

2006年4月28日発行
第2巻第2号（通巻6号）
ISSN 1349-8622

SEC

6

journal

Software Engineering Center

特集

SEC2005年度活動概要

IPA[®]

独立行政法人 情報処理推進機構

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



SEC journal

Software Engineering Center
No.6目次

1	巻頭言 棚橋康郎(社団法人 情報サービス産業協会 会長)
2	所長対談：畑村洋太郎 工学院大学教授・東京大学名誉教授 畑村創造工学研究所長 「失敗学」の視点から ソフトウェア開発の課題と解決策を考える
6	技術解説 ソフトウェアエンジニアリング基礎知識体系の背景と活用 松本吉弘
16	特集 SEC2005年度活動概要
	エンタプライズ系
18	エンタプライズ系活動
20	見積り手法
22	定量データの定義・収集・分析 - ソフトウェア開発データ白書2006&データ定義Ver.2.0のリリース -
24	ソフトウェア開発における要求品質の確保
26	要求工学、設計・開発技術
28	プロジェクト見える化とは - 下流工程編 -
	組込み系
30	組込みソフトウェア・エンジニアリング
33	組込みソフトウェアの組織的な開発の実現 - 開発プロセスガイドと開発計画書ガイドの策定 -
36	組込みソフトウェア実装品質向上 - 組込みソフトウェア向けC言語コーディング作法ガイドの策定 -
38	組込みスキル標準(ETSS)
40	ETSSスキル基準 Version 1.1
42	ETSSキャリア基準 Version 1.0
44	ETSS教育研修基準 Version 1.0
	先進ソフトウェア開発プロジェクト
46	先進ソフトウェア開発プロジェクト
	共同研究
48	ベイズ識別器に基づく混乱予測に利用するマトリクスの統計的選択
49	先進ソフトウェア開発プロジェクトにおけるコードクローン分析
50	組込みアーキテクチャとそのプログラミングについての調査
51	ETSS向け教育プログラム評価手続きの策定
52	デジタル製品向け組込みソフトウェア - 開発力と競争力について -
53	組込みソフトウェア教育における体験型学習の調査
54	協調フィルタリングを活用した工数見積り
55	組込みソフトウェアにおけるユーザビリティ
56	形式的手法の実用化を目指して
57	見積り手法の実証評価 - CoBRA法とOSR法 -
	地域からの発信
58	新潟県での情報産業振興の取り組み「N-SECプロジェクト」-IT産業クラスター形成を目指して- 高畑悦武
	組織紹介
62	社団法人 組込みシステム技術協会 JASA 高橋重真
64	独立行政法人 情報処理推進機構 人材育成推進部 萬井正俊
66	BOOK REVIEW
67	ソフトウェア・エンジニアリング関連 イベントカレンダー
68	あとがき
69	お知らせ(論文募集 / SEC journal バックナンバー)

巻頭言

魅力ある産業への旅立ち



社団法人 情報サービス産業協会
会長
棚橋 康郎

情報サービス産業の4課題

昨年、東京大学の情報工学系は入学志望者が少なく、定員割れをしたそうである。この情報を耳にした時、情報サービス産業に携わる者の1人として、これは由々しき事態になったと思った。なぜならば、定員割れなどという事態は、旬を過ぎた産業の関連学科で起こるのが通例であり、その原因は当該産業に魅力が欠けるからである。具体的には次のような解決すべき課題を抱えていることが、この事態を招いたと考えられる。

1つ目は、重要インフラともいえるべきシステムであるにもかかわらず、機能不全が引き起こすトラブルが絶えないことである。

2つ目は、適正さを欠いた会計処理、顧客の情報資産(含む個人情報)の流出等、次元の低いコンプライアンスに関する問題が絶えないことである。

3つ目は、人材がすべてと言いつつ、必要とする人材像を描ききれておらず、必要な人材像を世の中に発信できていなかったことである。

4つ目は、日本語のバリアに守られた、ぬるま湯的市場しかなく、合理的な競争の場がないことである。

情報サービス産業は変わろうとしている

さて、情報サービス産業の最大で唯一無二の経営資源である高度IT人材の補給が先細ってしまうことは、産業の衰退を暗示するものと受け止めなければならず、益々気分は憂鬱になる。

