその普及に取り組んでおり、その一貫として、開発企業のプロジェクトの実績データを2004年度より収集・分析し、「ソフトウェア開発データ白書」として発行している。

「ソフトウェア開発データ白書2009」[データ白書2009]では累計2,327プロジェクトのデータが集まり、規模、工数、工期、信頼性等のソフトウェア開発プロジェクトの主要な要素に関する分析結果を提示している。

本稿では、これらのデータをプロジェクト現場でより深く活用していただくため、いまだデータ活用が不十分な状況や、更に活用度合いを向上したいという課題に対し、ソフトウェア開発の様々な場面で、どのように定量データを活用するのかを、幾つかの事例を交えて解説する。

1 はじめに

データ白書にあるようなデータは、企業や組織の強み・弱みの可視化や評価、分析を通じたプロセス改善、プロジェクトマネジメント等、様々な目的や場面で利用されていると考えられる。とくに、開発プロジェクトの見積りや計画における予測に利用されることが多い、もしくはニーズが高いことであろう。

データ白書では様々なモノサシを定義し、新たなモノサシや課題抽出の切り口を提案すると共に、活用ニーズに応えるため、その精度を更に高める活動を続けている。

定量データの収集・活用がまだ不十分な企業や組織においては、データ白書を参照していただき、データの収集項目や相関、傾向性の把握、プロジェクトの見積りや計画策定等に利用していただくことを推奨している。

例えば、規模と工数の関係を用いて、規模測定した結果から工数を見積ることや、見積った工数を基に、工数と工期の関係から妥当な工期を予測するということを、データ白書にある散布図のグラフや基本統計量の表を目安に利用していただくということである。

しかし、データ白書は複数企業からデータを収集していることから、データのばらつきがあることも事実である。

これは、既に定量データの収集・活用を行っている企業でも見られるように、複数のプロジェクトや組織から収集した全社的なデータの状況に似ているかもしれない。また、そのような状況から、収集したデータの分析結果は、予測に利用するには現場の感覚と異なってしまう等、利用度合いが高まらない要因にもなっているのではないだろうか。

Xに値を代入すればYという答えが求まる、といった予測を主眼とした考え方も必要ではあるが、ここでは、更に定量データを活用していただくために、予測だけではな

い活用方法や、広い範囲から収集したデータならではの活用方法について、事例を交えながら解説する。

開発現場の視点に立ち、ソフトウェア開発ライフサイクルを通じた、見積り、計画、コントロール、評価について考え、以下の場面について、データ白書内の図表の利用方法や事例を示す。

- ・工数、工期の見積り
- ・要員山積みの計画
- ・プロジェクト遂行時での今後の見通し
- ・実績の評価

2 工数、工期の見積り

見積り手法は、①類似システムとの比較・類推、②積み上げによる方法、③パラメトリックなモデルによる予測の3つに大きく分類出来るが、各手法を相互補完的に使用することが重要である。

データ白書では、③のように実績に基づいた関係式、例えば工数等を目的変数として、説明変数に規模や要因等を設定し、数学的な関数として表している。

ただし、説明変数に値を設定して答えを算出するというだけでなく、定式性や特性を分析し、例えば「適正な工数や工期の範囲」を提供することも、狙いの1つとしている。

分析結果として、傾向や相関が見えれば、散布図上での信頼幅を利用し、見積りや計画の妥当性確認や、工数、工

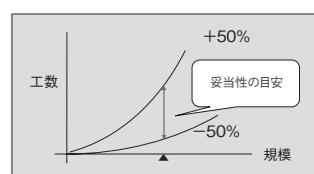


図1 規模と工数の関係

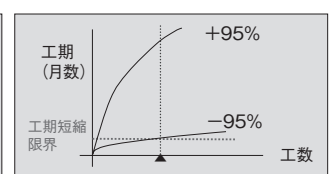


図2 工数と工期の関係

期短縮の目標の目安としても利用出来る。

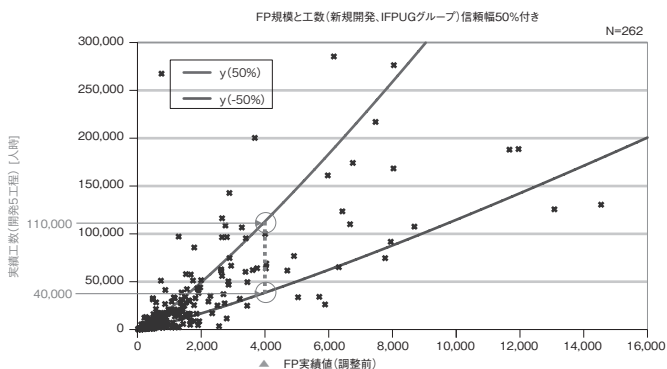
- ・ ± 50%の信頼幅に入っていれば、妥当性が高いとみなす (図1)。
- ・ 工数・工期短縮の要求に対し、それが対応可能かどうか、信頼幅 95%の下限値を限界の目安とする (図2)。

2.1 工数の見積り

図3は、FP規模と工数の関係を表したもので、図の中に引かれているのは50%の信頼限界で囲まれた領域を示している。

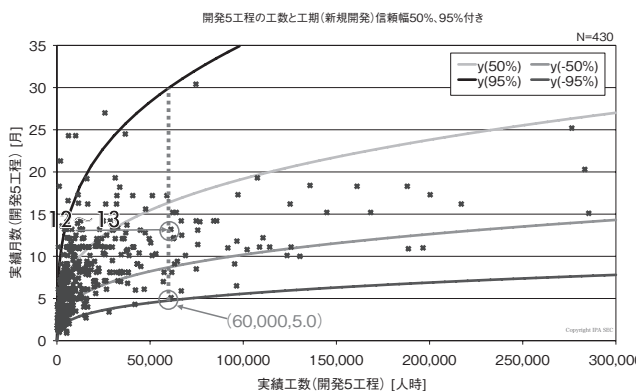
この図においては、「± 50%の信頼幅に入っていれば妥当性が高い」とみなし、見積った規模と工数が信頼幅のどの位置に当たるかにより、判断の目安にすることが出来る。

例えば、4,000FPのプロジェクトを計画している場合、見積工数が約40,000～110,000人時(約250～690人月)の範囲にあれば妥当とする考えである。しかし、ここで気を付けなければならないのは50%幅内ではあるが、幅に大きな差があるように見て取れる。従って、この範囲に入っていれば妥当というよりは、外れていればリスクが高い、または実現が難しいという目安に利用するとよいだろう。



[データ白書2009] P128, 図表6-4-9

図3 FP規模と工数の関係



[データ白書2009] P118, 図表6-3-2

図4 工数と工期の関係

2.2 工期の見積り

プロジェクトの工期、とくに納品日については、発注者からの要求事項の1つになることも多いが、実現可能性の面で工期を見積り、その妥当性を確認することは必要である。

図4は、工数と工期の関係を表したもので、図には50%及び95%の信頼幅が示されている。

この図においては、信頼幅95%の下限値を限界とみなし、期間短縮や納期変更への対応が可能かどうかを判断する目安が得られる。

例えば、工数が約60,000人時(約375人月)の場合、工期(月数)の中央値は12～13カ月で、信頼幅95%の下限値の工期(月数)は約5カ月である。中央値をよくある工期と考えても工期短縮を目指す場合、5カ月以下にするのは難しいと考える。

広い範囲から集めたデータの分析であるからこそ、ほとんど実績のない部分は、限界点であると言えるという考え方である。

また、信頼幅50%の下限値は9カ月であることから、まずは9カ月を工期短縮の目標にする考え方もある。

3 要員山積み計画

表1、表2は、工数の比率及び工期の比率の基本統計量である。

ソフトウェア開発プロジェクトのデータは、正規分布をしていない場合が多く、全体のプロファイルを表す数値としては、平均値よりも中央値が適切であることが多い。例えば表1の工数の比率においては、製作工程の中央値を見ることができ、プロジェクト全体の工数の35%弱を製作工程が占めていることが分かる。

このような工数の比率と工期の比率(データ白書では月数の比率)のデータを利用して、見積り結果として引き継

表1 工数の比率

(新規開発)		(単位:比率)						
工程	N	最小	P25	中央	P75	最大	平均	標準偏差
基本設計	428	0.001	0.095	0.147	0.205	0.589	0.161	0.096
詳細設計	428	0.016	0.114	0.166	0.220	0.613	0.174	0.086
製作	428	0.018	0.272	0.349	0.449	0.847	0.370	0.148
結合テスト	428	0.002	0.106	0.155	0.211	0.938	0.167	0.096
結合テスト(ベンダ確認)	428	0.000	0.061	0.117	0.175	0.564	0.128	0.087

[データ白書2009] P239, 図表8-1-8

表2 工期の比率

(新規開発)		(単位:比率)						
工程	N	最小	P25	中央	P75	最大	平均	標準偏差
基本設計	115	0.016	0.155	0.230	0.302	0.522	0.234	0.107
詳細設計	115	0.026	0.143	0.189	0.248	0.645	0.201	0.093
製作	115	0.047	0.209	0.260	0.350	0.902	0.280	0.118
結合テスト	115	0.016	0.091	0.143	0.185	0.386	0.146	0.068
結合テスト(ベンダ確認)	115	0.014	0.074	0.121	0.184	0.571	0.140	0.091

[データ白書2009] P236, 図表8-1-2

がれた工数、工期、スケジュールに基づき、工数山積みの妥当性を確認することが出来る。

以下のような手順により、各工程にどの位の工数と期間がかかるかの目安と共に、要員を各工程に何人投入するかの目安も導き出すことが出来る(表3)。

- ① 開発すべきシステムの構築に必要な総工数、全体工期を算出
- ② 工数比率によって、各工程に必要な工数を算出
- ③ 工期比率によって、各工程に必要な期間を算出
- ④ 上記②と③から、各工程に必要な要員数を算出

表3 工数、工期、要員数の算出手順

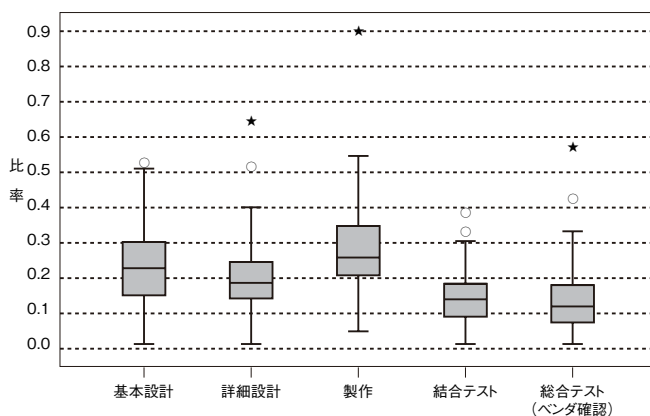
	基本設計	詳細設計	製作	結合テスト	総合テスト	合計
工数	②					①
工期	③					
要員数	④					

4 プロジェクト遂行時での今後の見通し

プロジェクトが進むに従い、作業進捗や工数等、当初の計画と実績に差異が発生する場合がある。差異を吸収して計画通りに完了させるため、それまでにかかった期間、消化工数、生産性等から、今後の見通しを予測することが必要となる。

ここでは、プロジェクトの基本設計工程を終えた段階で、定量的に現状把握を行い、今後の見通しについて評価する事例を挙げる。

図5は開発5工程の工程別の工期の比率を箱ひげ図で表したものである。この図からは基本設計工程は全工期の約15~30%を占めており、中央値は23%であることが分かる。それ以降の工期は基本設計工期の割合を引いたものであり、試算すると約2.3~5.6倍、中央値からは3.3倍かかると予測出来る。



[データ白書2009] P236, 図表8-1-1

図5 工程別の工期の比率(箱ひげ図)

例えば、計画値で全工期が6カ月半のプロジェクトの場合、基本設計が2週間遅れて、1.5カ月の予定が2カ月かかったとすると、計画上は残り4カ月半である。基本設計以降の工期は基本設計の3.3倍が必要だとすると、必要な工期は6.6カ月となり、計画よりも2カ月遅れる計算となる。基本設計の2週間の遅れを安易に考えていると、最終的に2カ月も遅れるリスクを見落とすことになるのである。

このように定量データは、見積りや計画時の予測だけでなく、今後の見通しやリスク評価に利用することも出来る。

5 実績の評価

プロジェクトが完了すると、工数、工期、生産性等の実績データからプロジェクト評価を行い、また後のプロジェクトで利用していくために、データ収集、分析、フィードバックといった、データ活用サイクルを回すことになる。ここでは、プロジェクト評価を行う際の留意点を事例から示していく。

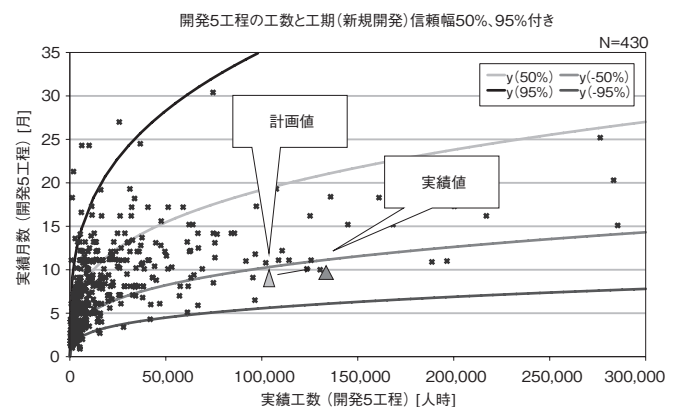
5.1 プロジェクト完了時のプロジェクト評価

図6は工数と工期の関係におけるプロジェクトの計画値と実績値の推移を示したものである。

このプロジェクト事例は、短期開発のため要注意プロジェクトと指定し重点監視したが、計画時(100,000人時、10カ月)の予定が、実績(130,000人時、11カ月)となってしまったものである。

結果から見て開発力が妥当な範囲か評価してみると、工期と工数がここまでオーバーしてしまったことは、マイナスの評価であろう。

しかし、図から認識出来ることとして、工数と工期の信頼幅下限50%を下回り、計画段階から工期が厳しく挑戦的な工期であったことが分かる。実績も同様で、工数の割に工期が厳しく、短期での開発力、管理力が備わっている



[データ白書2009] P118, 図表6-3-2

図6 工数と工期の関係(実績表示)

と評価出来るのではないだろうか。また、工数の増加は、工期を守るためのトレードオフと推察出来る。

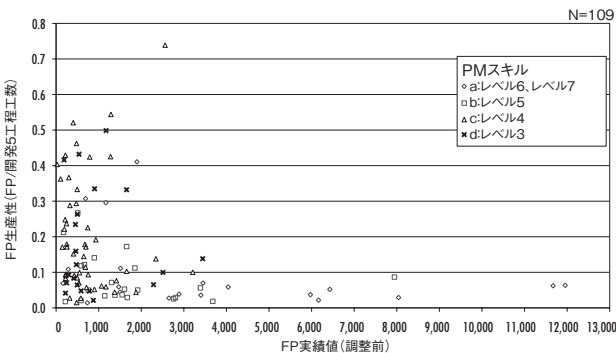
このような評価から、PM（プロジェクトマネージャ）は平均要員数60人規模のプロジェクトを短期間で遂行する能力があり、これを強みと捉えることが出来る。プロジェクトで実践した施策、手法をプロセスとして文書化すると共に、工程別工期の比率を自社の特徴と捉えて保管することにより、新たな類似プロジェクトにおいて利用することが出来る。

5.2 生産性から見た担当PMの評価

「ソフトウェア開発データ白書2009」では、図7のように新規開発プロジェクトを対象に規模と生産性をPMスキル別に層別して示している。

規模の小さいプロジェクトは生産性のばらつきが大きく、PMスキルの分布もばらついている。規模の大きいプロジェクトは生産性が低い傾向と同時に、PMスキルが高い傾向が見られる。

つまり、生産性が低いという部分のみでPMを評価するのではなく、規模の大きさや工期の厳しさ等、生産性に影響を与える要素を含め、多角的に分析することで、適切な評価を行うことが出来る。



[データ白書2009] P275, 図表9-2-6

図7 PMスキル別のFP規模とFP生産性

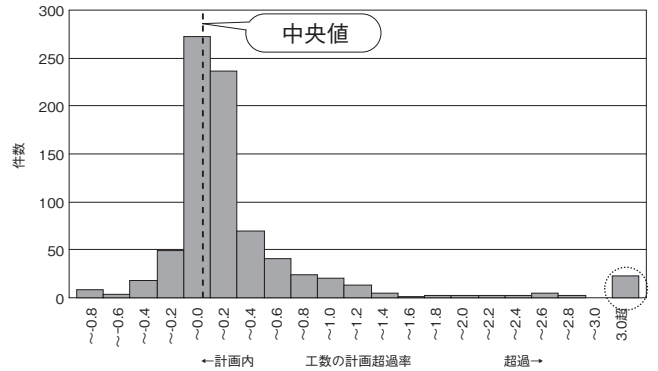
5.3 プロジェクトマネジメントの視点

図8は工数の計画と実績の差の比率の分布である。

中央値は0.023 (2.3%)、平均値は0.483 (48.3%)であったが、平均値が大きいのは、超過率が3.0以上のものに引きずられているためである。

定量データの活用においては、予測だけではなく、このように問題が突出してしまうリスクを未然に防止するといった、プロジェクトマネジメントに活用することも重要である。

$$\text{計画超過率} = \frac{(\text{実績値} - \text{計画値})}{\text{計画値}}$$



[データ白書2009] P267, 図表9-1-10

図8 工数の計画と実績の差の比率の分布

6 おわりに

今回は信頼幅を用いた限界の考え方等事例を示したが、定量データも活用次第で効果を上げることが出来る。

「ソフトウェア開発データ白書」は紙ベースであり、今回ご紹介した事例等を実際に利用するには不便な点もあるが、SECでは、SEC Web サイト上で利用出来る、「プロジェクト診断支援ツール^{※1}」を公開している。

データ白書に掲載されている様々な図表の参照だけでなく、利用者がツールに入力した自社のプロジェクトデータを、画面の図表上にプロット出来、他社データとのベンチマーキングを可能にしている。

プロジェクト診断支援ツールは、「SEC journal No.19」でもご紹介しているので参照して、データ白書と共に活用していただきたい。

なお、プロジェクトの実績データを提供していただける企業を随時募集しております。SEC Webの「SECへのお問い合わせ」よりご連絡いただければ幸いです。

参考文献

- [SEC2006] IPA/SEC: SEC BOOKS ソフトウェア開発見残りガイドブック～ITユーザとベンダにおける定量的見残りの実現～, オーム社, 2006
- [データ白書2009] IPA/SEC: ソフトウェア開発データ白書2009～2327プロジェクト 定量データ分析で分かる開発の最新動向～, 日経BP社, 2009

※1 プロジェクト診断支援ツール: <http://sec.ipa.go.jp/tool/pasa.html>

流用開発を意識した ソフトウェア実装品質向上手法



山口 有紀† 田中 尉氏† 市川 裕章†

多くの組込みシステムでは既存製品の設計資産を流用して開発するスタイルが主流となっている。この方式の場合、製品の主要部分を構成するソフトウェアの保守性が重要になる。本稿では弊社が開発を進めている民生用光学機器のソフトウェア開発において、IPA/SEC が策定したコーディング作法を参考に、静的コード解析ツールと目視レビューを組み合わせ、流用開発におけるコードの保守性を向上させた試みについて報告する。

An Approach to Quality Improvement in Software Implementation by Assets Reusing

Yuki Yamaguchi †, Yasushi Tanaka †, and Hiroaki Ichikawa †

The style of software development based on design assets reusing of existing products has become the mainstream in most embedded system development. In above case, the maintainability of the main part of the software products becomes important. In this paper, we proposed a novel approach of quality improvement using combination of code static analyzer and code review, by referring to IPA/SEC coding manner. This paper also reports a trial of this approach for actual development of optical consumer products.

1 はじめに

近年、多くの組込みシステムでは製品のシリーズ化等を図り開発効率を向上させる工夫がなされている。製品のシリーズ化に向けた取り組みとしては既存のソフトウェア資産を戦略的に活用し、ソフトウェア再利用を中心に据えた開発が主流となっている。一方で、近年の組込みソフトウェアでは、その機能の多様化に伴いソフトウェア規模が増大する傾向にあり、主要な部分を構成するソフトウェアの品質確保は極めて重大な課題となってきている。

こうした傾向は、弊社が手がける民生用光学機器ソフトウェア開発の分野でも同様であった。そのため、弊社では製品シリーズとして開発される光学機器に搭載されるソフトウェアについて、派生開発を意識し流用部分のソフトウェアの資産価値の向上を目指すべく、コード品質向上の一環としてコード品質のチェックを推進してきた。

具体的には、C 言語で実装されるこれらのソフトウェアに対して、C 言語コードを解析する静的コード解析ツールを導入し、人手を介すことなくその品質をチェックする方

策を模索してきた。しかしながら、弊社ではこれまでに幾つかの開発プロジェクトでこうした方策を実施してきたものの、ソフトウェア品質向上に関する効果は限定的なものにとどまってきた。

このため、今回、社内の開発部門、品質部門メンバから構成されるワーキンググループ (WG) を中心に、弊社で利用している静的コード解析ツールを提供している企業と組込みソフトウェアの実装品質に関するガイドを策定している IPA/SEC にも協力をお願いし、流用開発におけるソフトウェアのさらなる品質向上施策について検討し試行適用を行った。

2 流用開発に求められる品質と従来方式の課題

2.1 流用開発に求められる品質

弊社が手がける民生用光学機器分野は関連するハードウェアデバイス等の進化に伴い製品で実現する機能が多様な方向に多彩に変化していくことが特徴であり、既存の製

† 富士フイルムソフトウェア株式会社, FUJIFILM SOFTWARE CO., LTD.

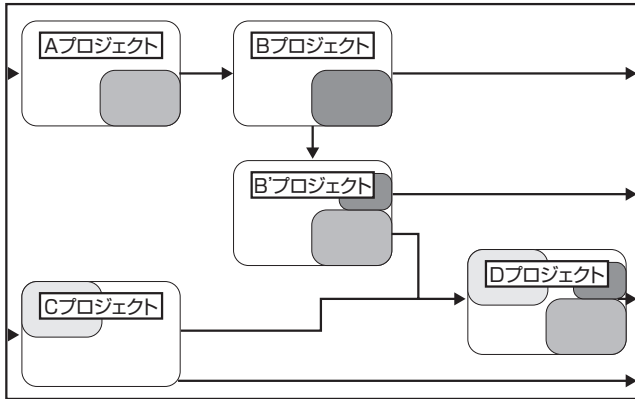


図1 流用開発における複雑な製品系列化の様子

品をベースにした流用開発が行われることが多い。例えば図1に示すようにAプロジェクトのソフトウェアを利用してBプロジェクトのソフトウェアを開発し、その一部を再使用してB'プロジェクトを作成する。一方で独立に開発したCプロジェクトのソフトウェアの一部とB'プロジェクトの一部を組み合わせてDプロジェクトのソフトウェアを開発するといった具合にソフトウェアの流用やそれに伴う派生バージョンが次々と開発されていく。このように一般に流用開発では、流用やそれに伴う派生を繰り返すことで、あるベース機種から複数の派生機種が生まれ、そこで実現された機能の中から更に幾つかが融合して更に新しい機種が生まれるという複雑な製品系列を構成する場合が少なくない。また、こうした製品系列の複雑さは、そこに搭載される組込みソフトウェア自身の複雑さをも誘発することにつながってくる。すなわち、こうした複雑な製品系列の中で多くの開発者が部分的に手を加えることで流用部分が場当たりの保守されていくといった問題が発生している。中長期的にソフトウェアの流用開発を進めていくためには、とくに保守性等の品質特性が極めて重要になる。しかし流用開発が場当たりに進められてしまうと、ソフトウェアの保守性について戦略的かつ効果的なアプローチをとりづらくなってしまふ。結果として流用されるソフトウェアの保守面での不安が弊社のビジネスを展開する上での大きな課題となってきた。

2.2 従来方式とその問題点

(1) 従来方式

こうした流用ソフトウェアの保守性に関する不安を解決することを目的として、弊社ではまず、ソフトウェア資産価値の向上が重要と判断し、コードレベルでの品質向上を試みてきた。具体的には、社内で開発されるプログラムのコードを対象にC言語プログラムの静的コード解析ツ

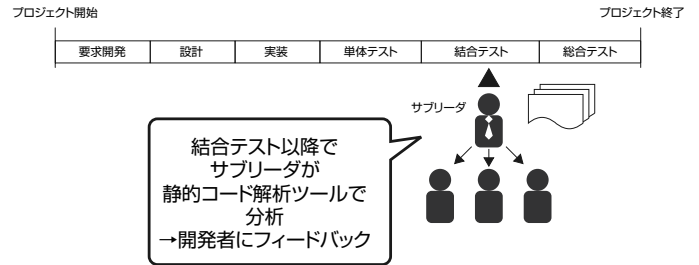


図2 従来方式のコード解析のタイミング

ルを導入し、コード品質面からのチェック強化を試みてきた。

・静的コード解析ツール

C言語プログラムの静的コード解析ツールは、C言語で記述したコードを入力として与え、C言語の文法や様々なコーディングルール等をチェックし、問題のある箇所をワーニングとして指摘するものであり、組込みシステムの開発の現場でも数多く利用されているものである。弊社ではこのツールを導入し、以下のような方法でコード解析を実施してきた。

・静的コード解析ツールの従来方式での活用方法

図2に示すように、従来の方式ではプロジェクトで作成されたコードを集約して、サプリーダが静的コード解析ツールを操作・解析し、その結果を開発者にフィードバックするという方式を採用してきた。この理由は、出来るだけ開発者の負担を少なくするためであり、開発者はサプリーダから挙げられるレポートを参考にコード内で指示された部分を修正する形をとっていた。

また、静的コード解析ツールを使用するタイミングは結合テスト以降の開発終盤であった。この理由は、流用ソフトウェアの品質は安定しており、開発部門では静的コード解析ツールによる確認を安心材料のようなものとして位置付けていたためである。

(2) 従来方式の問題点

弊社では上記のような方法によるコード品質チェックを数年前より導入し推進してきた。しかしながらこの方式を繰り返す中で、下記に示すような幾つかの課題が明らかになってきた。

・課題1：従来の方式では、コード品質チェックに重点を置いていた反面、コードを作成する際にどのような点に注意すべきかについて、開発者に事前に十分周知してこなかった。このため、開発者はコードの品質等をあまり意識することなくコーディングし、玉石混淆のワーニ

グが発せられる場合も少なくなかった。

- ・課題2：開発終盤でコード解析を行うため、指摘事項の検討や修正をするための十分な時間が取れないケースや、リリース直前の段階でコードに手を加えることへの抵抗感も大きく、結果的に指摘事項への対応を次版以降に見送るケースがある。
- ・課題3：一般に静的コード解析ツールからの指摘事項は膨大な量となる場合が多い。このため弊社ではサブリーダがツールを利用してコードを解析した後、その結果を整理して開発者にフィードバックするための解析レポートを作成する時点で、ツールからの指摘事項をある程度絞り込む作業を行っている。しかしこの解析レポートを作成する際に、サブリーダもツールからの膨大な指摘事項をすべてチェックし切れず、結果的に重大な指摘事項が解析レポートから漏れてしまうケースもある。
- ・課題4：サブリーダによって作成される解析レポートにも数多くの指摘事項が記載される。このため、開発者は指摘された事項について、そのすべてに対応しきれない場合も少なくない。
- ・課題5：従来方式では静的コード解析ツールでチェックするルールは主に過去の不具合等を参考に決めているため、チェックの視点が若干偏ったものとなり、本来、見直したほうが良い視点がそのままになる。
- ・課題6：静的コード解析ツールによる品質チェックに慣れてくると、開発者は解析結果レポートを単純な作業リストとして見るようになってしまう。例えば、変数の型宣言の違い等に対して、使用している型について妥当性の検討をせずに機械的にキャストする等、指摘に対して表面的な回避を行い本質的なコード修正の必要性検討がおろそかになるといったケースが見られた。このように開発者が静的コード解析ツールの指摘への対応を単純作業と見なしてしまった場合、指摘事項の意味を十分に吟味せずに単なるコードデバッグの感覚でワーニングをとることのみに集中してしまいがちになり、本質的なコードの誤りを見落としがちになってしまう。

3 コード品質向上に向けたアプローチ

3.1 従来方式の課題の解決の方向性

弊社が開発するソフトウェア製品の特徴でもある複雑な製品系列を前提とした開発において、流用部分の価値向上やコード品質の向上のためには、まず、上記の課題をいかに解決していくかがその出発点になる。社内で従来活動に関わってきた関係者等と議論を重ね、それぞれの課題を精査した結果、課題解決に向けて以下のようなポイントが浮

かび上がった。

① コード解析で採用するルールセットについて

静的コード解析ツールの機能には、特定のチェックルールについては指摘されないように絞り込むチェックルールを選択しフィルタの作成を行い、静的コード解析ツールの解析実行時にそのフィルタを適用することにより、ツール自体で問題箇所の指摘を絞り込むという方法もある。しかしながら採用するルールセット次第では依然膨大な指摘になったり、逆に必要な指摘がされなかったり、課題3や課題4にもあるような問題が発生してしまう。また課題5のように静的コード解析ツールのルールの偏り等の影響で、流用部のコードに求められる品質特性が十分に検出されないといった問題も生じてしまう。このため我々は、静的コード解析ツールではコード品質面での重大な問題に着目して検出効果のある最適なルールセットを準備することがポイントとなると考えた。一方で、静的コード解析ツールによってチェックが出来る項目は主として記述面のチェックや文法面でのチェックが中心となり、コードの意味面の誤り検出は難しいという側面がある。そこで、コードのロジック等のチェックについては経験者による目視レビュー等も併用していく必要があると考えた。

② コード品質チェックのタイミングについて

コードはコンパイル・ビルドされた時点では実行可能な形態となるため、一般的な開発者は単体テストによる動作確認を優先する傾向が強い。そして実際に動作させてうまくいった場合に、課題2のように改めてそのコードを見直すといった作業に抵抗感を抱く開発者は少ない。このため、我々はコードの品質チェックや改善はビルド前のコーディング作業の途中段階で行うのが、開発者が実施するプロセスとしては理想的であると考えた。一方で静的コード解析ツールはコンパイラ等と同様に構文解析を行うため、ある程度コンパイル可能なコードが対象となる場合が多い。こうした制約等を考慮すると、コードのコンパイル直前にまず静的コード解析ツールを用いてコードを解析し、そこでの指摘事項に対応した後に、経験者による目視レビューを行うことで、段階的にコードの品質を確認していく方法が有効であると考えられる。このようにコード解析ツールによるチェックと目視レビューを分離することで、軽微な文法ミスや記述ミスはツールによるチェックであらかじめ除去することが出来るため、目視レビューはプログラムの意味的な側面に集中することが出来、品質のチェック効率の面での向上が期待出来る。

③ 開発者の主体性について

従来、弊社が採用してきたサブリーダが解析を行う方式の場合、開発者は解析結果レポートを受け取り修正するという、ある意味で受身の立場に立つことになる。しかしこ

の場合、課題6のように「個々の開発者自身がコードに対する品質意識を積極的に持ち自身が作成したコードの品質を良くする」というよりは「指示されたから直す」という色合いが強くなってしまいます。結果的になぜコード解析をするか、あるいはなぜ目視レビューをするかといった品質を改善することへ意義を見失うことにつながりやすく、指摘された事項の修正漏れや何度も同じ指摘を繰り返し受けるといった状況に陥りやすい。そして最終的には指摘に対する対応の度合いも大きく変わってきてしまう。

このため、我々はコードを作成した個々の開発者が自らコードの品質を良くするという意識を高めるために、開発者自身によるコード解析やレビューへの積極的な参画を後押しする仕組みづくりが重要であると考えた。

④ 事前のコーディングルールの教育について

弊社ではそれぞれの製品群ごとに幾つかのコーディングルールが用意されてきた。しかしながら、これらのコーディングルールはインデントの付け方やコメントの書き方等、主にコードの記述スタイルを規定する事項が中心となっており、それらが遵守されなくとも直接的なバグとして現象が発生することも少ないので、ソフトウェアの品質にどのような影響が出るかといった関連性が個々の開発者には十分に伝わり切れないものとなっていた。それが弊害となり、実際の開発の中ではこれらのコーディングルールはあくまでも参考情報として位置付けられており、どこまで遵守すべきかといった判断は個々の開発者に任されていた。また、静的コード解析ツールのチェックルールとコーディングルールの対応付けも十分に考慮されておらず、弊社独自の記述スタイルが中心のコーディングルール違反が静的コード解析ツールに検知されることも少なかった。この結果としてコーディングルールの遵守率が下ってしまう傾向にあった。更に、このような形骸化が進むと開発者はコーディングルール策定の意図も十分に理解しないままコーディングすることになり、コーディングルールの遵守は更に期待出来なくなってしまう。そうした結果、課題1のような玉石混淆のワーニングを受ける状況に陥るケースも出てくる。これらを未然に防ぎコード解析や目視レビューでの実装品質の向上をより効果的なものとするためには、静的コード解析ツールで利用されるチェックルールとコード作成時のコーディングルールが明確に対応付けられ、かつ事前に開発者の頭に入った状態でコーディングが行われる必要があると考えた。

3.2 コーディング作法を用いた品質向上への取り組み

従来方式の課題解決に向けた上記の方向性を考慮し、弊

社ではIPA/SECが策定したコーディング作法(ESCR^{*1})を取り込み、新たなコード品質改善に向けた取り組みを実施した。新しい取り組みでは以下の3点を特徴としている。

- 施策1: コード品質向上プロセスの見直し
- 施策2: コード解析のルールセットの見直し
- 施策3: コーディング作法の事前教育

(1) コード品質向上プロセスの見直し

コーディング以降の開発工程で品質向上を図る作業としては、コード解析、目視レビュー、動作テスト等が考えられる。実際の開発においてこれらの品質向上作業に割くことの出来る時間は限られており、これらをどのように位置付けて実践していくかが大きなポイントである。そこで社内の有識者で議論し、「コード品質改善点の検出」、「コード構造欠陥の検出」、「ロジックミスの検出」、「過去機種開発でのノウハウ」、「機能要求との整合確認」といった各確認視点で、それぞれの品質向上作業に対するの優劣評価を表1に示すような形でまとめた。

社内有識者による検討を重ねた結果、表1に見るように静的コード解析ツールの場合、「コード品質改善点の検出」や「コード構造欠陥の検出」に関する能力は高いものの、「ロジックミスの検出」等コードの意味的な側面での誤り検出には限界があると我々の間では結論付けた。こうした検討を元に、コーディング以降の品質向上作業の位置付けや狙い、役割を整理し、それぞれどのタイミングでどの程度まで実施するかといった品質向上プロセスの見直しを行った。この結果、コード解析はツールの特徴を生かして、早期にコード全体を網羅する形でコードの「品質改善点の検出」や「コード構造欠陥の検出」を目的に実施すると位置付けた。その後、コードの意味面等を含めた「ロジックミスの検出」や「過去機種開発でのノウハウ」を加味した

表1 コード品質向上手法の優劣評価結果

	コード解析	目視レビュー	動作テスト
品質改善点の検出	◎ * ESCR 対応	△	○
構造欠陥の検出*複雑度等	◎	○	△
ロジックミスの検出	△	◎	○
過去機種開発でのノウハウ	△	◎	○
機能要求の整合の確認	×	○	◎
品質保証	○	△	◎

*1 ESCR: Embedded System development Coding Reference, 組込みソフトウェア開発向けコーディング作法ガイド

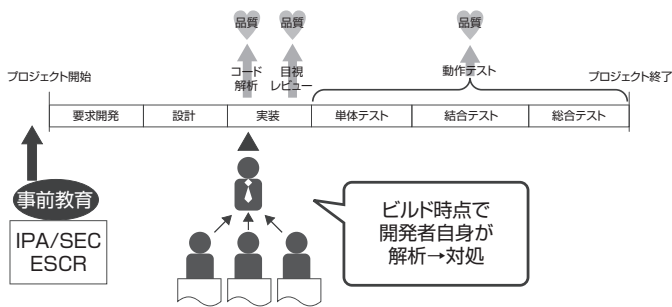


図3 プロセス見直し後のコード品質チェック

チェックは主に経験者による目視レビューで対応することとし、更に機能要求との整合性や品質を保証するための確認については動作テストによって実施するといった3段階からなる品質チェックのプロセスを採用した(図3)。

この3段階の品質チェックプロセスの中であってコード解析は初期段階での品質スクリーニングを行う役割も併せて持たせることで、それ以降の品質向上をより円滑に進める効果を期待している。このために施策2で示すように静的コード解析ツールで採用するルールセットの見直しも併せて行った。

(2) コード解析のルールセット見直し

コード品質を向上させるためには、どのようなコーディングルールにするかを定め、それをどのように実際のコーディングの際に徹底して実践していくかが極めて重要である。また同時に実際に作成されたコードの品質を管理するためには、当初決められたコーディングルールが守られているかどうかといった点から確認し、適切な品質向上へのフォローをすることで、コーディングルールの遵守率が向上し、結果的にコード品質の向上が実現出来る。このため、弊社では開発で採用するコーディングルールの検討と静的コード解析ツールのルールセットの検討を並行して進め、相互に対応が可能になるようにした。具体的な方法として、まず、静的コード解析ツールによるコード解析の目的を系列開発における流用部分の品質向上に着目した。その中でもとくに流用ソフトウェアという意味では、コードの信頼性、保守性を確実にしていくことが最重要である。コードの持つ特定の品質特性に着目して品質を改善するためには、利用するコーディングルールにおいても品質特性を意識しておく必要がある。IPA/SECが策定したコーディング作法では品質特性を意識して作法やコーディングルールがカテゴライズされており、弊社が重視する保守性のルールを検討する上で参考になると考えた。このため、我々はESCRの信頼性、保守性に関係する選択指針等も参考にして、静的コード解析ツールで利用するルールセットの選択

を行った。表2は従来方式で採用していた静的コード解析ツールのルール数と上記によって新たに用意したルール数を比較したものである。表3に見るように、新たに用意したルールセットは、保守性に関するチェックルールを多く加えることで、コア資産となるコードの保守性を静的コード解析ツールで効率的にチェックすることを狙っている。また、これらのチェックルールはあらかじめ開発部門のコーディングルールにも反映させることにより、コード作成時に開発者が意識しながらコードを作成出来るようにした。具体的なルールセットの見直しでは、まず、弊社が扱う民生用光学機器ソフトウェアに求められる品質を考慮し、信頼性、保守性といった品質特性ごとに品質レベルを設定し、それを実現するためのルールセットをESCRを参考に選び出した。そして、これらについて弊社が利用している静的コード解析ツールのチェックルールとも対応をとりながらルールセットの全体数や品質特性ごとのバランス等を取る形で最終的に採用するルールセットを決定した。表3はこのようにして今回用意したコーディングルールを抜粋したものである。例えば信頼性のR2.5.1のよう

表2 静的コード解析ツールのルール数の比較

	信頼性	保守性	移植性	効率性	合計
従来開発	47	53	6	0	106
今回開発	71	117	6	0	194

表3 採用したコーディングルール例

SECルール番号	SECルール
R2.5.1	情報損失を起こす可能性のあるデータ型への代入(= 演算, 関数呼出しの実引数渡し, 関数復帰)や演算を行う場合は、問題が無いことを確認する。機械的にキャストするのではなく、使用するデータ型の変更を検討する。
M1.4.2	式中に優先順位の異なる複数の2項演算子を含む場合には、優先順位を明示するための括弧をつける。ただし、四則演算に関しては、括弧を省略しても良い。
M1.8.1	論理演算子 && または の右側のオペランドには、副作用があってはならない。【MISRA 12.4】
M3.1.2	goto 文は、それを使う明確な理由があるときのみ使用する(多重ループを抜ける場合とエラー処理に分岐する場合等)。
M3.1.4	(1) switch 文の case 節, default 節は、必ず break 文で終了させる。 (2) switch 文の case 節, default 節を break 文で終了させない場合は、"FALL THROUGH" コメントを挿入する。

※下線文字列・・・弊社が独自にESCRの表現をカスタマイズしているもの

に変数のキャスト等の利用に制約を与えることで、データ型が変換されることを未然に防ぎ課題6に示したような表面的なコード修正を防止することを狙っている。また保守性のM1.4.2, M1.8.1等は、過去に弊社内での指摘が多くあった演算順序の表記に関する保守性面の問題について、記述方法を統一することで問題を未然に回避することを狙っている。このようなルールを規定し導入することで信頼性や保守性向上に効果が期待出来るものである。