しかし、我々の提供する情報システムが社会、経済、

産業の中で、いかに重要な基幹的役割を担っているかを考えれば、三段論法から導き出せる悲観的結論にそう容易に同意を与えるわけにいかないし、してはいけないと思う。

そのためには、情報サービス産業が魅力を取り戻さなければならない。

先の4つの課題を率直に認め、その解決に積極的に取り組み、その進捗状況や成果を発信することが、直接、間接を問わず、情報サービス産業に関係を持つ人々に、「情報サービス産業が変わろうとしている」ことをアピールすることに繋がり、情報サービス産業への関心と呼び覚ますことになるだろう。

課題解決に向けて

こういった取り組みは、本来、産業界自らが、主体性を持って行わねばならないのは当然である。しかし、その解決には、あまり時間が残されていないし、主体性を持って取り組むための産業界の組織再編なども遅れている。また、産業界の力だけでは解が見つからない、困難な課題もある。

その典型が、1つ目に挙げた「情報システムを構築する組織的ソフトウェア・エンジニアリング力の向上」と3つ目に挙げた「IT人材の類型化とそのスキル・スタンダードの提示」である。

この2つの課題は、幸いIPAが喫緊の課題として取り上げ、SECのイニシアティブの下で、産官学連携によって着々と推進されており、関係者として大いに期待しているところである。

情報サービス産業が魅力ある産業へ脱皮するまでには、当分の間、苦しい時期が続くと思うが、IPA/SECの力強い支援を引き続き期待するとともに、当協会としても最大限の努力を持って取り組んでゆきたいと考えている。

「失敗学」の視点から ソフトウェア開発の課題と解決策を考える

企業の情報システムも組み込みソフトも、開発の短期間化が急速に進行している。そのなかで、ソフトウェア開発の現場では高い信頼性を確保するというミッションを果たしていくことが求められている。そのために、取り組むべきことは何か。「失敗学」を提唱し、産業界から広く注目を集めている工学院大学教授・東京大学名誉教授の畑村洋太郎氏とIPA/SECの鶴保征城所長がソフトウェア工学やソフトウェア開発プロジェクトの問題点について語り合った。



鶴保 征城(つるほ せいしろう)
1966年大阪大学大学院工学研究科電子工学専攻修士課程修了後、同年4月日本電信電話公社(NTT)入社。1989年11月NTTソフトウェア研究所長、NTTデータ通信株式会社常務取締役技術開発本部長、NTTソフトウェア株式会社代表取締役社長を歴任し、2004年10月独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター所長に就任。高知工科大学工学部情報システム工学科教授(2003年～) 奈良先端科学技術大学院大学 客員教授(2003年～) 独立行政法人 日本学術振興会「基盤的ソフトウェア技術開拓」に関する研究開発専門委員会委員(2003年～) 社団法人 情報処理学会 会長(2001年～2002年) XMLコンソーシアム 会長(2001年～) 社団法人 電気情報通信学会 フェロー 社団法人 情報処理学会 フェロー

鶴保 経済産業省は以前からソフトウェアの振興策に取り組んできましたが、その結果、日本のソフトウェア産業の競争力は決して満足できるものではありません。問題は大きく2つあります。1つは輸出入の差が非常に大きいことです。パソコンOSが典型ですが、日本はソフトウェアの輸入が大きく、輸出が少ない。輸出入の比率が1対30近くにもなっています。もっとクリエイティブなソフトウェアを作ってグローバルに展開しなければいけない。もう1つの問題は、東証システムの事故に見られるように、ソフトウェアの品質が不十分なことです。この2つの問題意識で業界としてソフトウェアエンジニアリングに真面目に取り組もうということでSECが設立されたのです。

畑村 まず、第1の点に関して申し上げます。日本は、知的所有権に関してはずっと赤字です。知的所有権を輸入しても、それを利用して上手にものを作るから日本の技術はすばらしい、それが技術力だといって引っ張ってきた国全体の考え方が間違っていたのではないかと。しかし、鉄鋼業だけは、ある種の戦略を考えていたので、相当早い時期から自前の技術開発に取り組んだのです。その典型が制御圧延です。これは機能の作り

込みという技術です。同じ素材でも、ある条件である加工を行うことで50の力しか出ないものが80の力が出る材料が作れる。それを使うと、薄くて軽い自動車用の鋼板ができるのです。鉄鋼業は、材料が「目方勝負」から「機能勝負」に切り替わることを知っていて時間をかけて技術開発をしてきたのです。