(3) コーディング作法の事前教育

コードの品質向上を進めるためには、個々の開発者がコード作成段階で品質を意識したコーディングをすることが重要である。我々は新たな試みとしてIPA/SECのコーディング作法を参考に、とくに信頼性、保守性に関するコーディングルールを決め、開発者がそれらを意識してコーディングするようにした。この一環として、今回の取り組みでは、IPA/SECの協力を得て、まず対象者全員にコーディング作法教育を実施した。コーディング作法の教育については、IPA/SECが一般企業向けに提供している教育コースをベースに行った。教育対象は今回の取り組みを意欲して行うプロジェクトメンバを中心に数十名程度を対象に、約4時間のコースで実施した。参考として表4にコーディング作法教育のタイムテーブルを示す。

IPA/SECのコーディング作法は、組込みソフトウェアの実装品質を向上させるためにISO/IEC 9126で定められた品質特性を考慮し、信頼性、保守性、移植性、効率性の各側面の品質を向上させるためのコーディングルールが整理されている。

教育ではコードが持つべきこれらの品質特性について解説を行った後、それぞれの品質特性を実現するためにコード作成時点でどのような事項を考慮すべきかについて作法

及び具体的なコードの書き方についてコーディングルールの詳細等を解説した。またこれらの基本的な作法やルールの理解を促進するために一部、これらを活用したコードレビューの演習等を交えて教育を行った。具体的にはまず、医療機器や通信機器、運輸機器等実際の組込み分野の製品を題材にして、その製品に求められる品質特性を分析し、幾つかのコーディングルールを検討し選択する演習を実施した。次にこのようにして選択したコーディングルールを用いることを想定し、仮想のプロジェクト条件下で実際のコードを提示し、その目視レビューを実施する演習をグループで行った。その上で、弊社のコーディングルールに関する考え方や特徴、具体的なルール等について説明を行い、開発担当者のコーディングルールの理解度を上げ、実務での実践に向けた意識の向上を図った。

4 評価

4.1 先行トライアルプロジェクト

前述の「コード品質向上に向けたアプローチ」を確実に社内を展開するために、今回は第1弾として先行プロジェクトを選定し、そのプロジェクトでこのアプローチの実践を試みた。先行プロジェクトはある民生用光学機器のバリエーション開発を行うプロジェクトで、数十名の開発者が半年程度をかけて開発を進めるプロジェクトであった。今回の先行トライアルでは「コーディング作法の事前教育」「静的コード解析ツールのルールセット見直し」「コード品質チェックプロセスの見直し」の3点を重点的に実施した。ここでは「コーディング作法事前教育」の効果と、「ルールセットの見直し」「コード品質チェックプロセスの見直し」の結果としてのコード品質の改善の様子について実際のデータを基に紹介していく。今回の取り組みの効果を評価するにあたり、従来方式のコード品質の確保手段を取った前機種開発のプロジェクトの開発データと、今回方式のコード品質の確保手段を取った先行トライアルプロジェクトの開発データを比較する。

4.2 コーディング作法教育の効果

図4はコーディング作法教育実施後に行ったアンケート調査の結果を示したものである。教育はトライアルプロジェクトが開発に入る前に行ったが、図4に見るように担当者の意識は事前教育を行うことでコーディングルールの遵守について理解度がかなり高くなっていることが分かる。経済産業省が実施している組込みソフトウェア開発に関する産業実態調査によると、コーディング規約を導入して開発している企業は数%にとどまっていることを考える

表4 コーディング作法教育のタイムテーブル

IPA/SEC より		
講義 1	コーディング作法ガイド概説 コーディング規約の重要性と品質特性に応じたコーディング規約の分類等の講義	1.0H
演習 1	コーディング規約の作成演習 製品特性に応じたコーディングルールの選択演習	1.0H
演習 2	ソースコード可視化演習 サンプルソースコードを自分達で決めたメトリクスを用いたソースコード品質の計測実習	1.5H
WG より		
講義 2	プロジェクト内コーディングルールの解説 静的コード解析ツールの活用改善と新たに追加したコーディングルールの具体的な内容についての講義	0.5H

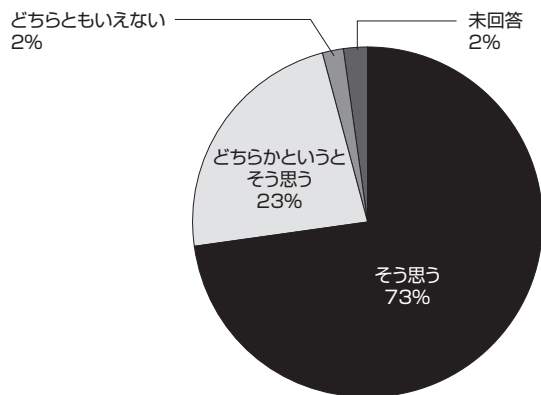


図4 コーディング規約の重要性認識率

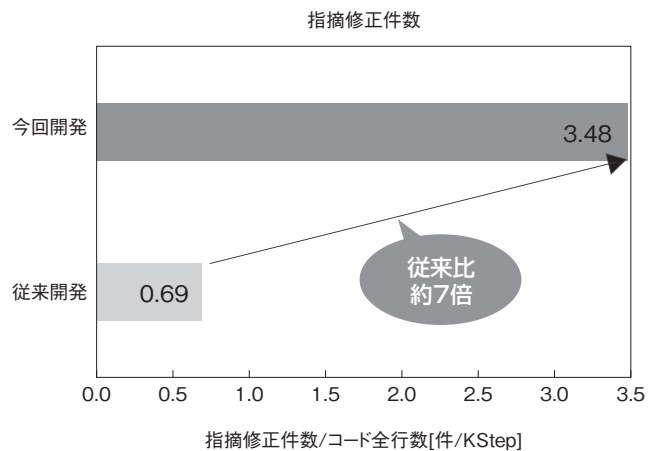


図5 コード解析の指摘修正件数の変化

と、今回実施した教育によって開発担当者への品質の重要性を大きく意識付けるきっかけになったと考えることが出来る。

実際に教育を受講した開発者の意見でも、「他で起きているソフトウェアの市場障害でも、関心を持ってニュースやネット上の情報サイトを見るようになった」「自分の作るコードの品質を意識するようになった」「コードを書く際に次開発への影響を意識するようになった」「ちゃんとルールの意味を感じて実装出来る様になって嬉しい」等が寄せられた。

4.3 コード品質の変化

次に今回のアプローチによってコード品質がどのように変化したかをデータから評価していく。

(1) コード品質チェックプロセスの変更の影響

今回のアプローチではコードの品質チェックのプロセスを結合テスト時点に行っていたのをビルド時点で行うように大きく変更した。

このプロセス変更に伴う一連の改善活動により、開発者は静的コード解析ツールの指摘に対する対応が格段に取りやすくなり、図5に示すように従来開発に比べて約7倍と大幅に修正対応することが可能になった。

この結果、図6に見るように開発の過程での不具合検出状況も大きく変化した。

図6は開発者が行う品質確認工程での不具合指摘状況を従来開発と今回開発で比較したものである。各工程において指摘された不具合指摘件数を比較すると、今回方式の方が従来方式に比べて少なくなっており、品質面での大幅な向上が確認出来る。今回方式での品質面での優位性が出た結果の考察を以下に述べる。

まず、静的コード解析ツールによるコードチェックのタ

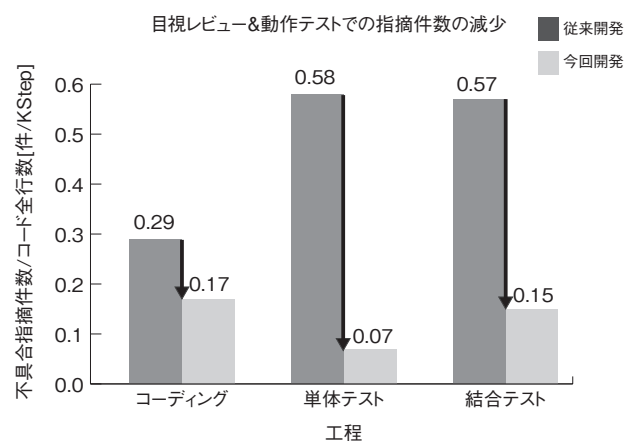


図6 工程別不具合指摘状況の変化

イメージを変えた結果、従来は単体テストで検出していた不具合の一部が、静的コード解析ツールによるワーニングとしてコーディング工程で開発者によって検出されるようになった。開発者がコーディング後のコード見直しの一環として静的コード解析ツールのワーニングに対応することが可能となったことで、以前は単体テストまで検出出来ずにいた不具合の除去効果が得られている。また、静的コード解析ツールに指摘された経験があるワーニングについては学習が済んでいる開発者が、コーディングをする際に同じ指摘を受けないようにと自身で気を付ける習慣もついてきており、開発者は静的コード解析ツールに指摘されるようなコードは書かなくなってきているので、不具合の混入予防にもつながっている。

また、同時に、従来は静的コード解析ツールによるチェックと目視レビューがオーバーラップしていたため、経験者による不具合指摘が明確に区分出来ていなかったが、今回の方式では静的コード解析ツールによるチェックを早期段

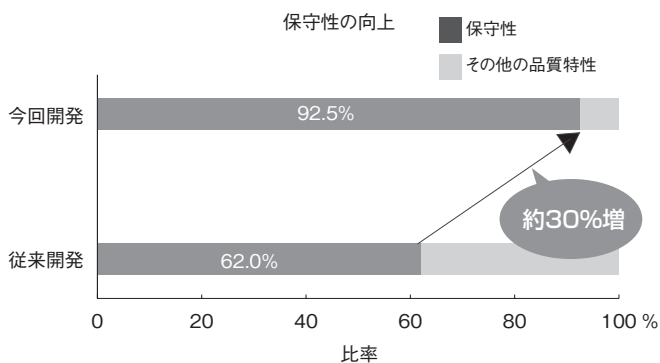


図7 静的コード解析ツール結果の推移

階に移行したために、経験者による目視レビューが明確に分離出来た。この結果、経験者による目視レビューでどの程度の指摘がされているかが明確になり、コードの意味的な側面等重要な指摘事項への対応がやりやすくなった等の効果が得られている。更に、テスターによるテストフェーズについては、今回方式の方が検出される不具合数は約75%少なくなっている。これは、コード解析、目視レビュー、動作テストという3段階の品質チェックメカニズムがそれぞれのレベルで効果的に不具合の検出を可能とした結果、軽微な不具合のテストフェーズへの持越しが少なくなった効果であるとも考えられる。

(2) 静的コード解析ツールによる指摘内容の変化

今回のトライアルでは、静的コード解析ツールでチェックするチェックルールの選択も併せて行った。図7はコード解析で指摘されたワーニングの内訳を示したものである。今回の方式を適用した先行トライアルプロジェクトと従来開発では静的コード解析ツールの指摘の内訳は大きく異なっていることが分かる。従来方式ではバグの傾向から静的コード解析ツールのチェックルールを選択していたので、保守性以外の指摘の割合も40%程度占めている。これに対し、今回の方式ではあらかじめ保守性に重点を置いたルールを選択したため、表3に例として示したM1.4.2, M1.8.1等演算の記述方法に関するワーニングが多く指摘され、静的コード解析ツールによって保守性観点での指摘が90%以上を占めた。そのため開発者は保守性の観点で重点的に自身のコードの見直しをすることが可能となった。今回の開発のようにあらかじめチェックルールをコーディング作法に準拠して保守性等の側面に重点を置いて指摘不具合をコントロールした場合には、コード品質の保守性等特定の側面の向上を実現するため必要な指摘項目に着目し、確実に狙った品質特性面を向上させることが可能となると考えられる。

5 まとめ

本稿では流用開発を考慮したコードの資産価値を向上させることを目的とした弊社の活動について紹介した。

今回の取り組みでは、静的コード解析ツールによるコード解析と目視によるレビューを開発プロセス上で効果的に連携させ、コードの保守性を積極的にチェックし見直していく仕組みを中心とした。またこの活動に併せて弊社内のコーディング規約の見直しや開発者向けのコーディング規約に関する教育を行うことで開発者の意識改善等も進めた。これらの活動によりトライアルプロジェクトではコードの保守性が大幅に向上したと考えられ、コードの資産価値向上に向けた第一歩が踏み出せたと考えている。また、本文中では触れなかったが、こうした一連の活動を行うことにより、品質チェックに要するコスト面の軽減も認められ、開発のトータルコスト低減にも寄与出来たと考えている。今回の取り組みを端緒として、今後も弊社内におけるソフトウェア品質の向上に引き続き取り組む所存である。

6 謝辞

ESCRと静的コード解析ツールの徹底活用を検討するため、勉強会を開きたいといった弊社にご賛同頂き、ご助力を賜りましたIPA/SECの皆様と株式会社富士通ソフトウェアテクノロジーズ 佐々木孝次様、様々検討して運用まで持って行ってくれたWGメンバ始め、日々現場で頑張っておられる開発者の皆様に感謝申し上げます。

参考文献

- [ESCR2006] IPA/SEC：組込みソフトウェア開発向け コーディング作法ガイド [C言語版]、翔泳社、2006
- [ESMR2006] IPA/SEC：組込みソフトウェア開発向け プロジェクトマネジメントガイド [計画書編]、翔泳社、2006
- [ESPR2007] IPA/SEC：組込みソフトウェア開発向け 開発プロセスガイド、翔泳社、2007
- [ESQR2008] IPA/SEC：組込みソフトウェア開発向け 品質作りこみガイド、翔泳社、2008
- [ISTQB2007] International Software Testing Qualifications Board：ISTQB テスト技術者資格制度 Foundation Level シラバス 日本語版 Ver.1.1.0、2007
- [MCCONNELL2005] スティーブ マコネル：CODE COMPLETE ～完全なプログラミングを目指して～第2版、日経BPソフトプレス、2005
- [MISRA2006] MISRA - C 研究会：組込み開発者における MISRA-C:2004-C 言語利用の高信頼化ガイド、日本規格協会、2006
- [SPINELLIS2007] Diomidis Spinellis：Code Quality ～コードリーディングによる比機能特性の識別技法～、毎日コミュニケーションズ、2007
- [小池 2008] 小池輝明、藤墳真也：静的解析のサービス化による静的解析の開発プロセスへの定着、SQiP シンポジウム 2008 経験論文、2008
- [データ白書 2008] IPA/SEC：ソフトウェア開発データ白書 2008 ～IT企業 2056プロジェクト 定量データが語る開発の実体と傾向～、日経BP社、2008

ITプロジェクトを「見える化」する チェックシート

SEC 企画グループ 研究員
樋口 登

SEC におけるプロジェクト「見える化」の取り組みは、2005年にスタートし、現在も継続に発展している。

この活動の中では様々な「見える化」手法が検討され、その一部は実際のプロジェクトに適用してブラッシュアップした後で、ITシステム開発現場にもすぐに使えるようにまとめ、書籍等で公開してきた。本稿では、「見える化」手法の中でも、とくにマネジメントのエッセンスが凝縮されているチェックシートを中心に、これまでの取り組みと活用のポイントを紹介する。

1. SEC における「見える化」の取り組み

SEC でプロジェクトを「見える化」するための準備活動が始まったのは、2004年10月にSECが設立されてから半年以上が過ぎてからであった。

この活動において、プロジェクトを「見える化」とするのは、プロジェクトマネジメントの問題の有無をはっきりと把握することである。もし、プロジェクトマネジメントに問題があれば、その問題点をいち早く発見し、出来るだけ早く対応することで、プロジェクトが失敗することを防ぐ。

プロジェクト「見える化」のための最初の活動は、失敗プロジェクト（品質が悪い、予定費用を超過、納期遅延等により、当初の目的／目標を果たせなかったプロジェクト）の事例収集とその分析であった。

SECの有志が失敗プロジェクトの事例を分析して、プロジェクトを失敗させないためには何に気を付ければよかったのかを抽出し、チェックシートとしてまとめた。このチェックシートの妥当性を検討・検証するために、ITプロジェクトの火消し（失敗した、あるいは失敗しそうなプロジェクトを立て直すこと）に長年携わってこられた方々にもご参加いただき、2005年8月に「準備会」が発足した。

準備会では、当時、経済産業省の下でソフトウェアエンジニアリング技術研究組合(COSE^{※1})が実施したプロジェ

クト(SECでは「先進ソフトウェア開発プロジェクト」と称していた)に対してチェックシートを適用し、その有効性を確認すると共に、チェック項目の内容や順番、チェックシートの構成等を見直した[SEC2007]。

その後、2005年11月にはメンバを更に増やして、経済産業省の部会として「プロジェクト見える化部会」が正式に発足し、プロジェクトの「見える化」についての検討が本格化した。検討はプロジェクトの下流工程(ソフトウェア結合以降の開発プロセスを想定)を対象に行われ、「見える化」のための3つのアプローチが考案された。それぞれ、①俯瞰図、チェックシート、事例集による「定性的アプローチ」、②測定項目リスト、エンピリカルソフトウェアエンジニアリング^{※2}を実現するEPM^{※3}ツールによる

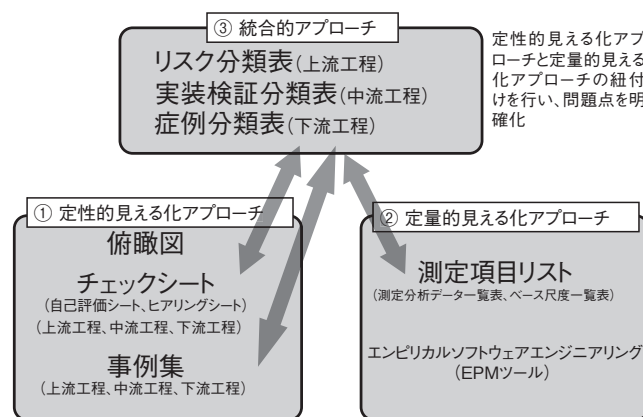


図1 ITプロジェクト「見える化」のアプローチ

※1 COSE : Consortium for Software Engineering

※2 エンピリカルソフトウェアエンジニアリング, Empirical Software Engineering

エンピリカルは“実証的”と訳される。プロジェクトから実際に観測、測定されたデータに基づいて分析を行い、分析結果をプロジェクトにフィードバックしてプロジェクトを改善する。

※3 EPM : Empirical Project Monitor

EPMは、文部科学省のEASE (Empirical Approach to Software Engineering) プロジェクトで開発された (<http://www.empirical.jp/>)。

IPAは、EPMをソフトウェア開発現場で利用しやすいように機能強化を行い、EPMツールとして提供している。 <http://sec.ipa.go.jp/tool/emp.html>

「定量的アプローチ」、③症例分類表による「統合的アプローチ」である(図1)。この検討結果は、SEC BOOKS『ITプロジェクトの「見える化」～下流工程編～』[SEC見える化-1]にまとめられた。

同様に、2006年度には上流工程[SEC見える化-2]、2007年度には中流工程における「見える化」手法[SEC見える化-3]をまとめ、更に、上流から下流までを総括した[SEC見える化-4]。

2007年度からは、「見える化」手法の普及を目指して、EPMツールを実際に使っていただくためのEPMツール検証プロジェクト(<http://sec.ipa.go.jp/tool/epm.html>)の募集を開始し、また、各地で『プロジェクト「見える化」セミナー』を開催した。この活動は現在も継続している。

2008年度からは体制が変わり、IPA/SECの「プロジェクト見える化WG」となったが、活動範囲は更に拡大し、日本からのオフショア先に「見える化」手法を展開することを考慮して、海外でも“MIERUKA”として論文発表を行っている。

2. チェックシート

チェックシートはプロジェクトを「見える化」するための1つの手法であるが、単なるチェック項目の一覧ではなく、前述のように、プロジェクトの失敗を防ぐためのエッセンスが詰まった資料であり、以下に示す特徴を備えている。

2.1 工程別チェックシート

前述のプロジェクト「見える化」手法をまとめたSEC BOOKS『ITプロジェクトの「見える化』』は、上流工程編、中流工程編、下流工程編に分かれており、それぞれにチェックシートが付いている。これは、各工程の特性によって「見える化」するポイントが異なるためである。すなわち、上流ではリスク、中流では要件の実装状況、下流では問題の予兆を「見える化」することが重要であり、そのためにチェックシートも工程ごとに異なっている。

2.2 自己評価と第三者評価

チェックシートは工程ごとに用意されるだけでなく、プロジェクト・マネージャ自身が自分でチェックする「自己評価シート」とPMO※4等、社内のベテランがプロジェクト・マネージャにヒアリングして評価する「ヒアリングシ