それに対して鋼板のユーザである造船業は、船体だけを作り、エンジンはヨーロッパのメーカーのものをライセンス生産をしていました。安物の船を造っていた日本の造船業は、1973年のオイルショックで大きな打撃を受けたのです。ところが、鉄の生産量はこれまでの35年間、ずっと1億トンを持っています。その背景には、前述した技術革新があります。機能を売る方向に切り換えた産業は貿易黒字になっているのです。

鶴保 現在、企業の情報処理システムや組み込み、システムの高度化・複雑化が急速に進んでいます。しかし、これらを支えるソフトウェアの品質が不十分であり、従来以上に、高信頼で高生産性で作らなければならないという課題が突きつけられています。

畑村 ソフトウェアは、極めて戦略的に考えるべきものだと思います。しかし、本当の意味での戦略は、だれも考えないまま、量的なもので勝負することがよいのだという思想のままで進んでいるように見えてしょうがない。典型的なのが総合電機産業の一部門としてコンピュータシステムや巨大システムを作っている人たちです。彼らはその思想を変えなければ、半導体と同様にコンピュータシステムでも負けが見えています。それはなぜかということ、人材とお金の投資をしなくてはいけない時期に、従来型の重厚長大の投資方法しか考えられない人たちがトップにいるからです。

鶴保 従来型の投資とは、どのような投資なのですか。

畑村 確実に収益が上がるような投資です。リターンは大きくないけれども確実に収益が上がるような投資をして、社会の先を見て賭をするような投資はしていない。社会インフラに関わるような領域では賭をして欲しくないのですが、将来産業にな

るようなビジネス領域では賭をすることが必要です。ということが起こるかかわからない、めったにうまくいかないけれど、うまくいったときには大きな収益を得られるようなものを、きちんとかけ算をして、期待値の大きさや、どの分野で行うべきなどの戦略を練る人材が日本にはいないまま産業が動いているように思えます。

いいソフトウェアをつくるための基本は 「現地」「現物」「現人」

鶴保 昨年には、東証の問題が発生し、ソフトウェア業界としては、どうすればそういう失敗を繰り返さないで済むかという問題に直面しています。

畑村 東証のトラブルは、起こるべくして起きたと思います。証券会社の入力ミスに批判する声もありますし、誤発注を止めようとしたのに止められない東証のシステムを批判する声もあります。しかし、私からみると両社とも間違った方向へ向かっているように思えます。証券会社は、あのようなミスをこれまでも起こしてははずです。誤発注が起きないようなシステムを作らずに、東証が止められなかったと、東証を責めるのは間違っています。一方、東証も入力ミスを訂正する仕組みを作らなかったのも問題です。

鶴保 証券会社によっては誤入力を防止する仕組みを導入しているということですか。

畑村 もちろんです。3年くらい前に外資系の証券会社が誤発注し、大きな損失を出したことがありましたが、それをきっかけに人的要因による失敗の研究会を約3年間にわたって開催しました。その研究会のメンバーが勤務する外資系の証券会社では、誤入力で大きな損を出し、システムを見直して入力ミスがなくなったということです。しかし、他の証券会社は、システムの見直しをしてこなかったのです。システムの担当者ではなく、トップがわかっていると失敗するのです。そして、「現地」「現物」「現人」で考え行動する人を育てるのが大切なことです。

鶴保 ソフトウェア業界の問題点は、他の製造業や流通業と異なり、データを集めて分析し、経営のベースとするという習慣がないということです。このため、プロジェクトは、個人の器量にまかされて、部分最適に進めているというような状況です。全社的な仕組みができていない。全社的なメカニズムが構築され、その上で現場が創意工夫を発揮するのが基本的な企業経営のあり方だと思いますが、そうなってはいません。

畑村 企業のトップには、現場の要求を基づきシステムにもつ



畑村 洋太郎
(はたむら ようたろう)
1941年東京生まれ。東京大学工学部機械工学科修士課程修了。東京大学大学院工学系研究教授を経て2001年退官。現職は東京大学名誉教授、工学院大学グローバルエンジニア学部教授。
【専門分野】
専門は、ナノ・マイクロ加工・カセンサ・加工の知能化・医学支援工学・技術の伝承と教育の方法論を研究するほか、失敗知識活用研究会を通じて失敗学の構築を行っている。また、実際の設計研究会を主催し創造設計原理の研究を行っている。
【主な著作】
「実際の設計」「続-実際の設計」「続々-実際の設計-失敗を生かす」「失敗学のすすめ」「機械創造学」「直観でわかる数学」等。