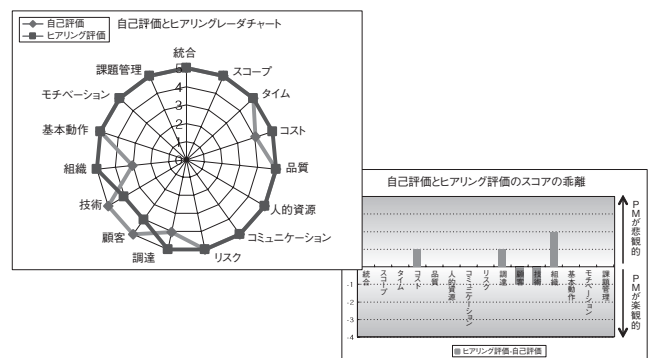


図2 チェックシートの分析結果

ト」に分かれている。

自己評価シートとヒアリングシートの評価結果を比較することで、プロジェクトマネジメント上の弱点をより明確に把握することが出来る(図2)。

2.3 プロジェクト・マネージャの気づきとヒント

自己評価シートは、プロジェクト・マネージャが自ら実践しているマネジメントについて自分で評価するものであり、以下の3段階の評価をつける(表1)。

表1 自己評価シートの3段階評価

1	チェック項目のマネジメント方法を知らない。あるいは、理解しているが出来ていない
2	チェック項目のマネジメントとして何をすべきかは理解しているが、事情によりうまく出来ていない
3	チェック項目のマネジメントを理解しており、ほぼ出来ている

自己評価シートには、評価点をつける場合の「評価基準」がチェック項目ごとに記載されており、プロジェクト・マネージャはどのレベルまで求められているのかを把握することが出来る。

また、自己評価シートには「マネジメントにおけるヒント」が記載されており、どのようなことに注意してマネジメントをすべきかを把握することが出来る。

更に、該当するチェック項目の評価が低かった場合の「対策案」も記載されており、プロジェクト・マネージャは施策の検討に対策案を活用することが出来る。

2.4 カスタマイズ

自己評価シートは、業務システムを受注した企業のプロ

※4 PMO : Project Management Office

プロジェクト・マネージャを想定して作成されている。想定と異なる開発プロジェクトの場合には、チェック項目の内容を読み替えることが必要になる場合がある。

また、組織やプロジェクトにとって、より重要なマネジメント項目がある場合には、対象とするマネジメントに関するチェック項目を追加すべきであり、逆に関係の無いチェック項目は削除する等、必要に応じてチェック項目を見直し、運用しやすいように変更して欲しい。

更に、同じ自己評価シートによるチェックを繰り返し実施している場合、マネジメントが定着した項目については削除し、新たにレベルアップすべきマネジメント項目を追加する。これにより、自己評価シートを常に現状のマネジメントレベルに合わせて改良することが出来、プロジェクト・マネージャによるプロジェクトマネジメントの改善を加速させることが出来る。

3. チェックシートの活用

以下にチェックシートの活用事例を紹介する。運用方法やチェック項目のカスタマイズ等、チェックシートによる「見える化」を行う場合の参考にして欲しい。

3.1 フィードバック重視のヒアリング

COSE 先進プロジェクトへのチェックシート適用では、ヒアリング結果のフィードバックに重点を置いた。

このプロジェクトは、6社がそれぞれの分担分を各社に持ち帰って開発を行っており、ヒアリングは各社の開発現場にヒアリングチームが出向いて実施した。

ヒアリングチームはプロジェクト見える化部会メンバー4、5人で構成されたが、COSE プロジェクトメンバーとは全く面識が無く、各社のプロジェクトについては、プロジェクト概要記述用紙（自己評価シート、ヒアリングシートと共にSECから提供されている）と自己評価シートをプロジェクト・マネージャから送付いただき、事前にメンバーに配布した。

ヒアリング終了後、別施設の会議室等でヒアリングチームメンバーの評価結果を持ち寄って評価会議を開催した。ヒアリングチームメンバーの評価点の平均を評価結果とし、評価結果が悪かった項目やヒアリングを通じて気になった点をまとめ、それぞれに対策案を付して評価報告にまとめた。

ヒアリングを実施してから1週間程度で報告会を開催し、ヒアリングチームの代表からプロジェクト・マネージャに評価結果のフィードバックを行った。

3.2 チェック項目と運用の工夫

株式会社エヌアイデイ様（東京都新宿区）では、業務維持管理プロジェクトに対して、チェック項目数を少なくしたチェックシートを適用している。

1回のチェックを20分以内で出来るようにして、チェックする頻度を高める運用にしている。これにより、慣れによる見落としが少なくなり、改善意識が芽生えたと報告されている。

チェック項目は独自に検討した内容で、年間・月間・定型作業等に区分して、新規要員にも全体像がつかみやすいよう工夫している。

3.3 一斉チェックと教育への利用

株式会社ジェイマックスソフト様（新潟県長岡市）は、内部監査の一環として上流工程用チェックシートによる「見える化」を行っている。2007年度から取り組まれ、自己評価シートとヒアリングシートの差分分析を中心として、データも着実に蓄積されている。

また、新たにプロジェクト・マネージャになった人に対して自己評価シートを用いた研修を行っている。自己評価シートがプロジェクト・マネージャ自身に気づきを与えることが出来るという特徴を生かした利用方法で、チェックシート利用の定着化と新任プロジェクト・マネージャのマネジメント力の向上が期待出来る。

4. 「見える化」セミナー開催と検証プロジェクト

プロジェクト「見える化」手法を多くの方にご理解いただき実践してもらうことによって、少しでも失敗プロジェクトを減らすことが出来ればとの思いを込め、2007年度からEPMツール検証プロジェクトとプロジェクト「見える化」セミナーを開催している。

4.1 EPM ツール検証プロジェクト

EPM ツール検証プロジェクトでは、ご応募いただいた

※5 Magi：EASEプロジェクトで開発された見積りツールで、協調フィルタリングという技術を利用している。<http://se.naist.jp/magi/>

※6 CCFinderX：ソースコード中のコードクローン（類似部分）を検出するツールで、2005年度 IPA 未踏ソフトウェア創造事業で独立行政法人産業技術総合研究所の神谷年洋氏が開発したもの。<http://www.ccfinder.net/ccfinderx-j.html>

方に、定量的見える化アプローチを支援する EPM ツール、Magi^{※5}、CCFinderX^{※6}をセットにして提供している。これらのツールを使うことで、エンピリカルソフトウェアエンジニアリングを実践（実感）し、プロジェクトの「見える化」を推進していただきたい。

EPM ツールは今までに 70 を超える組織からご応募いただいております。ご要望等も受け付けています。いただいたご意見、ご要望は、EPM ツールの機能強化の検討に使わせていただいております。

4.2 プロジェクト「見える化」セミナー

2007 年 11 月に福岡で開催したのを最初に、各地でプロジェクト「見える化」手法の紹介と「見える化」の実現についてのセミナーを開催している。

セミナーでは、受講いただいた方にアンケートをお願いしている。2009 年度のアンケート（2010 年 1 月末現在）で最も「見える化」したい工程は上流工程、次いで中流工

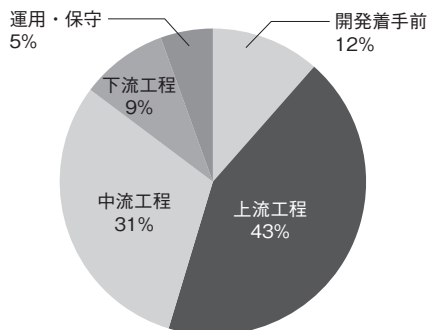


図 3 最も「見える化」したい工程

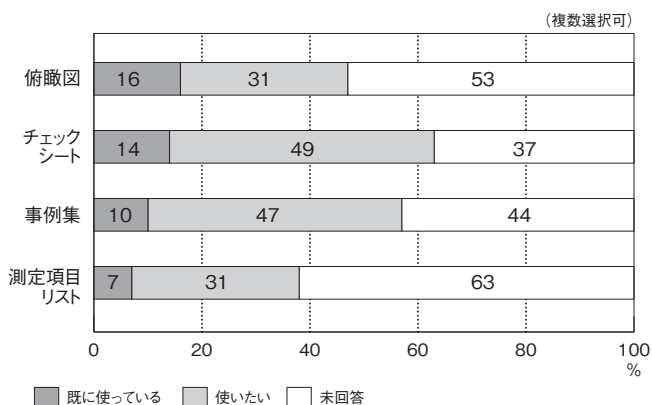


図 4 使いたい「見える化」手法

程で、この 2 つで全体の 3/4 を占めている（図 3）。

また、「見える化」手法の中で「既に使っている」、「使いたい」という回答の割合の合計が最も高かったものがチェックシート、次いで事例集であり、ともに参加者の約半数から使いたいとの回答をいただいている（図 4）。

セミナーや SEC BOOKS で紹介している手法やツールは、すぐに使えるものであり、これらを利用してプロジェクトの「見える化」を推進していただきたい。

ここで紹介したものは、アンケートのほんの一部だけであるが、他の結果やアンケートでいただいたご意見は、今後のセミナー開催の検討に使わせていただく。

5. おわりに

プロジェクトの「見える化」に限らず、チェックシートを使う場合、チェックが形骸化してしまえば効果が上がらない。またチェック項目が多すぎるとモチベーションが低下し、正しくチェック出来なくなってしまう等の問題が起こる。

チェックシートは組織やプロジェクト、環境変化等に合わせてチェック項目を見直し、チェックのタイミングを工夫する等、運用に注意しながら活用していただきたい。

EPM ツール検証プロジェクトや SEC セミナーを通じてご意見をお寄せいただいた方々に改めて感謝の意を表したい。いただいたご意見を参考に、今後もチェックシートを含めて、プロジェクトを「見える化」する手法の普及を促進していきたい。失敗プロジェクトを 1 つでも少なくするために役に立てば光栄である。

なお、プロジェクト「見える化」に関する情報は SEC Web サイトや EPM ツール^{※7}のページを参照していただきたい。

参考文献

[SEC2007] IPA/SEC 編：SEC BOOKS ソフトウェアエンジニアリングの実践～先進ソフトウェア開発プロジェクトの記録～，翔泳社，2007
 [SEC 見える化-1] IPA/SEC 編：SEC BOOKS IT プロジェクトの「見える化」～下流工程編～，日経 BP 社，2006
 [SEC 見える化-2] IPA/SEC 編：SEC BOOKS IT プロジェクトの「見える化」～上流工程編～，日経 BP 社，2007
 [SEC 見える化-3] IPA/SEC 編：SEC BOOKS IT プロジェクトの「見える化」～中流工程編～，日経 BP 社，2008
 [SEC 見える化-4] IPA/SEC 編：SEC BOOKS IT プロジェクトの「見える化」～総集編～，日経 BP 社，2008

※ 7 EPM ツール：http://sec.ipa.go.jp/tool/epm.html

オリンパスソフトウェアテクノロジー株式会社

ETSS システムを独自に開発。 部内の人材を「見える化」

SEC journal 編集部

組込みソフトウェア開発分野における人材育成や、人材活用のための指標として使えることを目的としている「組込みスキル標準 ETSS ※1」。オリンパスソフトウェアテクノロジー株式会社は 2008 年よりこの ETSS を導入し、社内人材のスキル可視化に取り組んだ。同社は「スキル診断」「診断結果に基づいた研修の受講によるスキルアップの実行」「再度のスキル診断による、更なるスキルアップへ」という PDCA ※2 サイクルを確立し、ETSS の活用を軌道に乗せることに成功。また PDCA の全フェーズを支援する「SkillNavi」という社内システムを開発した(図 1)。今回は同社が ETSS を導入した背景、成功のポイントを中心に、開発担当者の玉澤康至氏、小出英樹氏に話を伺った。



トップ画面



スキル診断結果の表示



業務経歴の表示

Copyright 2009 OLYMPUS SOFTWARE TECHNOLOGY CORP. All Rights Reserved.

図 1 独自の ETSS システム「SkillNavi」

「人材が足りない」の本当の理由を探る

オリンパスソフトウェアテクノロジーは、オリンパス株式会社の子会社として 2006 年 7 月に設立した、まだ新しい会社である。

同社が ETSS を導入した理由は、「人材を可視化したい」という動機からだった。常に人が足りないと訴える現場の求めに応じて、いくら人を投入しても不足感が消えたという声は聞こえなかった。「要するに、何が足りていないのかが分かっていなかったのです。もちろん各人のスキルはそれなりに診断していましたが、結局現場に即した適材適所への配置まではいかなかった」(玉澤氏)。人はいても欲しいスキルが足りない、本当に欲しいスキルが何かが見えない。人材育成のために教育を実施したいが、まずどんな教育が必要かが分からない、という状態だった。この課題を解決するために、同社は ETSS を導入した。

ところが、ETSS を導入したものの、満足した結果を得るまでには紆余曲折があった。導入当初、スキル診断は出来たが、PDCA サイクルがうまく回らなかったのである。その原因を玉澤氏はこう分析した。「スキル可視化後のプランが無かったことが原因でした。」

『何のための診断か』という説明が十分に出来ていなかったため、現場からは診断のための入力作業が大きな負荷に感じられ、また「人事評価に使われるのではないか」という心理的不安も抱かせてしまった。現場へのフィードバックをうまく出来なかったことも、大きな反省点だったという。

なぜ導入が必要かを全社員に正しく伝える

後を受けた玉澤氏らが行ったのは、原因を分析し、PDCA サイクルが上手く回るようにする対策方針を作ることだった。

※1 ETSS : Embedded Technology Skill Standards, 組込みスキル標準

※2 PDCA : Plan Do Check Action

つまずいた要因を大きく3つに分類した。それは、①メリットについて全社員の理解不足（導入時の説明不足）、②診断・分析作業の重い負荷、③現場の否定的感情（心理的不安への配慮不足）。メリットが見えなければ、現場にやる気は起きない。

これらの対策として、次のような取り組みを行った。

まず、マネジメント層には診断結果の利用の仕方を提案した。

ETSSのスキル診断では、診断結果は本人にも見えるが、マネジメント職にとっても現場全体のスキルが俯瞰出来る。診断が浸透することにより、ゆくゆくは、新しく始めるプロジェクトに必要なスキルを持つ人材を、社内全体から探し出すことも可能となる。また、共通の言葉・考え方のコミュニケーションの実現につながり、面談等においても、相手がどんなスキルを持ち、どの技術を何年使ってきたのかが見えることで、ぐっと充実したものになるという。

次に行ったのは、診断・分析のための負荷を軽減するシステムの構築である。

ETSSのスキル診断ではスキル項目を3階層程度まで細分化して診断する。当然、自己のスキルレベルを入力するには時間と労力が必要であり、また、集まったデータを分析する側にも相応の負担がかかる。この問題を解決するために、診断から分析まで行える「SkillNavi」というシステムを自社開発した。このシステムは単に入力の手間を軽減しただけでなく、判断に迷ったときは「ガイド」ボタンを押すと判断基準の詳細が読める等、レベル判断に迷わない工夫がなされている。このシステムを導入・展開するにあたって、担当者が重視したのが「事前の十分な説明」だ。

「人事評価には使わないことを、きちんと理解してもらえようにしました」（玉澤氏）。また、現場の“やらされ感”を無くすために、この診断が各自のキャリアパスの形成に役立つことを、丁寧にオリエンテーションしたという。自分のスキルに合った仕事を見つけるためにも、客観的な評価をすることが大切であることを説明した。「評価は自己採点ですから、過大評価すると身の丈より大きい仕事を任されてしまうし、過小評価すると簡単な仕事に来てしま

う。だから自分の能力を正しく評価して欲しい…と訴えました」スキルレベルの設定も、自社の実態に合わせて基本のETSSで設定されている3段階から4段階へ調整する等、各所に工夫を凝らした。

見えてきた人材育成教育プラン

システム導入開始から約1年。ようやくPDCAの1サイクルが回って、見えてきたものがあるという。

「スキルアップ計画実現にめどが立ってきました。スキル診断をしてみると、1～2あたりのレベルまでは外部講習や、書籍等による自己学習での対応が可能だと分かりました。ただその上となると、社内のプロが自社のプロセスに合わせた研修を行わなければならないのです」（小出氏）。そして今期の目標の1つは、レベル1～2の人がどの書籍を読めばよいかをシステム上で表示出来るようにすることだという。レベル3～4の人材を育成するための仕組みも、検討していかなくてはならない。また、人材育成も、漫然と実施するのではなく、どこに比重をかけるかが見えることで、効率的に行うことが出来る。

スキルを診断しただけでは、現場の問題は解決しない。診断結果をどう生かすか、最終ゴールを明確にしないと、ETSSのPDCAサイクルは回り始めない。オリンパスソフトウェアテクノロジーはその課題を見事にクリアした事例と言える。

会社情報

商号	オリンパスソフトウェアテクノロジー株式会社
設立	2006年7月（創業1983年1月）
本社	〒163-1414 東京都新宿区西新宿3-20-2 東京オペラシティタワー14F
資本金	1億円
代表者	代表取締役社長 天野 常彦
従業員数	521名（2009年4月1日現在）
事業内容	オリンパス製品のプロダクトウェア開発 医療・健康、映像・情報、工業関連分野の製品の組込みソフト、看護支援システム、歯科用動画ファイリングシステム等のアプリケーション・システムの開発

株式会社日立情報制御ソリューションズ

事業と技術の特徴に合わせて 3つのスキル標準を活用

SEC journal 編集部

株式会社日立情報制御ソリューションズの事業は、ソフトウェアの設計・開発から基板やLSI等ハードウェア設計、制御系システムの開発を行う等、領域が広いことが特徴である。その特徴に合わせて同社は「ITSS※1」「拡張ETSS※2」「CTSS※3」という3つのスキル標準を組み合わせて、人材育成への活用を進めている。この取り組みを担当者の高松良一氏、小野孝秋氏に伺った。

「あるべき人材像」をはっきりさせたい

すべての技術者（約2,200人）を対象として、組織における技術者の役割とスキルを明らかにし、キャリア計画とスキルアップへの道筋をガイドする「海図」に相当するキャリア開発制度を作成しよう——。日立情報制御ソリューションズ技師長の高松良一氏はかねてからこう考えていた。しかし、それを実現することは容易ではなかったようだ。その背景にあるのは同社の事業領域の広さと技術の多彩さである。

同社の事業領域は、①電力・鉄道会社向けの情報システムや火力・原子力発電所の監視制御システム等を開発する「社会インフラシステム分野」、②製造業・流通業・小売業向けの情報システムを開発する「産業・流通システム分野」、③カーオーディオやセキュリティシステム等の「組込みシステム分野」と幅広い。技術領域も同様だ。基板やLSI等ハードウェアの設計も行っているし、組込みシステム分野ではアプリケーションソフトウェアだけでなくOSも提供している。

同社はこのように幅広く多彩な事業領域を持つため、組織ごとにあるべき人材像と技術者のキャリア計画を同一のものとするのが難しいという事情がある。そこで高松氏は、ITSSとETSSに着目した。「世の中の物差しを参考にしよう」（高松氏）とITSSとETSSを基にして同社のキャリア開発制度は動き出した。

すべての技術者をカバーするスキル標準を考案

同社はまず、ITSSとETSSをカスタマイズして、自社独自のキャリア開発制度を設計した。それが図1である。

ETSSをベースにして拡張項目としてハードとLSIを設計する技術者のスキルもカバーする拡張ETSSを、また

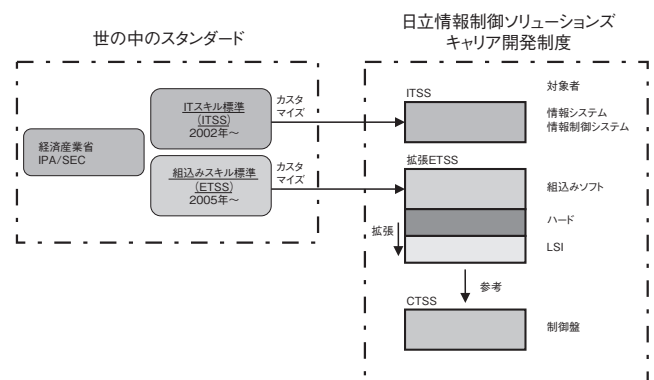


図1 自社独自のキャリア開発制度

制御盤システムの設計・開発部門のみを対象とした「CTSS」を作成した。これは、制御盤の設計文化・スキル領域が、その他の開発部門と異なるためである。

各標準のスキルレベルは、ETSSの4レベルに合わせ、スキル内容の定義書類を統一して記述し、フレームワークの構造を共通化していることが重要なポイントになっている。同社には複数の仕組みにまたがる技術者が存在する。そのために、人材の流動性に対応出来るよう工夫したのだ。こうして、ハード・ソフト・制御すべてのエンジニアのスキルが診断出来るようになった。