てくる技量が必要だと思います。セブン-イレブンの鈴木さん(敏文会長)は、店に来たお客ではなく、来なかった人を見てどうするかを考えるのが本当の商売だと言っています。それを数量化して判断するのが本当の商売だと。

鶴保 ソフトウェア業界では、鈴木さんが言われる、仮説を立ててデータで検証するという手法ができていません。データを集めて分析するのと同じように、ソフトウェア業界からの反応はデータを分析したらどのような役に立つのか、どれ位利益に結びつくのかというようなものです。ソフトウェア業界には、自ら仮説を立て、それをデータで検証するという能力がなくなっているのです。

畑村 それは、計算屋さんになっているということですね。あるコンピュータシステムがあり、「そのシステムを動かすので、何か持ってきてください、そうすればアウトプットを出します」という態度になっているのですね。それは、私も感じます。単なるソフトウェア屋になっているということですね。

鶴保 どうすればよいでしょうか。

畑村 本当によいソフトウェアを作ろうと思うなら、商売の末

「失敗学」とは

失敗はマイナスをもたらすが、その失敗を上手に生かすことによって大きなプラスを得ることが可能となる。失敗に学ぶは創造へのヒントともなるし、またミスを防ぐことにも効果をもたらす。東京大学で機械工学を教える過程で畑村洋太郎氏が、失敗とうまくつきあうために、失敗の定義から失敗の生かし方など、失敗を総合的に整理・体系化したのが「失敗学」である。

端へ行くことです。現地、現物、現人が必要です。例えば、コンビニエンスストアのソフトを作ろうと思うなら、まずは、コンビニエンスストアへ行き、レジに立ってみることが必要です。そして、何がどのように動いているかを見ることです。

プロジェクトマネージャに求められるのは 個の独立

鶴保 大学教育に対してソフトウェア業界が問題視しているのは、大学におけるソフトウェアの教育が適切に行われていないということです。東京大学では電気工学科の人気の低下しています。2005年6月に経団連は、大学には期待できないため、経団連が自前で先生を大学へ送り込むので、大学は受け入れる準備をしてほしいという提言をしました。現在、文部科学省と連携して検討が進んでいると聞いています。現在の大学の先生は、大学で育ててきた人が多いため、産業界で起きている新しいことを教えられないという意見がありますがどう考えられますか。

畑村 おっしゃるとおりだと思います。大学は、新しいものを生み出すのだと言いながら、古いものを繰り返し教えているだけで、考え方を根本的に変えるという努力をしているようには見えません。東大の機械科は伝統的に、外部からDNAを取り入れようという考え方を持っており、日立製作所で働いていた私が先生として迎え入れられました。これと同じような人事は現在でも行われています。一方で、このような動きをとれなくなってしまう大学もあり、そのような大学は学生も集まらなくなっています。

鶴保 比較的論文を書きやすい分野では、産業界出身の先生が出やすいと思いますが、ソフトウェアの場合には論文を書くこ

とが難しいので、博士号を取得できず、大学ではポストを得にくいという問題があります。

畑村 それは、大変困ったことだと思います。論文だけでなく、ソフトウェアを本当に審査できる人材がないということにもつながります。

鶴保 文部科学省が基準を変えるか、講師として産業界のプロフェッショナルを呼ぶ、経費を出すといったような仕組みを設ける必要があると思います。

畑村 一方で、産業界に教えられる人がいるかという問題もありますね。コンピュータシステムを構築するということは、どういうことかという基本から考えると、最初に要求機能があって、その要求機能を細かく分析すると、それに付随する制約条件がしだいに明らかになります。制約条件をもちながら、機能の分解や分析を行い、単位機能にまで落とし込んで、個々の単位機能を実現する解決手段を考える。1個ずつの課題要素を実現する方法は数多く考えられますが、1個ずつの利害特失も考え、全体についての解決策をトータルとして見たときに、どう見えるかという総合判断をしながら、その解決策を具体策にまとめる。そして、具体策を全部まとめて実現可能な計画にするんですね。したがって、最初の半分は分析、残りの半分は統合・総合になんてですね。企業にいる多くの人ができるのは、分析して解決策を見つけるところです。それらを統合しながら、きちんとした解決策にして、全体計画にまで持っていくことができる人は、ほとんどいません。とくに巨大システムを考えられる人は、1社に数人しかいないと思います。