現在のキャリアから、将来の目標まで可視化

高松氏は、まずITSS、ETSS、CTSSごとに委員会組織を立ち上げて項目を決めたが、全エンジニアを対象にするため、ETSSのスキル項目だけでも、技術要素201項目、開発技術226項目、管理技術等共通技術48項目…と膨大なものになった。そのため、職種によって必要なスキルをグループ分けした「スキルセット（A3サイズ3枚分）」を用意し、自分の職種のある部分だけを入力すればよいように工夫した。

※1 ITSS: IT Skill Standard, ITスキル標準

※2 ETSS: Embedded Technology Skill Standards, 組込みスキル標準

※3 CTSS: Control Technology Skill Standard

また、一口にキャリアやスキルと言っても人によってイメージする中身は違う。そこで管理職に対してこの診断表の説明を、20人単位で、1カ月間合計26回かけて行った。「やれば出来る」ではなく「実績・経験」で入力してもらうこと、「いつまでをキャリアとみなすか」については「現在でもその実務能力があるか（現場で作業出来るか）」を基準にする等を、個人によって判断に乖離が出ないように配慮したのである。

管理職教育を終えて、2009年秋には技術者各自がスキル項目へのレベル入力等を行い、管理職と面談してキャリア計画を作成するというフェーズに入った。ここで行ったのは、①キャリア診断表の作成（技術者の現在の職種・専門分野の3年後のキャリア計画を記述する）、②スキル診断表の作成（技術者の現在のスキルレベルを入力すると共に、1年後のスキル向上計画と教育計画を定める）の2つである。

フレームワークへの記入ルールは、現在及び将来の技術者のキャリアとスキルのレベルを明示的に表現するものだ。例えばキャリアフレームワークに入力するのは、「◎」「○」「☆」という3つの印。「◎」は現在の主たるキャリアを表し、「○」は現在、他に担えるキャリアを示している。「☆」は、1～3年後の目標キャリアだ。それぞれを記入することによって、職種や専門分野ごとに現在と将来のキャリアレベルが可視化される訳である。

この診断表に基づいて図2の例のようにキャリア面談が行われ、上司と本人が現在の状況を確認するだけでなく、将来のキャリアパスについても話し合いを持って、どういった教育が必要かが、具体的に見えるようになった。

「実利」の提供をキーワードに定着を目指す

同社は今後、毎年10月に管理職と部下との間でキャリア面談を行っていく。キャリア開発制度は、まだサイクルが回り始めたところであるが、速やかな定着のために、まずは管理職にそのメリットを実感していただき、自発的に活用されることを狙っている。「見える化した後の、具体

1回/年のキャリア面談シーン

キャリア診断表

職種	EA		EB		EE		EF				EL	
	ソフトウェア		ハードウェア		システムアーキテクト		ソフトウェアエンジニア				アプリケーションエンジニア	
	S	S/H	S	S	S	S	S	S	S	S	S	
専門分野	組込ソフトハードLSI	組込ソフトハードLSI	組込ハードウェア開発	組込ソフトウェア開発	情報系アプリケーション開発	制御系アプリケーション開発	OSドライバ開発	通信ソフト開発	OSドライバ開発	通信ソフト開発	対ソフトウェア注I/P等	対LSI製造現場立ち上げサポート
レベル	レベル7	レベル6	レベル5	レベル4	レベル3	レベル2	レベル1	レベル7	レベル6	レベル5	レベル4	レベル3
ハイレベル												
ミドルレベル												
レベル3												
レベル2												
レベル1												

Aさんの評価

図2 診断表に基づく面談の例

的なアクションが大事だと思っています」（高松氏）

例えばこの診断表があれば、管理職は部下の診断表を基に、「ソフトウェアエンジニアのレベルを4に上げると共に、プロジェクトマネージャとして働いてもらいたい」と部下に働きかけたり、異動後でも、すぐに部下たちのキャリアやスキルを把握することが可能になる。

技術者のスキルをデータベース化しておけば、新たな事業に参入しようとした際にその事業に必要なスキルを持つ人材を探し出すことも容易だ。また、今まで漠然と社内教育コースを案内していたが、個人のスキル目標に合わせて受講を勧める等、きめ細やかな育成が可能になる。

増減するスキル項目のメンテナンス、入力や分析のシステム化等、制度を続けていく上での課題はあるが、目標とする「海図」作りは着々と進んでいるように見受けられた。

会社情報

商号	株式会社日立情報制御ソリューションズ
設立	2006年4月
本社	(茨城本社) 茨城県日立市大みか町 5-1-26 (東京本社) 東京都港区港南 2-5-3 (オリックス品川ビル)
資本金	22億7千万円
代表者	取締役社長 茅根 修
社員数	2,607名 (2009年3月末)
売上高	532億円 (2008年度)
事業内容	システムエンジニアリング及びソフトウェアの製作・販売 情報処理機器及び情報・制御システムの開発・設計・製造・販売・保守 電気工事・電気通信工事・機械器具設置工事 電気・電子機械器具の製造・販売
関連会社	茨城日立情報サービス株式会社



SEC journal 論文のまとめ方

SEC 組込み系プロジェクト エンジニアリング領域 幹事

平山 雅之

ソフトウェア開発の現場では様々な工夫や努力が続けられている。こうした貴重な経験を少数の組織内で埋もれさせてしまうのは非常にもったいない。ぜひとも、論文等の形にして多くの人たちが利用出来るように整理しておくとのよいのではないだろうか。本稿では日頃のソフトウェア開発現場での経験を基に、ソフトウェア工学の実践論文を作成していくための考え方等を紹介してみたい。



1 ソフトウェア工学実証論文とは

ソフトウェアの品質を向上させ、安全・安心なソフトウェアを効率的に開発するためには、ソフトウェア工学を活用することが有効である。ソフトウェア工学は実際のソフトウェア開発に適用されることによって広く役に立つという実学の性質を持っている。その一方で、実際のソフトウェア開発は様々な企業の思惑や製品市場の競争の中で行われるために、ソフトウェアを開発する過程でどのような工夫がなされ、またソフトウェア工学がどのように生かされているかといった情報はあまり公にはされていない。結果として、様々な企業、様々な人たちが同じようなところで悩み、そして同じような問題で苦労したという話をよく耳にする。ソフトウェア開発に関するこのような現状は本当に健全なものと言えるのだろうか？ 我々が日々の業務の中で得た様々な経験や知見をできればこの分野の多くの仲間と分かち合い、参考になる情報があれば整理し公開していく、という姿勢も重要なのではないだろうか。「技術論文」というと極めて格調高く、難しいものを連想しがちであるが、「論文を書く」ということの本質は実はこうした地道な活動の延長線上に位置しているものなのである。本稿ではこのような「日頃の活動の中から得られた経験や知見を整理する」という視点に立ち、その延長線上としてのソフトウェア工学の実践論文をどのように整理していくかを考えてみたい。



2 論文に整理することのメリット

(1) なぜ経験を整理するのか

日頃のソフトウェア開発等をしている中で、自分が体験したこと、考えたことについて、他の人の意見や考え方を聞いてみたい、あるいは自分の考えた工夫等を人にも紹介したいと思った経験はないだろうか。この場合、まず自分の考えや意見、工夫等を他人に伝わる形にまとめることが近道である。まとめる方法としては、スライドにする方法

もあるし、もちろん文章にする方法もある。それぞれ一長一短があるが、技術や工夫を相手に正しく伝えることを考えると、文章にまとめたほうが有効である。文章にまとめる場合には、ポイントに合わせて様々な説明等を加えることも出来、またその文章を読めばある程度、筆者の考えを追うことが出来る。

(2) 文章に整理することのメリット

しかしながら、自分の経験を文章に整理する場合、時系列的に実施した作業を書き連ねるだけでは、自身の経験や考えは必ずしも十分には伝わらない。そもそも、「相手にどのようなことを伝えたいのか」「自身の経験からどのようなことが分かったのか」あるいは「従来に比べてどのような点で工夫したのか」そして「それによってどのようなうれしいことが起きたのか」等相手に伝えたい事項の核となる点を整理しておかないと散漫な文章になってしまう。これは言い換えると、文章を書くということは「自身が体験したことを再構成しポイントをまとめ直す」ことに他ならないことを意味している。実は実践論文として文章を起こす最大のメリットは、この「自身の体験の再整理」をすることにある。

一方でこのようにして整理した論文は査読者によってチェックを受けることになる。論文のチェックを受けるといって、気後れしてしまう方も少なくないかもしれない。しかし、論文の査読者は、技術的な側面、経験の整理の仕方、文章表現等様々な視点からのアドバイスを著者にフィードバックしてくれる頼もしい存在でもある。SEC journalの場合、投稿された論文の分野を考慮し、その分野に関して産業界で経験を積んだ技術者と学術機関等での分野を専門にされている研究者の合わせて2名の方が査読を担当している。査読者からは、より良い論文とするためのアドバイスやより分かりやすい整理の仕方等がコメントとして付けられるが、こうしたコメントは自身の経験を整理し見つけ直すという点からも貴重なものが多いと思う。



3 分かりやすい論文にまとめる方法

ここでは「分かりやすい論文」をまとめるために、図1に示すように5つのステップを考え、その詳細を紹介していく。

3.1 Step.1 主たるメッセージを整理する

論文は人に自分の経験や考えを伝える手段である。その点からは、まず、その論文によってどのような人たちに何を伝えたいのかを考えるとところから始めなければならない。

SEC journalの場合、主な読者は産業界を中心としてソフトウェア開発に何らかの形でかかわっている方々が中心となっている。これらの読者の多くは、「より良いソフトウェアを効率的に開発するためにどのような工夫をしたらよいか」という期待を持ってSEC journalの記事をご覧になっているということを理解しておく必要がある。そして、こうした読者の方々が、皆さんが執筆した論文を読み終わったとき、「要はその論文で取り上げた問題は何で、どのような方策によりそれがどうなったのか」という部分がクリアに伝わるように整理しておく必要がある。このためには、まず、「その論文は一言で言うと、何を伝えようとしているか」を出来れば2～3行で言い切れるように考えてみるとよい。そして、この数行の中から、更に論文の中核をなすキーワードを紡ぐことによって、論文のタイトルとすることが出来る。

実際の開発プロジェクト等の中で、この「一言で言うと何を伝えようとしているか」を鋭く抜き出していくことは実はかなり難しい作業である。こうしたことにあまり慣れていない場合、出来れば、開発等の中で感じたこと、問題と思ったことを書き出したり、実際の開発の中で作成した図や表、収集したデータを並べて眺めてみるのところから始めるとよい。もちろん、これらの情報は玉石混淆のもので、論文を書く側から言うと、あれもこれも言っておきたいということになるかもしれない。しかしそれら様々な情報の

中、枝葉のような情報を除いていくと1つ2つ、ポイントとなる点が見えてくると思う。そうした部分を更によく整理して考えていくと、「要はどういうことか」が見えてくる場合もあり、それをヒントに「伝えようとするメッセージ」を考えていくとよい。

もちろん、論文とするためには、既に公開されているメッセージや経験と全く同じものを重ねて取り上げることは好ましくない。しかし、多くの場合、たとえ同じような経験であっても、その細部の条件等が異なっている場合も少なくなく、その違いに応じた手法のアレンジや考え方の工夫が入る場合がほとんどである。ぜひ、そうした部分に着目して、論文の主たるメッセージ、主要テーマを切り出す努力をしてみたい。

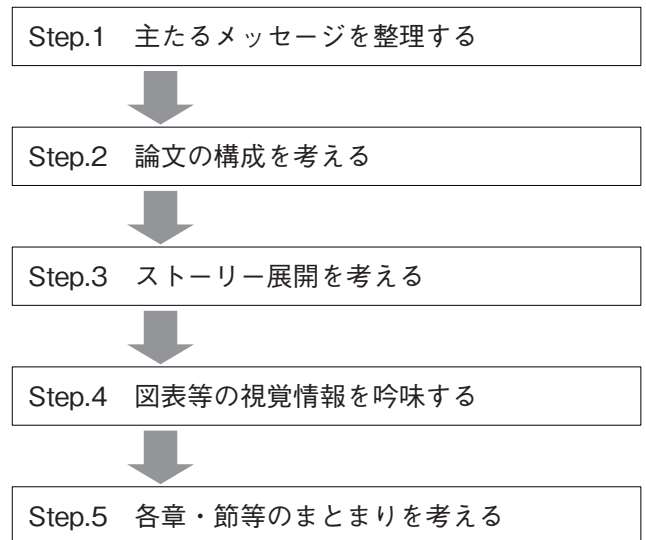


図1 論文をまとめる手順

3.2 Step.2 論文の構造を考える

多くの読者の方々は、起承転結ということを知っているかと思う。実は文章をまとめる場合にも起承転結を意識してまとめていくとよいと言われている。「論文を書こう」として最初の1行目から悩んでしまう方も少なくないようだが、まず、最初の1行を

書き出す前に、文章全体の設計をすることを意識して見て欲しい。論文の起承転結は以下のように考えておくとよい。

- ・起：その文章ではそもそもどのような問題意識で何を取り上げようとしているのか
- ・承：その問題意識に対して、どのような方向からどのように切り込んでいったか
- ・転：具体的な施策や実施内容によって、どのようなことが分かり、当初の問題はどのようになったか
- ・結：結果的に今回の経験は当初設定した問題意識に対して、どのような解をもたらしたか

これまで査読委員長としてSEC journalに投稿された論文のすべてに目を通してきたが、実は意外とこうした文章の設計が出来ていないものが少なくない。通常読みやすさを考えると、起承転結の文章量のバランスは1:3:3:1ないし、1:4:4:1程度を意識してみるとよいかもしれない。6ページの論文を書く場合であれば、起及び結の部分はそれぞれ約1ページ程度に相当する(図2)。このようにあらかじめ論文の要素を意識しその分量を決めておくということは、分かりやすい文章を書く上で極めて重要である。そして各要素(起承転結)の記述目安量を念頭に、各要素にどのようなポイントを織り込むかを簡条書きにしてみよう。SEC journal論文で過去に不採録となったものの中で、比較的好く見受けられるものとしては、文章の行間で大きく論旨に飛躍があるパターンや、文章や論旨が冗長で回りくどいもの、あるいは本論と関係の無い余分な情報でメタボ状態になっているパターン等がある。これらはいずれも文章の起承転結のバランスが崩れてしまうことが多い。

3.3 Step.3 ストーリー展開を考える

「起承転結を意識しながらそれぞれのセクションで記述するポイントを簡条書きにする」と説明した。しかし多くの方は、それではそれらのポイントはどのようにして抜き出せばよいのかという次の問題に直面してしまうかもしれない。論文を構成するポイントの抜き出しを行う場合、例えば、起承転結でおおよそ決めた記述量に合わせて、論文半ページあたり1枚程度のスライド資料を作成してみるとよい。各要素で盛り込むべきポイントは以下のような点である。

(1) “起”の部分

論文の“起”の部分では、論文で取り上げる分野の特徴、その特徴からくる課題等を中心に簡条書きで背景や課題を2~3挙げて考えてみるとよい。

(2) “承”の部分

ここでは課題を解決するという面から、まず、①これまでの課題の原因がどこにあったかを分析し、②その課題解決に向けてどのような方向から、③どのようなアプローチをしたかをそれぞれ1スライドくらいずつ整理してみるとよい。これらは出来るだけ分かりやすく図等を用いて視覚的に整理してみるとよいかもしれない。

(3) “転”の部分

ここでは、課題解決に向けた努力の結果、どのようになったかを中心に整理する。場合によっては以前の状況との対比等を行い、課題解決の努力の価値や効果を分析してみるとよい。この場合にも図で整理したり、具体的なデータ等があれば、それを丁寧に分析しグラフ化する等してみるとよい。更に得られた結果等についての考察した事項等も整理しておく。論文としての面白さを考えるとこの“転”の部分で、いかに発想を深化させていくかということになる。もちろん論文である以上、論理的な飛躍はあまり好ましくないが、様々な視点からの読者に驚きや感心を与える考察を加える努力をしてみるとよい。

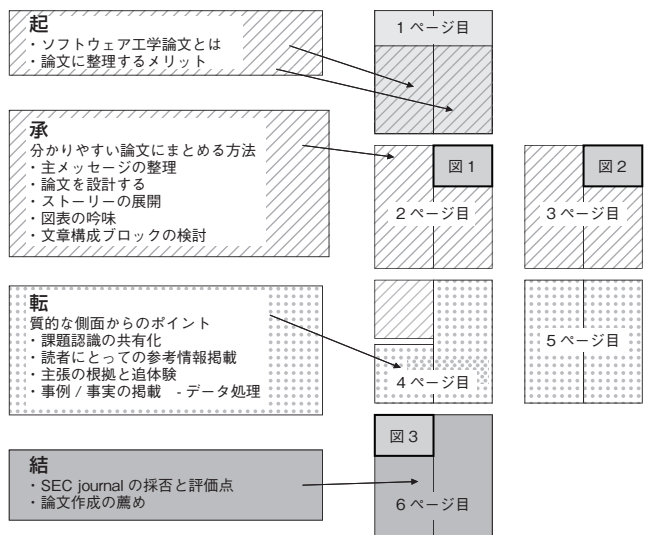


図2 論文の構成を考える

(4) “結”の部分

“結”の部分は論文の中ではとくに重要な箇所である。ここでは、「①論文の中で取り上げた課題に対して、②どのような方法によって、③その課題をどのように解決したか」を意識して、まずはこの3つを3行で簡条書きにしてみるとよい。その上で全体を振り返って、更にどのような工夫が求められるか、あるいは、その論文や研究での今後取り組んでいくべき事項等も簡潔に整理しておくるとよい。

3.4 Step.4 図表等の視覚情報を吟味する

さて、このようにスライドを活用して論文ストーリーを組み立てる場合、とくに留意したいのは、出来るだけ、論文内で紹介していく図や表、グラフ等の素材も合わせて吟味していくことが必要という点である。読者にとっての分かりやすさを考えた場合、筆者の言いたいこと、伝えたいことを視覚的に表す補助的な情報が含まれているのと、含まれていないのとでは雲泥の差が生じてしまう。極端な話、本文を読まなくても、本文の各章・節のタイトルやそこに配置された図表やグラフを追いかけていくだけでおおよその話の内容が追えるようになっていたほうがよい。図や表、グラフ等のボリュームについては、論文で扱うトピックにもよるが、全体の約2割前後を目安に考えておくとよい。6ページの論文の場合、余裕を見て約1ページくらいを図表に充てる心積もりをしておくとうい。

3.5 Step.5 各章・節等のまとまりを考える

さてストーリーとそこにちりばめる素材が見えてきた段階で、次にすべきことは、より具体的に論文を構成する章や節等の要素に分解していく作業である。ここでは、先に大まかに整理した論文全体の起承転結を意識しながら、論文の章節立てを考えていく。SEC journalの場合、通常は2段組で各段は横25字、40行で1ページが構成される。文章を構成する最小要素としての節レベルでは、読みやすさを考えると20行程度を1つのまとまりとして考えると分かりやすい。ただし、1つの節でこの分量に収まり切れない場合には、節の中で(1)(2)等の形でまとまりを作り、それぞれの記述はやはり20行程度に抑えたほうが分かりやすくなる。この20行程度のまとまりの中で言及するトピックは、1つないし2つ程度として、先にスライドでストーリー整理をした際のポイントに対応させる形で考えていくと分かりやすい。



4 論文整理のポイント

ここまでは論文をまとめる場合の手順や方法等を中心に説明してみた。ここからは、本稿の“転”にあたる部分として、論文整理を行う上での質的な側面からの着眼点等について考えてみたい。

4.1 課題認識を共有する

論文の書き出しに当たる“起”の部分では、論文の背景や論文で取り上げる課題認識を書くことと説明した。実は読者の興味や共感を呼ぶためには、著者と読者の間でこの課題認識が共有出来ているかどうか極めて重要となってくる。ソフトウェアの開発の中では様々な課題を抱えている

開発者は少なくない。このため、論文の書き出しで筆者が解決しようとした課題を丁寧に説明すれば、たとえ読者と筆者がかかわっている製品分野が異なっているとしても多くの読者にいわゆる「思いあたる課題」あるいは「以前に遭遇した同じような体験」として、読者の興味を引くことが出来るはずである。もちろん、この場合、筆者がどのような製品分野でその課題認識を持ち、その課題解決を試みたかといった実践の対象に関する説明も丁寧にすべきである。例えば、以前、SEC journalに掲載されたある実践論文では、投稿時には「弊社が扱う製品分野では、ソースコードの再利用促進がビジネス上の重要な要素であり・・・」と課題背景が説明されていた。しかし、この部分を見る限り、筆者がどのような製品分野を念頭に置いているかが読み取れず、なぜこうした部分が重要になってきたかが分からないとのコメントが査読者によって付けられ、最終的に「弊社が扱う家電機器分野のソフトウェアでは・・・」という形で修正をさせていただいた。このようなわずかな修正であっても、読者からすると、筆者が取り組んだ分野をある程度念頭に置いて、論文のそれ以降を読み進むことが出来、筆者が抱える課題や課題解決のアプローチを身近に感じ取ることが出来るようになる。

4.2 読者にとって役に立つ情報を盛り込む

実践論文では読者にとって役に立つ情報をいかに盛り込んでいくかも大きなポイントである。ここでは具体例として、SEC journalに比較的多く投稿されるトピックの1つである技術者教育の問題を例に考えてみたい。さて技術者教育というテーマについて実践論文を作成する場合、どのような情報を盛り込んだら読者にとって参考になるかを考えてみよう。この場合、以下のようなポイントを考えることが出来るのではないだろうか。

- ① 教育の受講対象者は誰で、誰がどのような目標を持って教育を行ったか
- ② どのような教育スタイルや形態で教えるか
- ③ 教育の中でどのようなトピックを取り上げて、それにどの程度、時間をかけて教えているか
- ④ 上記を行う上で、受講者への定着や理解度を向上させるために、どのような工夫をしているか
- ⑤ 教育の効果がどのように見られたか

さて、これらの点について、読者に参考になる情報を盛り込むことを考えると、例えば、教育のシラバスや時間割、あるいは、実際の教材や演習課題の一部サンプル等を参考掲載したほうがよいかもしれない。もっとも技術論文は技術解説とは異なるため、これらの事例の中でどのような工夫が施され、それらがどのような効果をもたらしたかまで、きちんと考察し事例の汎化を考えることが重要である。こ

これらの情報は時として、外部への公開をためられるものもあるかもしれないが、その場合には、差し障りの無い範囲で部分的にでも特徴を説明出来るように工夫して掲載することを考えて欲しい。また、⑤の教育の効果については、教育の内容にもよるが、教育受講前後の受講者の行動パターンや取り組み方の違い等を観察評価し定量的な違いを論じる方法や、受講者に対してアンケート等を行った結果を分析する方法等、様々な効果のアピール方法が考えられる。この例からも分かるように「読者に参考になる情報」を論文中に盛り込むポイントとしては、次の2点に留意しておくといよい。

- ・ポイント1：取り組みの中で工夫した点等を中心に、具体的な取り組みの内容が分かるような資料やデータを図表やグラフとして盛り込む
- ・ポイント2：取り組みの効果について、その取り組みをする前後の違い等を、定量的あるいは定性的な効果を客観的に分析整理し、図表等を用いて盛り込んでいく

4.3 主張の根拠を盛り込む

論文とは自己の思想や考え方を他人に伝えるためのものである。とくにSEC journalに投稿されるソフトウェア工学の実践論文では、筆者が実際のソフトウェア開発で得られた経験の中から、参考となると考えられた技術や考え方、方法等を整理して伝えていくことが中心となる。このため論文の中には筆者の主張や技術への思い等も含まなければならない。ただし、こうした主張や思いに関しては、常にその根拠となる情報やなぜ、筆者がそうした主張をするに至ったかの思考過程等も併せて記述すべきである。

しかし、これまでにSEC journalに投稿された論文を見てみると、こうした筆者の主張や思いが一方的に述べられる形で、その根拠等の情報が足りないものも少なくない。実際、数年前に投稿されたある論文では「ソフトウェアプロセス改善を進めると組織風土が変わり、品質が向上する」という考え方を主張されている論文があった。しかし、残念なことにその論文では、その主張の裏付けとして、「プロセス改善を進めた結果として、どのように組織風土が変わったか」あるいは「組織風土が変わった結果、品質はどのようになったか」という2つの問いについて、具体的な事例や数値が掲載されていなかった。このように主張の根拠となる情報が提供されない中で、一般の読者に筆者の主張を納得させるのは極めて難しいと言える。つまり実践論文の価値は、ソフトウェア工学をどのように実適用したかという事実や事例の情報を含めて、そこから導かれた筆者の考えを伝えて初めて、読者の期待に応えることが出来る。

4.4 事例や事実を掲載する

上記でも述べたように実践論文を書く上で、とくに重視されるのは事例やそこから得られた事実であり、それらの汎化した考え方である。しかし、自部門のソフトウェア開発を事例として紹介したり、そこで観察された事実やデータを公開することはためられる場合も少なくないのではないだろうか。このため、こうした事例や事実を論文に掲載する場合には、

- ① データの匿名化
- ② データの正規化

等の工夫が必要となってくる。例えば、「従来に無い新しいコードレビューの手法を考え、実際のプロジェクトで適用してみたところ、より確実に不具合を検出出来るようになった」という主張を実際のデータを用いて説明することを考えてみよう。

- ① データの匿名化

この場合、まず、重要なのは、新しい手法の有効性を主張するのが目的であるから、その手法を適用する前後の比較が必要ということになる。実際の開発の中では新しいレビュー手法をA3320という機種に適用した場合に、以前の開発機種B7767という機種との対比を行うこととしてみる。この場合、A3320、B7767という機種名はさほど重要ではない。新手法の適用前後の比較という点では機種A、機種Bとしてしまっても構わない。逆に論文として必要な情報は、仮にこの機種A、機種Bを比較する場合に、比較の観点はレビュー方式の優位性であることを考え、比較の観点に影響を及ぼすであろう条件が同一であったかどうかだけを明確にしておけばよいということになる。この場合であれば、それぞれの機種で実現するソフトウェアの難易度がほぼ同じなのか、あるいはレビューア的能力は同じだったのか等を明確にすることが重要である。

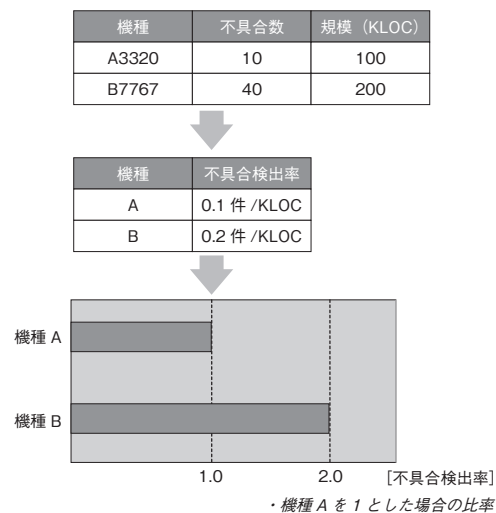


図3 データの加工の考え方

② データの正規化（ノーマライズ）

更に前出の事例で機種 A では規模 100KLOC で不具合 10 個、機種 B では規模 200KLOC で 40 個を検出したとしよう。このような実際の開発の中で観察された不具合数等をそのまま論文に掲載することは、様々な制約で難しい場合が多いかもしれない。この場合、実際に得られた数字を次のような方法でノーマライズ処理を施していくとよい。

まず、図 3 に示すように実際に検出された不具合をコード量で割って、それぞれ機種 A:0.1 個 /KLOC、機種 B:0.2 個 /LOC という数字が得られるが、この数字でもレビュー手法の優位性の違いは論じることが出来る。更に、レビュー方式の優位性を論じようとするならば、機種 A を基準に機種 B での不具合検出率の相対比を求め、「機種 A での不具合検出率を 1 とした場合に、機種 B では 2.0 となった」と表現することでも十分である。ここまでノーマライズすると個々の機種で検出された不具合数という直接的な数字は完全にオブラートに包むことが出来る。このように、手法等を実際に適用して得られたデータは、そのまま掲載が難しい場合に、比率で表す等の処理を施して、いかようにも説得力を持ったデータに変換することが出来る。ただし、元データをあまりにノーマライズしすぎると元データが持つ意味が見えづらくなる場合もあるため、常にデータの意味や比較の意味を考えながらノーマライズする必要がある。



5 おわりに

ここまで SEC journal に掲載することを念頭に、ソフトウェア工学の実践論文を書く際のポイントを整理してみた。ソフトウェア関連の論文の書き方については、SEC journal No.16 の中でも詳しく紹介されている。具体的な目次構成や論文作成のポイント等については本稿と併せて読んでいただくとよいと思う。

ここまで述べたように、SEC journal では、「読者に参考となるかどうか」という点が採否判断の大きなポイントとなっている。通常の研究論文等の場合、論文の採否判定では研究や技術の新規性、有用性、あるいは論文に書かれたことの信頼性等が重要視されるが、SEC journal の場合には、とくに実践経験の情報としての有用性に重点が置かれている。もちろん、掲載される経験が、既にどこかで発表されていたり、一般的に周知の事例であれば掲載は難しい、内容面に明らかな誤り等が含まれるものは掲載は難しい。本稿をご覧になっている方々の多くは、ソフトウェア開発に日夜携わっていらっしゃる中で、恐らく、様々な技術や手法をいろいろと工夫を加えながら利用されているのではないだろうか。表 1 は本稿のまとめとして、分かり

やすい論文を書くためのポイントをチェックリストに整理したものである。これらも参考にして、ぜひ、今一度、日頃の開発の中で試みている工夫を見つめ直し、実践論文として整理し投稿することをご検討いただければと思う。

表 1 分かりやすい論文にするためのチェックリスト

Check 1	論文全体で述べたいことが一言で整理出来ているか
Check 2	論文の全体構成として、起承転結が明確になっているか
Check 3	論文を構成する要素（起承転結）に盛り込まれた主張や考えは簡潔にまとめられているか
Check 4	論文の流れを章節のタイトルや図表から追うことが出来るか
Check 5	論文の主張を補うための必要な図表、グラフが盛り込まれているか
Check 6	論文のはじめにや背景説明、課題説明は読者の共感を得られる書き方になっているか
Check 7	読者の視点から参考になる情報が盛り込まれているか
Check 8	筆者の主張のみが展開されていないか
Check 9	筆者の主張や考え方を理解するために必要な背景情報やデータが説明されているか
Check 10	筆者が経験したことをベースとして、事例や事実が簡潔に紹介されているか
Check 11	提案方法等の効果を評価するための客観的なデータが提示されているか
Check 12	論文の章節は適切なボリュームや情報を含んでいるか
Check 13	記述内容に論理的な矛盾や飛躍は無いのか
Check 14	読み物として見た場合に、興味を持って読み通せる工夫や記述がされているか

参考文献

- [井上 2009] 井上克郎：ソフトウェアに関する論文の書き方, SEC journal No.16, pp.51-57, 2009
[清水 1959] 清水幾太郎：論文の書き方, 岩波書店, 1959

経営者はアーキテクチャと形式手法を忘れてはいけない

高信頼性システム技術作業部会委員
グローバル情報社会研究所株式会社 代表取締役社長
藤枝 純教

本記事は、次の3項目を念頭に執筆した。

- (1)自動車や宇宙、航空、原子力、医療、スマート・グリッド、ITS^{*1}等の高性能・高信頼性が要求されるシステムの開発と検証管理の技術能力を確保し、国際競争力を持つ。このことに日本のIT企業が寄与する。
- (2)クラウド・コンピューティング及び国際的なアウトソーシングにより低コストかつ短い時間で開発し、高信頼のサービス提供により高い顧客満足度を獲得するにあたり、アーキテクチャの持つ意味が極めて高く、かつ高信頼性検証能力確保への投資は経営的に成り立つ、ということを経営者に認識していただく。
- (3)上記2項目の実践のために、IT管理者がアーキテクチャ・ベースのモデル化と形式手法の採用の必要な理由を理解し、それを推進する。

1. 21世紀の日本のIT戦略を考える

1.1 これまでのIT化とこれからの挑戦

1980年代後半からの日本は、積極的な投資と標準化を進めた結果、世界一流の通信インフラを持つに至った。しかし、ユーザ企業においては、西欧流のCIO^{*2}やアーキテクトを育てずにベンダ任せとすることが多く、またグローバルでオープンなスタンダードを導入する戦略をとらなかった。その結果、BPR^{*3}に基づく新規アプリケーションの開発というより、レガシー・システムの保守・維持により情報システムが守られてきた。

一方、組込み業界においては、主力製造部門の家電、携帯、自動車、産業ロボット機器等、成功したセクターがあるが、現在では国内需要の飽和と海外の消費低迷に米国、中国、インド、欧州、東欧成長諸国からの追上げが加わっており、大変厳しい状況である。

そこで、このような問題を受け、本稿では、2030年に向けて、アーキテクチャと形式手法を駆使する新しい2つの産業戦略を提案したい。

- ①製造業のソフトウェア技術をより一層、安全で高度なものにし、高機能・高信頼性を確保出来るミッション・クリティカル・システム分野へシフトする。

- ②サービス業の生産性を、EA^{*4}をドライバに環境や顧客中心目線で徹底して最適化し、システム化する“サービスの最適化プロジェクト”を興す。

1.2 欧州と米国の高機能・高性能・高信頼性産業への取り組みの歴史から学ぶ

欧州における、形式手法への取り組みは、1960年代のTony HoareのTheorem Proving、Robin MilnerのLCF^{*5}、1973年Cliff JonesのVDM^{*6}、Jean-Raymond Abrialの1977年のZ言語^{*7}、1996年のBメソッド^{*8}とVDMの標準化に続き、2002年にはZ言語がISOで認証された。また、2005年以降、これら形式手法を、鉄道、自動車、航空管制、原子力、電子マネー等に利用するプロジェクトに対し、欧州では戦略的な支援が行われている。一方、アーキテクチャは、1960年代のSOP^{*9}、1970年代のBSP^{*10}、次いでザックマンのEA、1995年からのTOGAF^{*11}、FEA^{*12}、DODAF^{*13}等米国主導となった。しかし、2005年頃から欧州でも対米航空宇宙機器ビジネス市場に参入すべく世界標準を取り入れ、米国の自動車及び航空機製造エンジニア協会(SAE^{*14})が標準として決めたAADL^{*15}をAirbus、EADS、Thalesら各社が取り込み、その上で形式手法を使うアプローチを2006年から始めた。2009～2010年の欧州高信頼性プロジェクトでは、次の節で述べるが、AADLを取り入れている。

※1 ITS: Intelligent Transportation Systems, 高度道路交通システム
 ※2 CIO: Chief Information Officer
 ※3 BPR: Business Process Re-engineering
 ※4 EA: Enterprise Architecture
 ※5 LCF: Logic for Computable Functions
 ※6 VDM: Vienna Development Method, IBMのウィーン研究所で開発された形式手法の1つ。
 ※7 Z言語: Z notation, Jean-Raymond Abrialを中心にオックスフォード大学のプログラミング研究グループによって開発された形式手法の1つ。Z記法、Zメソッドとも言う。

※8 Bメソッド: B method, Jean-Raymond Abrialを中心に開発された形式手法の1つ。Z言語とも関連がある。
 ※9 SOP: Study Organization Plan, IBMで開発された設計手法の1つ。
 ※10 BSP: Business System Planning, IBMで開発された、業務、データ、組織を構造化する手法。
 ※11 TOGAF: The Open Group Architecture Framework
 ※12 FEA: Federal Enterprise Architecture
 ※13 DODAF: Department of Defense Architecture Framework
 ※14 SAE: Society of Automotive Engineers
 ※15 AADL: Architecture Analysis & Design Language, 米SAEが策定したアーキテクチャ記述言語。

1.3 AADLを利用した欧州高信頼性プロジェクト

このプロジェクトには、①Eclipse ベースのオープンソースである TOPCASED ツールを使った形式検証 (2,000万ユーロ、28社)、②欧州宇宙開発局主導のAADLベースプロジェクトのSPICES (1,600万ユーロ、15社)、③同じくAADLとUMLプロファイルを使うAxlog、Thales社ら29社連合プロジェクトのASSERT、④米Carnegie Mellon UniversityのSEI^{*16}、University of Illinois at Urbana-Champaign、Vanderbilt大学と欧州の42社の組込み共同プロジェクトのARTIST2、⑤2006年からの次世代自動車設計に関する研究会Beyond AutoSAR、⑥2007年からのIntegrated Modular Avionicsに関する欧米共同の基礎研究等が含まれる。

1.4 AADLを利用した米国信頼性プロジェクト

米国での高信頼性プロジェクトの中心は、個別の形式手法から1995年になると米国国防総省におけるアーキテクチャ事件(2.2節)を契機に、アーキテクチャに焦点が移ってきた。一般企業はEAを、そして航空や自動車等の高機能高信頼性製品を作る企業はAADLをそれぞれ採用し、設計を分析検定するアプローチをとるようになった。

2007年からは、AADLベースでRFP^{*17}を出し、AADLで書かれたソフトを仮想検査・検定し、下流までの発注判断を可能にするためのプロジェクトであるAVSIが実施された。このプロジェクトでは、米国航空・防衛システムの中心ベンダ8社、Boeing、Lockheed Martin、Honeywell、Raytheon Company、Rockwell Collins等に、SEIがまとめ役として参加し、2009年に第1次結果(全部で第4次まで)をまとめている。

1.5 日本が欧州、米国の高信頼性プロジェクトから学ぶこと

日本においても、宇宙航空・重工業・精密機械・自動車等の各社のトップ・アーキテクトにアーキテクチャと検証能力を集中研修するプロジェクトを興す必要がある。

欧米ではアーキテクチャと形式手法を駆使出来るコンピュータ・サイエンスの博士数百名(推定)が1960年代以降の40年間で実社会に出てこれらの仕事をしているという。日本はどうだろうか? 残念ながら、最近20年のバブル崩壊後、不景気対応に追われ、多くの民間企業にはそこまでの余力が無い。そのため、国がリードすべきであろう。今、本腰で強化しなければ日本が尊敬される未来は実現出来ないのではないかと危惧している。

2 高信頼性検証能力への投資は経営的に成り立つのか?

情報システムのコストは、金融業務システムや航空会社等の業務システムでは膨大と広く知られているが、それ以上

に半導体工場の自動化設備等の設備投資では数千億円単位である。また製品開発には開発費だけではなく、失敗や補償のためのコストも含まねばならず、衛星1個の打ち上げ失敗の損失は保険を入れると200億円、自動車はもちろん、1個1個は安い携帯電話、ペースメーカーやAED^{*18}等でもリコールコストや人命、飢餓等の補償金等を入れれば1件で何百億円になりかねない。航空機であれば約数百億円以上、宇宙船になれば数千億円、原子炉やITSやスマート・グリッドの構築等大規模なインフラシステムにおける致命的損失となれば兆円レベルにも及ぶ可能性がある。よって事故や不具合がソフトウェアやシステムに絡んで起きたりするリスクに対しては、企業存続どころか国家の威信をかけて取り組まなければならないものもある。

事故や不具合における非がベンダにあったとしても、発注側であるユーザにも、検収した責任があるであろう。社会がソフトウェアやシステムに対してますます高い信頼性を求める環境においては、開発規模が増大していく中で、経営者はコストと信頼性を鑑みながら投資しなければならない。そのためには、高信頼性保証の自己管理能力を確保することであり、そのための人材教育が必要であるが、育成には大学卒で数年から10年はかかる。

2.1 不適当なソフトウェアテストによって生じた経済コストの証言-1

ソフトウェアの信頼性に対する議論を高める火付け役となったのは、2002年5月に米商務省技術標準局(NIST^{*19})が発表した“The Economic Impact of Inadequate Infrastructure for Software Testing”という報告書である。この中で、不適当なソフトウェアの品質と、そのテスト方法が不適切なためのコストが米国経済全体で年間595億ドル、そのうち約1/3(223億ドル)は、最適なテストが実行されていれば不必要なコストであったと報告されている。更に、ベンダでなく、“ユーザ”がソフトウェアの欠陥を修復する費用が約60%(383億ドル)に達していると報告されている。

2.2 アーキテクチャ管理が高信頼性管理の基礎であることの証言-2

—エンタプライズ・アーキテクチャの必然—

1980年代後半米国国防総省において、システムトラブルを原因とする幾つかの事故が起きたという。この時の原因調査委員会が、その真の原因は、プログラマのエラーではなく、陸海空軍がベンダごとに固有の、部分的に相互接続性に問題のあったアーキテクチャで設計されたことにあると指摘した。このことから、オープンな共通アーキテクチャを確立するために、TAFIM^{*20}という共通のアーキテクチャ設計ガイドラインが1986年に最初に書き始められ、1995年に完成した。TAFIMは、既に大統領府予算管理局と米国国防総省のDISA^{*21}に渡されており、1994年にはDISAから正式にThe Open Groupに渡され、1995年

*16 SEI : Software Engineering Institute

*17 RFP : Request For Proposal

*18 AED : Automated External Defibrillator, 自動体外式除細動器

*19 NIST : National Institute of Standards and Technology

*20 TAFIM : Technical Architecture For Information Management

*21 DISA : Defense Information Systems Agency, 米国国防情報システム局

に官民両用に使えオープンアーキテクチャとして、TOGAF1という名称で発行された。その後改訂が進められ、2002年にTOGAF7で技術アーキテクチャ、DODAF設計にもマッピングされ、2002年にTOGAF8でビジネス、2009年にはTOGAF9でソリューションアーキテクチャへの橋渡しがされた。現在では、TOGAFホームページから12万件ダウンロードされ、80カ国、1万4,000人がTOGAF9を含めThe Open Groupのアーキテクト認証者となっている。米国国防総省・英国国防省がDODAF/MODAFを推進したことは、“ディペンダビリティはアーキテクチャに始まる”という最初の教訓となった。一方、日本でのEAは遅れており、民間ではEAの実践がほとんど進まず、外資系以外は例外を除いて、人事的にアーキテクトという職種がユーザにも、ベンダにも定着していない。

なお、The Open GroupのDependability Through Assuredness WGではTOGAFと形式手法をつなぐことを目標に、AADLグループとも交流し、コンピュータで実行出来るアーキテクチャの研究を続けている。

2.3 アーキテクチャ段階でのエラー摘出が高信頼性確保の基礎であることの証言-3 —アーキテクチャの段階で正しい要件を正しく表記し、誤謬を排除出来るかが鍵—

高信頼性システム開発には、EAとして正規にアーキテクチャを構築するプロセスを実施し、その要件定義に関係する部品間、アクター間の関係条件をモデル化し、形式手法を駆使して、その“正しさ”を検証する必要がある。

2.1節で紹介したNIST発表の2002年の論文に基づきSEIは、経営者に認識して欲しいとしたポイントを以下のように報告している。

これは、要求から設計完了までのアーキテクチャを決める段階の管理こそ高信頼性確保と検証の要であるという証明である。

- ①要求からコンポーネント設計の終了までに、ソフトウェアシステム全体の70%のエラーが発生し、組み込まれる。そのうちこの段階でエラーとして発見されたのがたった3.5%であり、この段階で直すコストを1とする。
- ②コーディング・ユニットテスト段階で組み込まれたエラーは20%だが、16%はこの段階で見つかり、ここで直せばコストは5。
- ③インテグレーションテスト・システムテスト段階で生まれるエラーは10%だが、ここで見つかるエラーは50.5%で、エラーを修正するコストは10。
- ④アクセプタンス(検収)テストで組み込まれるエラーはゼロだが、見つかるエラーは9%、修正するコストは15。
- ⑤その後出てくるエラーはなんと20.5%もあり、修正コストは30。

本記事を読まれる経営者の皆様は自社において上記①～⑤をチェックされ、問題があれば解決策を提案するようにCIOに指示をして欲しい。

先の2.1節の証言-1では、適正な検証法があればテストのコストの37.5%がセーブ出来、本節の証言-3では、要求

からコンポーネント設計の終了までに70%のエラーが発見され、訂正されると見て44%、楽観的には、70%超のテストコストの節約が出来ると指摘されているが、実際にそれを行う方法については答えが出ていない。私はこの4年間にThe Open GroupのDependability Through Assuredness WGを通して、アーキテクチャ・フォーラムのメンバ等に個人的に会い、意見を聞いた。Tony Hoare、Daniel Jackson、Manfred Broy、Patrick Cousot氏ら世界の形式手法の巨匠たち、NASAやBoeing、Lockheed Martin、Rockwell Collins、Raytheonらのアーキテクトたちとの議論で得た私の上記コスト節約に関する結論は、①オープンで標準化された正規のEAプロセスを通して要件とビルディングブロック間の関係を精査し、②問題の性質に合わせて、多角的な複数の形式手法の最適適応による信頼性確保を基本とすることである。

3 標準化された正規のEAプロセスと多角的な形式手法の最適適応と信頼性確保が基本

—日本ではアーキテクチャも形式手法も世界から10年以上は遅れていると思われる—

3.1 アーキテクチャの定義

アーキテクチャの定義に関し、IEEE Std 1471 : ISO/IEC 42010を少し拡張したThe Open Groupの定義は次の通りである。

「アーキテクチャとは、複数のコンポーネントが一体となった共通の目的を持った組織体が、現在の、各コンポーネントと全体、またコンポーネントの相互関係や環境との関係を論理的に表現し、未来のターゲット・ゴールに向けて、いかにこれを設計し、発展させるかのプリンシプルである。」

3.2 形式手法の定義

形式手法の定義は、「設計の正しさの証明を集合論・論理学・数学を土台に証明するフォーマルな検証理論」である。方式的には述語論理証明によるLCF、VDM、Bメソッド、Z言語等に対し、最近はややライトウエートな新しい形式手法として、次の2つ等が登場してきた。

- ① MITのDaniel Jackson教授のALLOY^{*22}
- ② コンピュータ処理能力が向上したことから静的アナライザとして形式化するÉcole normale supérieure ParisのPatrick Cousot教授の抽象解釈によるモデルチェック方式

3.3 EAも形式手法も抽象化・論理化が基礎

形式手法を経験した誰もが感じる困難は、高い抽象化能力を持った“形式手法アーキテクト”を育てることだと言う。この抽象化はシステムが高度化しソフトのステップが数億ステップになるシステム複数を相互に接続するには、高信頼性システムが必要となる。このような構造の精密性・強靭性を確保するためには、高度な抽象化能力が必要になり、また複雑性を抽象化・単純構造化するトップダウン設

計を行った上で、形式手法を駆使して無矛盾性を確保する必要が出てくる。

1990年頃から生まれてきたEAは、まさに、企業やその製品群、プロジェクトの目標から構造、成果物を設計するための抽象化プロセスの標準化であり、テンプレートであり、マネージメント・プラクティスである。

アーキテクチャは、howのエンジニアリングではなくwhat、who、where、when、why、why notを追求するサイエンスと考えるのが適切である。アーキテクチャが、正確に、パラレルプロセスを可能に出来る形で、トップダウンで、抽象解釈された分割設計出来れば、指数関数的に広がる状態推移のモデルチェックにも形式的検証の一般での利用のフィージビリティが現実味を帯びてくる。

4 形式手法を駆使した高信頼性確保の標準化の今後の課題

4.1 ディペンダビリティ（高信頼性）は経営の目標、アーキテクチャはその構造設計方式で、形式手法はその検証手法

経営者の方には、次の方法を提案したい。「自社の長期計画と短期計画のゴールと経営プリンシプルに合わせて、最適統合性確保のためにEAを駆使し、提供する主力製品やサービスの関係をモデル化し表現する。」このための最初のステップでは、経営面で①経済性、②生存持続性、③顧客満足度、④グローバル・オープン対応性の4つのビューポイントをEA的に確認する。その後、ディペンダビリティに関し、下記の非機能6つのビューポイントでアーキテクチャのモデル検証を、出来れば形式手法を駆使して総合的検証を行う。

①安全性（Safety）の証明、②信頼性（Reliability）の証明、③可用性（Availability）の証明、④機密保持性（Security）の証明、⑤統一性（Integrity）の証明、⑥保守サービス性（Serviceability）の証明。

米国では、高信頼性システム検収において、ある条件下で形式手法が要求されており、そのための「非機能条件に対する検証の調達ガイド」として、DO-178BやDO-178Cが存在している。

4.2 アーキテクチャ・ベース・ディペンダビリティの専門課題と今後の課題

今後の課題を以下に列挙する。

- ① アーキテクチャのオープン統合はTOGAFで統一。
- ② コンピュータで実行可能なアーキテクチャに関する統合標準言語の策定—AADL、Archimate^{※23}、SPEM2^{※24}の位置付けと標準化—。
- ③ トップダウン設計と並列処理を意識したアーキテクチャ分割の仕組みの確立。
- ④ UML2、SysML2、AADL2とのインタオペラビリティ

を持つモデル化でTOGAFソリューション・トレーサビリティを確保。

⑤ 抽象解釈モデルに必要な上位モデルと下位モデルとのディペンダブルな関係のモデル化、そして抽象解釈の定式化と自動化への挑戦。

⑥ 設計から検証までアーティファクト（生成物）を一貫管理する設計支援・検証レポジトリの確立。

5 提言：2030年に向けアーキテクチャ・ベース形式手法の産業戦略提案

5.1 製造産業の高品質ブランド戦略

最終目標としては、世界の高品質機器産業と先端成長各国の先端的インフラ事業を日本が取り込むことであろう。

ものづくり日本の次世代産業政策は、自動車やロボテックの技術力の土台の上に、より高度な、高信頼性を保証するリアルタイム分野、国防、航空、宇宙に環境バイオ等次世代社会インフラ分野での超複雑で、安全性を担保するシステム開発と検証能力の分野へシフトしていくべきであろう。そのためには“アーキテクチャと形式検証によるディペンダビリティ”能力確保とブランド化を達成するための20年プロジェクトとするべきであろう。

5.2 インフラサービス産業の高品質ブランド戦略

サービスビジネスのプロセスを顧客中心主義経営に基づくEAの手法で分析し、ボトルネックを発見し、長期的なシナリオで、このサービスサブセクターごとの生産性を創造的に向上させる“サービスのインテリジェント装置化”を興すべきであろう。次世代の例を自動車で言えば、現在の高級車は1,600万～2,000万ステップのソフトウェアと80個以上のECU^{※25}を搭載しているが、次世代高級車では2～3億ステップ、将来のITS時代には宇宙・航空機と同じく10億ステップのソフトウェア開発が必要だという。業種別サービスのモデル化・形式化を通して生産性と信頼性を追及する狙いだが、IT関係では、システム・インテグレーションやサービスの形式手法を組み込んだ検証法の確立も重要な分野である。しかし、これは旧来の開発・検証方式では行えるものではない。

アーキテクチャと形式手法がシームレスに統合されたアーキテクチャ・ベース・モデル検証方式の確立と形式検証アーキテクトの育成が鍵となる。このクラスのアーキテクトを育成するには10年単位の研究・研修・現場経験サイクルが必要である。

“高信頼性は日本から”を合言葉に、20年計画で、EAアーキテクト及び形式手法検証アーキテクトを、自社SEと協力会社SE総数の3～5%とするための育成計画が必要だ。これを実現するためには、“高信頼性は日本から”という世界のマーケットに向けた産官学のプロジェクトが国家の重要政策として必要であろう。

※22 ALLOY：仕様記述言語。ALLOYはBメソッドとZ言語の合金を目指して名付けられた。

※23 Archimate：EA記述用の言語

※24 SPEM2：メタ・プロセス記述用の言語

※25 ECU：Electronic Control Unit

一般社団法人組込みスキルマネジメント協会

－ ETSS によるスキルマネジメントの体系化による人材育成を目指して－



Skills Management Association

<http://www.skill.or.jp/>

一般社団法人組込みスキルマネジメント協会
理事長

大原 茂之

組込みスキルマネジメント協会（SMA^{*1}）は、ETSS^{*2}をベースにスキルに関する研究、啓蒙、標準化等の活動に取り組んでいる。ここでは、SMA を設立した背景と活動目標等について紹介する。

1 SMA 発足の背景

2003年10月に経済産業省において約30名の識者が集まって「組込みソフトウェア開発力強化推進委員会準備会」が発足した。それから1年間にわたって方針と方針の具体化を練り上げた後、経済産業省の指導の下、IPAにソフトウェア・エンジニアリング・センター（SEC）が設置された。IPA/SECにおいては、ETSSを民間主体で利活用

出来るようにすることを目的に、表1に示す活動が行われた。

ETSSの普及促進をIPA/SECではなく民間で行う理由は表2の課題を解決するためである。

こうした背景をもって、ETSSをベースに人材を人財としてマネジメント出来るようにする当協会を2009年7月29日に発足させたのである。発足にあたっては、上記のような目的を果たせるように、経済産業省、IPA/SECによる多大なご支援とご指導をいただいた。

表1 SECにおけるETSSの活動

年	活動内容
2004	ETSSの策定作業
2005	ETSSの基本構造の確定
2006	企業の協力を得て、ETSSによるスキル可視化の実証実験を継続的に開始
2007	次世代車載ソフトウェア開発国家プロジェクトでETSS全面採用
2008	ETSS導入推進者のキャリア定義と育成準備を開始
2009	ETSSの民間への普及展開の方策を検討開始

2 SMAの活動内容

SMAでは次のような研究と活動等を行う予定である。

- ① ETSS実装手順の開発
 - ・ETSSをベースにした業務分析の手順の確立
 - ・業務分析からETSSのインスタンスを決定するまでの手順の確立
- ② ETSS活用によるスキルマネジメント手法の開発
 - ・組込み技術を展開する現場におけるスキルニーズの把握とスキルマネジメントモデル確立に向けた調査研究
 - ・スキルマネジメントモデルの開発と現場への展開に関する研究
 - ・スキルマネジメントを実施するキャリアとしてのスキルマネージャの定義
- ③ 中小企業のスキルマネジメント力向上への施策
 - ・中小企業向けのETSS啓蒙活動
 - ・中小企業向けにETSSを導入する協力体制の構築
 - ・スキルマネージャ育成支援
- ④ ETSSに基づくスキル育成・評価手法の開発研究
 - ・スキル育成のための研修教材研究開発
 - ・スキルと知識の相関モデルの研究
 - ・スキルと知識の相関モデルに基づく評価手法の研究開発

表2 ETSSの普及促進を民間で行う理由

課題	内容
課題1	ETSSの各企業への実装支援は個別的な解を作るもので、民間で可能にすべき課題
課題2	ETSSを経営指標等と連動させる実用化研究もIPA/SECの範疇を超えた課題
課題3	自動車、カメラ等製品に特化したドメインにおける人材のスキル分布の策定は、明らかに業界依存型であり、民間で解決する課題
課題4	組込み技術の魅力を発信し、若者をこの分野へ誘導する施策を講じるには、産学官の横断的な連携が必要であり、これについても民間で解決出来る課題が多い

※1 SMA: Skills Management Association, 一般社団法人組込みスキルマネジメント協会

※2 ETSS: Embedded Technology Skill Standards, 組込みスキル標準

⑤ ETSS 導入推進者育成支援

- ・ETSS 導入コンサルティングを行う導入推進者による交流組織の設置
- ・『ETSS 導入推進者向けガイド』[ETSSGUIDE]に基づく導入推進者育成研修

⑥ 企業経営へのスキルマネジメントの応用

- ・財務諸表と ETSS の関係による人材スキルのアセット化
- ・経営者向けスキルマネジメント研修

⑦ 産学連携による FD^{※3} 活動支援

- ・教育研修機関への組込み技術のカリキュラム開発支援
- ・教育研修機関における組込み技術教育力向上支援
- ・教育研修機関と企業との連携組織の設置

⑧ スキルマネジメント並びに ETSS の普及促進

- ・ETSS の国際展開によるデファクト化
- ・シンポジウム・研究会の開催

上記の活動は会員組織の拡大と並行して順次立ち上げていく予定である。

3 SMA の組織構造

SMA で予定している活動は、理事会の下に企画立案を行う幹事会を設け、その企画に基づいて、部会、研究会を設ける組織構造によって取り組んでいくことにしている。活動の展開方針は理事会で決定するが、この方針に沿った活動が企業の現場と乖離しないようにするために幹事会を置いている。既に掲げた SMA 活動を更に具体化するための企画立案は幹事会において行うことにしている。幹事会は正会員から構成されるものであり、幹事長 1 名、副幹事長 2 名を置く。幹事会の主たる役割は次の通りである。

- ・課題解決への企画立案
- ・研究部会の立ち上げと部会の運営リード
- ・各研究部会の担当ミッションの横串管理

2010 年度からの本格的な活動に向けて、2009 年 12 月に第 1 回の幹事会を開催し課題整理に取り組み始めている。

4 SMA と他の組織との関係及び連携

ETSS をベースとする SMA の活動内容と近接する組織としては、IPA/SEC と社団法人組込みシステム技術協会 (JASA^{※4}) 等がある。ここでは、この 2 つの組織を中心に SMA との関係について述べておく。

① IPA/SEC との関係

IPA/SEC は ETSS の体系維持強化を行い、SMA は ETSS の普及促進を担うという関係になる。SMA は ETSS の普及促進活動を通して得られた改善改良に関する知見をまとめ、IPA/SEC に提言することになる。

② JASA との関係

JASA は組込みシステム開発を行う企業におけるマイクロエレクトロニクスの応用技術に関する標準化の推進、権利の保護、調査研究等を行い、企業における組込みシステム技術の向上と利用者の利便性を高め、日本の産業の健全な発展と国民生活の向上に寄与することを目的としている。従って、JASA において開発あるいは普及させる技術のスキル領域に関して SMA は協力していく所存である。

こうした ETSS 普及促進への取り組みについては、正会員、賛助会員の皆様と手を携えて取り組んでいくことはもちろんである。このような会員による活動以外に、連携する活動も視野に入れている。特記することは、SMA の活動の中の産学連携による FD 活動支援である。SMA としては、組込み技術教育を大学が実施する際に障害となる事項の収集を行い、具体的な支援策を導き出し、実際の活動につなげていくことで社会貢献を行うことが出来る。そのために、社団法人私立大学情報教育協会 (JUICE^{※5}) と連携することになった。JUICE は非常に活発な組織であり、今後各大学に要求される学士力^{※6}の策定や教員の教育力向上等を目指している。

5 SMA の今後の発展

いかに優れた技術があっても、それを使いこなすスキルが存在しなければ、宝の持ち腐れである。プロジェクトをマネジメントするとき、あるいは製品戦略や技術戦略を立案するときは、常にスキルをセットにして考える必要がある。とくに戦略の立案においては、経験値を超えた未知のファクタが入り込むため ETSS のレベル 4 のスキル項目を持った人材が必要となる。このレベル 4 の人材は単なる作業員ではなく、価値を生み出す財と見るべきである。今後、SMA の活動はこうした“人的資源を貴重な財産”となるようにマネジメントする仕組みを創出し、日本を世界に冠たる“人財立国”とすべく努力していく所存である。

参考文献

[ETSSGUIDE] IPA/SEC : 組込みスキル標準 ETSS 導入推進者向けガイド, 毎日コミュニケーションズ, 2008

※ 3 FD : Faculty Development

※ 4 JASA : Japan Embedded Systems Technology Association, 社団法人組込みシステム技術協会

※ 5 JUICE : Japan Universities Association for Computer Education, 社団法人私立大学情報教育協会

※ 6 学士力 : 文部科学省の科学技術・学術審議会人材委員会で「学士課程教育の構築に向けて」としてまとめられたもので、大学の学部卒業生の分野別質保証の枠組み促進・支援することを目的とする。

九州地域組込みシステム協議会


<http://www.es-kyushu.jp/>

特定非営利活動法人九州組込みソフトウェアコンソーシアム
副理事長（株式会社ネットワーク応用技術研究所 取締役）
芦原 秀一

財団法人九州先端科学技術研究所
研究企画部
犬塚 智彦

九州地域組込みシステム協議会（ES-Kyushu^{※1}）は、組込みシステム事業に係るネットワーク形成、人材育成、競争力・技術力の向上による新事業・新産業の創出をもって九州経済の発展に資することを目的に、「産学官」一体の組織を構築し、関連組織・企業の連携、情報共有、情報発信、販路拡大等を行っている。

1 背景と経緯

1.1 組込みシステム

昨今、自動車、家電製品、携帯電話等、我が国経済の発展を牽引してきた製造業及びその製品群の電子化が顕著であり、それに伴いソフトウェアの占める比率が増大した。そして、これらの製品群の機能や性能は、搭載される組込みソフトウェアの品質・性能に大きく依存し、また、中国、インド等の国々でのオフショア開発の脅威にも晒されるため、世界的な大競争時代に突入している。更にこれらの組込み分野は新エネルギーや環境領域にも裾野を広げており、その開発需要と重要性はソフトウェアの開発規模の巨大化・複雑化・開発期間の短期化とも相まって、ますます拡大されると予測されている。このように組込みシステム開発能力はソフトウェア産業のみならずあらゆる分野の産業競争力を左右するため、我が国としても、IT企業の国内立地や規模の大小を問わず、関連技術力の維持向上や人材育成が強く望まれている。

1.2 九州の地域性と活動基盤

北九州が昭和30年代後半以降、京浜、中京、阪神と並んで4大工業地帯として太平洋ベルト地帯の一角を占めていたように、九州は、かつて石炭、鉄鋼、造船等、我が国の基幹産業を担ってきた。

その後、我が国の高度成長と共に半導体と自動車が主力産業になっているが、いずれも製造業が中心で研究開発や設計等の知識集約型の開発拠点は、関東や関西等に集中し、ソフトウェアの仕事も下請けの形で仕事を受けるという構造になっている。このような状況の中で、半導体関連産業や自動車産業の拡大を背景に、図1に示すようにシリコンアイランドからカーアイランド、そして組込み（ET）アイランドへと組込みシステムビジネスの拡大を目指し、平成17年頃から特定非営利活動法人九州組込みソフトウ

ェアコンソーシアム（QUEST^{※2}）等、九州各地域で大学・自治体・企業間連携等の組織が個別に発足した。このような背景と組織化により、人材育成や交流事業等の活動を行ってきたが、九州地域としてのまとまった組織や取り組みがなく、九州域外からは九州地域の組込みシステム業界の特徴や強みが見え難い状況があった。

2 九州地域組込みシステム協議会（ES-Kyushu）

2.1 設立

このような活動の中、九州経済産業局や各県自治体、有志が動き始め、大学、関連企業等、約200団体が参加して、平成19年11月29日、組込みシステム拠点形成の中核組織として九州地域組込みシステム協議会（会長 牛島和夫九州大学名誉教授、平成22年1月現在会員数：308）が発足した。九州の組込み関連団体を図2に示す。

九州の潜在的な産業の強みを生かしつつ、域内の組込みシステム関連企業等のさらなる発展を遂げるためにも、大学や産業の得意分野の結集による「強み連携」が必要不可欠であり、九州地域が一体となって人材育成を行う他、販

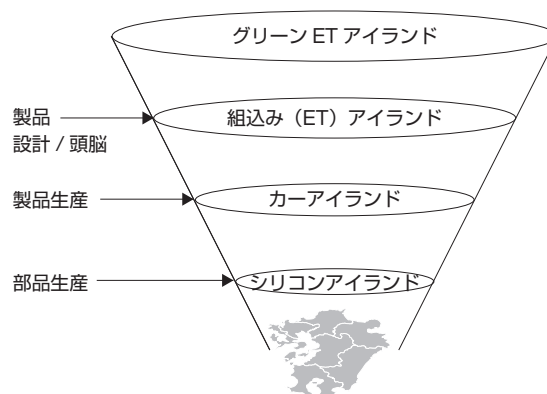


図1 九州の組込みビジネスの目標

※1 ES-Kyushu：Embedded System Association of Kyushu，九州地域組込みシステム協議会

※2 QUEST：Kyushu Embedded Software Technology Consortium，特定非営利活動法人九州組込みソフトウェアコンソーシアム

路開拓における弱みを克服し、個々の強みも生かす組織として活動を推進している。

2.2 最近の活動内容

九州地域イノベーションパートナーシップ九州ITパートナー事務局と事業を共催し、下記のような活動を展開。

(1) 連携促進事業

これからの低炭素化（グリーン）社会の実現に向けて、ET（Energy & Embedded Technology）を活用するための「組み込みシステム“グリーンET”セミナー」を熊本会場（平成21年10月28日）、長崎会場（平成21年12月3日）、福岡会場（平成22年2月4日）で実施。スマートグリッド技術、スマートハウス、デジタル電源制御等についてのセミナーを開催した。また、Android OSについてのセミナーを「鹿児島組み込みセミナー」にて実施、及び九州内での組み込みシステムビジネスの活性化を目的として全九州で行う「九州組み込み団体円卓会議 in 2009」を平成21年度内に実施予定である。

(2) 人材育成事業

「ETロボコン2009九州地区大会」をJASA※3九州支部、QUESTと協同で開催（平成21年9月6日）（図3）。九州地区からの代表が、2009年度の全国大会であるチャンピオンシップ大会にNXT部門の走行で準優勝、RCX部門で総合優勝した。

また、後述する会員提案型事業の1つである「九州プロダクトライン推進部会」の活動の1つとして、導入に向けての基本的内容のセミナーとしてプロダクトライン普及セ



図3 ETロボコン九州地区大会

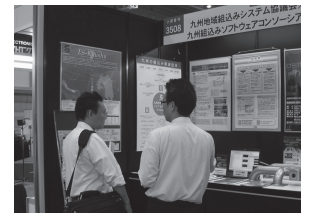


図4 AT International 2009 展示会

ミナーを大分会場（平成21年10月29日）、佐世保会場（平成21年11月27日）、宮崎会場（平成21年12月18日）で実施し、実習も含む実践セミナーを福岡会場（平成22年1月15日、22日、29日）にて行った。

(3) 情報提供事業

カーエレクトロニクス専門の展示会であるAT International 2009（神奈川県・パシフィコ横浜）に出展（図4）、第11回組み込みシステム技術についてのワークショップであるSWEST11（石川県・ホテルアローレ）等に参加した。

3 会員提案型事業

会員自らの提案による研究会や事業を行うための公募制度を整えたところ、「九州プロダクトライン推進研究会」が申請プロセスを経て採択された。これは、会員提案型事業の1つとして「九州プロダクトライン推進部会」となった。この推進部会は、平成20年夏から準備会議を重ね、九州ブランドの1つとしてプロダクトライン

開発手法や技術を世界に発信することを目指し、平成20年11月、熊本での第1回九州プロダクトラインワークショップにて発足した。発起人は11団体の企業、教育機関で、座長は九州大学中西恒夫准教授である。平成20年12月5日には第1回の会合を持ち、16団体からの参加があった。①人材育成、②共同調査/共同研究、③九州のブランド力向上を事業の柱としている。平成21年11月には第2回九州プロダクトラインワークショップを北九州小倉で開催した。

4 おわりに

今後は、具体的な活動コアメンバを募り拡大することで、各種産業分野間の交流、九州地域の一体化と共に他地域間連携、IPA/SECとの連携を深めていきたい。

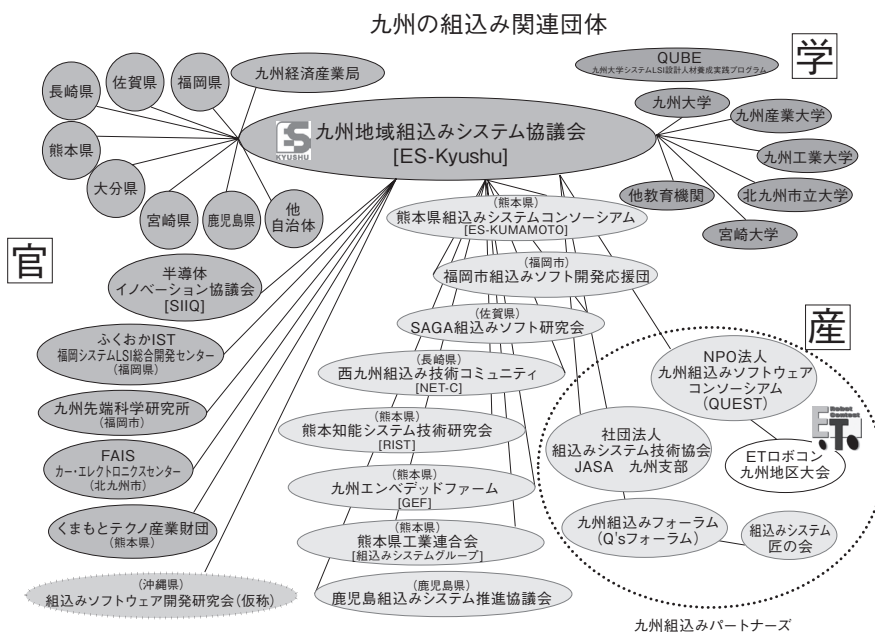


図2 九州の組み込み関連団体

出典:ES-Kyushuのホームページ: <http://www.es-kyushu.jp/kanrendantai.html>

※3 JASA : Japan Embedded Systems Technology Association, 社団法人組み込みシステム技術協会

我

タシニア世代には大変懐かしい東京・有楽町西武百貨店がいよいよ今年末で閉店になるようだ。駅の反対側に立地していた有楽町そごうは、2001年には、ビッグカメラに変わっている。大型量販店やユニクロ、H&M等のファストファッションに代表されるデフレ経済の激しい流れに対抗するには、百貨店の行動はあまりにも遅く、かつ小手先の対応に終始したように見える。

これは百貨店業界だけの現象ではない。日本全体が20年近くの長期間にわたって、世界の流れについていくことが出来ずに低迷を続けている。IT業界、ソフトウェア業界も全く同じだ。

既に本コラムでも触れたが、戦後の日本は「いいモノを安く早く作る」をスローガンに、製造輸出立国フレームで行動してきた。このフレームは、1980年代半ばまでの高度成長期には極めて効率的に機能した。日本の大量雇用を支え、海外からの輸入に必要な外貨を稼ぐことが出来た。しかし、1990年を境に、バブル経済崩壊、米国の金融ビッグバン、ベルリンの壁崩壊、グローバル市場経済の台頭、ユーロ創設等状況は一変するが、我々は古い産業構造や企業体質を温存しすぎた。

日本の1990年代の実質成長率は、年平均1.0%以下という極端な超低成長に陥り、「失われた10年」と言われた。2000年代に入ると、円安や米国の大量消費、BRICsの成長等に支えられて2003～2007年に景気を回復したが、2008年秋のリーマンショックで一気に暗転したことは周知の通りである。結果的に、ゼロ年代(2000～2009年)の実質成長率は年平均0.7%程度と見られ、日本経済は90年代からゼロ年代にわたる「失われた20年」を経験することになった。

そして2010年代が始まったが、このままでは間違いなく「失われた30年」になってしまう。では、どうすればよいのか。選択肢は3つだ。一番目は、Back to the Basic、つまり、輸出立国体質にますます磨きをかけることだ。日本にとって最も手慣れた方法だが、BRICs等の新興国との競合と為替変動リスク

という重大な問題がある。つまり、限りない低価格競争に巻き込まれると共に、今回のような円高に見舞われるたびに工場の海外移転を余儀なくされ、国内が空洞化していく。二番目は、内需型への転換だ。しかし、前回のコラムでも触れたが、今後50年で人口が約9,000万人程度に縮小するという急速な人口減少を考えると、国内市場が今後大きく拡大することは考えにくい。三番目は、米国のようなサービス立国、ソフトウェア立国を目指すことだ。米国の独壇場であった金融サービスは、今回の危機で根本的な欠陥を露呈させてしまったので、今後の日本のフレームとしては除外せざるを得ない。ソフトウェア立国は、日本の目指すべきフレームとして従来から取り組まれてきたが、Google、Amazon、eBay、Appleのようなインターネットビジネスに関しては、創造的破壊を良しとする風土や多様でオープンな文化を反映した米国の優位性を認めざるを得ない。

2010年代は我々に残された最後のチャンスなのだが、日本が置かれた状況は既に、選択肢のどれが良いという単純なものではない。中長期的に見れば、製造輸出依存から脱却しなければならないのだが、上述のように他の選択肢も切り札になり得ない状況下では、製造輸出を捨てて直ちに内需型やソフトウェア立国に転換するという解は現実的にはあり得ない。

では、どうすればよいのか。歯切れは悪いが、3つの選択肢の合わせ技を考えなければならないということだ。一番目の選択肢に関しては、生産拠点の海外移転を進めることだ。既に、キヤノンのように海外生産比率が50%をはるかに超える企業も出てきた。この動きは中小企業にも急速に広がるのではないだろうか。同時に進めなければならないのが、国内をR&D拠点化し、内需型への転換や製品の高付加価値化を支えることだ。このためには、企業の再編によって、人材と資本を集約することが避けられないのではないだろうか。

今回はソフトウェア産業について考えてみたい。皆さんのご意見を tsuruho@ipa.go.jp まで、ぜひともお寄せいただきたい。

Column

最後のチャンス!! 2010年代

IPA 顧問 学校法人・専門学校 HAL 東京 校長
鶴保 征城 (つるほ せいしる)



モジュール化の終焉 統合への回帰

田中 辰雄 著

ISBN : 978-4-7571-2250-5
NTT 出版刊
A5判・288頁
定価 3,780円 (税込)
2009年11月刊

潮目を立証する懸命のエビデンス提示

近年のIT産業の大きなトレンドを冷静に分析し、懸命にエビデンスを揃えて、モジュール化の終焉の立証を試みている。帯封に『「オープン化」は終わる?』と記されている。多くの人がなんとなく感じている最近の潮目の変化を明快に描き出している。立証の素材として誰でもが親しいワープロの歴史、そしてパソコンを取り上げている。その立証の方法は大変精力的で、客観性を確保しようとする著者の努力には頭が下がる。

考えてみれば、あのとてもエキサイティングだった1990年代はモジュール化とオープン化の時代だった。しかしこれは日本の産業界にとっては不得意領域だったと言えよう。この時代背景はその中にいると自明、そして永続的な産業環境のように思われたが、著者

によれば、むしろこの時代は特異であったと位置付けられている。

それが今、大きな潮目で、すっかり仕切り直しの時代を迎えたようだ。副題にもあるが、日本の産業界が優位を築いていた1980年代への単純な回帰は無いにしても、日本にとって、とても楽しい時代に突入しつつあることを示唆してくれて元気が出る。最終章に統合化の帰結として考えられることが11項目にわたって整理されている。例えば、摺り合わせ型企業の優位、垂直統合、企業グループの形成、スイッチングコスト、英雄時代の終わり、職人時代の幕開け、といった項目が並ぶ。これからソフトウェア産業を考えるときに、基点の1つとしてとても刺激される書である。

(神谷芳樹)



ものづくり敗戦 「匠の呪縛」が日本を衰退させる

木村 英紀 著

ISBN : 978-4-532-26036-1
日本経済新聞出版社刊
新書判・256頁
定価 893円 (税込)
2009年3月刊

日本型「ものづくり」の限界を明らかにする警告の書

本書は「ものづくり敗戦」や「匠の呪縛」等の衝撃的な文字が躍る。産業革命以降の歴史を紐解きながら、科学と技術の関係性を整理して話が進む。黒船来航以降の日本に関しては、ネガティブな話ばかりで気が重くなる。日本は欧米諸国のキャッチアップを必死に行うが、欧米諸国は資本集約による効率的なものづくりに対して、労働集約的なものづくりであるという。戦後の大量生産時代において“カイゼン”という全員参加の品質活動は、日本のお家芸であり日本経済を支えたと言っても過言ではない。

しかし、インターネットが普及しグローバル社会となった現代においては、日本が最も苦手な理論、ソフトウェア、システムという3つが重要な時代となっている。暗

黙知に依存し理論的でない開発現場や、そしてソフトウェアやシステムは欧米に比べ弱い事実等、これからの日本がどう戦っていくのかを考えさせられる。

技能としてのスキルは重要であるが、それ以前に理論をきちんと学び理解することの重要性を再認識した。それは従来からある自然科学だけでなく、近代の情報理論やシステムや人間等の複雑な問題領域を対象とした科学も対象である。また理論に基づき、人材育成や自動化等による開発の効率化が出来る人材が求められ、労働集約型開発を改善しなければならない。

先人が築き上げた「ものづくり神話」を逸話とするか否かは、学校教育を含めた人材育成と、開発現場の労働集約型開発からの脱却が必要である。(渡辺 登)

編集後記

新しい年となりました。華やかなオリンピックでは、多くの国民の目が一点に集まる場面があります。今年のお正月は、ソフトウェアエンジニアリングの領域で、これにちょっと似た、識者の耳目が一点に集中した場面がありました。本誌18号にもご登場いただいた株式会社東京証券取引所殿の新システム、アローヘッドへの移行です。多くの方々の大変なご努力によって目標は見事に完遂されました。関係の方々には、ちょうどスキージャンプ競技の国別対抗戦の選手のような、大変なプレッシャーだったものと推察されます。心より拍手をお送りしたいと思います。

新年になってもう1つビッグイベントがありました。ここ3年間、経済産業省の委託事業として自動車産業界で進められてきた車載共通ソフトウェアの開発プロジェクトの成果発表会です。この開発はSECで推奨させていただいているソフトウェアエンジニアリングの各種手法を総動員して進められました。乗用車にはLAN接続された60個ものマイコンが搭載され、運転席のステアリングはもはや歯車のような機構で車輪に接続されることは無く、車輪はソフトウェア制御のモーターで直接制御されています。こうした自動車が、発表会当日実際に近藤洋介 大臣政務官を乗せて目の前を快走する姿を見て、国民生活に浸透するソフトウェアの時代の到来、ということを実感しました。本誌でもこうした新時代の潮流をしっかりと捉えてゆきたいと考えています。 (神)

編集部 よ り

毎回「申込みをしても、すぐ満席になる」とお叱りの声をいただくSEC主催セミナーですが、1～3月は回数を増やして開催いたしました。少しはご要望にお応え出来たのではないかと考えております。また、今号からSECの研究成果を導入していただいている企業様の事例を、取材してご紹介させていただいております。こうした事例は、5月開催の「ESEC/SODEC」のブース内でもご紹介させていただく予定です。ぜひご来場ください。

SEC journal 編集委員会

編集委員長	神谷 芳樹
副編集長	渡辺 登
編集委員 (50音順)	遠藤 和弥
	矢野 亜希



春の鴨 (三溪園)

(撮影：神谷芳樹)

SEC journal® 第6巻第1号 (通巻22号) 2010年3月31日発行

© 独立行政法人 情報処理推進機構 2010

編集兼発行人 〒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

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 所長 松田 晃一

Tel.03-5978-7543 Fax.03-5978-7517

http://sec.ipa.go.jp/

編集・制作 〒101-8460 東京都千代田区神田錦町3-1 株式会社オーム社 Tel 03-3233-0641

※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

IPA（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・センターでは、
下記の内容で論文を募集します。

応募様式は、下記のURLをご覧ください。
<http://sec.ipa.go.jp/secjournal/papers.html>

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

- 開発現場への適用を目的とした手法・技法の詳細化・具体化などの実用化研究の成果に関する論文
- 開発現場での手法・技法・ツールなどの様々な実践経験とそれに基づく分析・考察、それから得られる知見に関する論文
- 開発経験とそれに基づく現場実態の調査・分析に基づく解決すべき課題の整理と解決に向けたアプローチの提案に関する論文

論文の評価基準

- 実用性(実フィールドでの実用性)
- 可読性(記述の読みやすさ)
- 有効性(適用した際の効果)
- 信頼性(実データに基づく評価・考察の適切さ)
- 利用性(適用技術が一般化されており参考になるか)
- 募集テーマとの関係

応募要項

投稿締切り

年4回、3ヵ月毎に締切り、締切り後に到着した論文は自動的に次号審査に繰り越されます。

(応募締切:1月・4月・7月・11月各月末日)

締切り後、査読結果は1ヶ月後に通知

詳細スケジュールについては、投稿者に別途ご連絡いたします。

提出先

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・センター内 SEC journal 事務局

eメール: sec-ronbun@ipa.go.jp

その他

- 論文の著作権は著者に帰属しますが、採択された論文については SEC journalへの採録、ホームページへの格納と再配布、論文審査会での資料配布における実施権を許諾いただきます。
- 提出いただいた論文は返却いたしません。

論文賞

SEC journalでは、毎年SEC journal論文賞を発表しております(今回は2008年10月28日SECコンファレンス)。受賞対象は、SEC journal掲載論文他投稿をいただいた論文です(論文賞は最優秀賞、優秀賞、SEC所長賞からなり、それぞれ副賞賞金100万円、50万円、20万円)。

論文分野

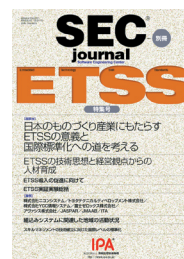
品質向上・高品質化技術
レビュー・インスペクション手法
コーディング作法
テスト/検証技術
要求獲得・分析技術、ユーザビリティ技術
見積り手法、モデリング手法
定量化・エンピリカル手法
開発プロセス技術
プロジェクト・マネジメント技術
設計手法・設計言語
支援ツール・開発環境
技術者スキル標準
キャリア開発
技術者教育、人材育成

SEC journal バックナンバーのご案内

詳しくは<http://sec.ipa.go.jp/secjournal/>をご覧ください。



No.16



ETSS特集号



No.17



No.18



ESxR特集号



No.19

SEC Journal No.20
第6巻第1号 (通巻22号)
2010年3月31日発行 © 独立行政法人 情報処理推進機構

編集兼発行人

〒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
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
所長 松田 晃一
Tel:03-5978-7543 Fax:03-5978-7517
URL: <http://www.ipa.go.jp/>



IPA®

独立行政法人 情報処理推進機構