鶴保 日本人の場合は、分析は得意なのですよね。

畑村 そうです。そして、具体化するところまでは、本当に詳細に作業します。しかし、その方法の最大の欠陥は、想定外が起こることです。したがって出来上がったら、フィードバック

をかけて、全体を見直す必要があります。スパイラルアップです。ソフトウェアをきちんと作る時も同じです。システム構築するためのプロセスとしての哲学が日本にはないと思います。

鶴保 我々がぶつかっているのは、まさにそういう問題です。

畑村 これを、きちんとした学問にして、きちんと説明することができる人はいません。私は、それが必要だと思い、『実際の設計 <第5巻>』（日刊工業新聞社）を出版しましたが、その中に新しい巨大システムを作るとき「思考展開図を活用したソフトウェア設計手法」という文章があります。そういうことに気が付いた人が来れば、話をしますが、来ない人たちは気付いていないのだと思います。

鶴保 ソフトウェア開発の仕事は、プロジェクト自体が小さくなっており、数十億円が数億円、数億円が数千万、期間も2年が1年、1年が6カ月という状況です。そして、プロジェクトマネージャが非常に大勢います。そのプロジェクトマネジメントのばらつきが非常に多いことが問題です。

畑村 その問題をきちんと解決しない限り、根本的な改善にはならないでしょう。下士官ばかりでは、戦争に勝つことはできません。

鶴保 しかし、今は、下士官だけで仕事がうまく行くような方法が求められているのですが、そろそろ限界ですね。プロジェクトマネージャ全体の質を上げる必要があるのです。

畑村 それには、だれかが決めたことを組織の歯車として正確にするのがいいという今までの教育を引継ぎ繰り返さないといけない。「個が独立」することです。まずは、「個」が自分で考え、次に「集団で共有」することです。日本は、高度成長期に個で考えるということをやめて、形の上では個で考えたことについていながら、いつでも集団でものを考えるか、集団でものごとを決め、その通りに動くという運営だけを行ってきました。その結果、品質のいいものができるようになったので、それがいちばんいいと考え、小集団活動や提案制度をすすめ、形だけになっているんです。その上でQC活動、TQC活動、ISOの導入をしている。そうしたことは、すべて正しいし、すべて必要なことだと思います。しかし、それを形骸化させ、なぜそういう決まりになっているか、外れたらどうなるか、違う道を考えるようにせず、ただこの道を歩くことだけを教えてきました。そのため、言っていることは正しいのに、それに従うと失敗してしまうということになってしまいました。真剣にTQC活動を行った会社はダメになり、ほどほどに行った会社が伸びているという状況です。両者の違いは、一人ずつ考えることをしたかということです。

鶴保 ソフトウェア分野ではプロセス改善という地道な活動を

しているグループがあり、年に数回、かなり大きな会合を開いています。私もそれらの会合に参加しているのですが、そこでの発表によく出てくるのは、「形骸化」と「我が社は部分最適で」という2つの言葉です。

畑村 それを改善する方法はあると思います。考え方の根本をトップから順番に変えていくことです。僕は、企業の経営者に求められて「畑村塾」を開いています。そこでは、僕がクリエイションと失敗の話をして、社員から社内で発生した大きな失敗を話してもらいます。最初にいうのは、僕が教えた子ネズミの人たちが自分の職場に帰ったら「孫ネズミを作りなさい」ということです。しかも、状況や経験がみんな違うのだから教えることが違ってないといけません。僕が教えた中身のオウム返してはダメだ、自分で考えなさいと。先日、孫ネズミの発表を聞いたら僕が教えたよりも中身がいいんですよ。その人たちが自分で真剣に考えているからでしょう。いまは、ひ孫ネズミまでいます。

鶴保 その「畑村塾」の仕組みは、小集団活動とやり方は同じなのでしょう。

畑村 似ているかもしれませんが、大切なのは、個の独立ということです。他人と相談するとダメになるので、他人と相談してはいけません。その点が小集団活動と全く異なります。一人で考え、一人で作り上げる。そこまでいってから、議論して解決策を作っていけと指導しています。一人だと10個気付くべきところが3個しか気付かない。しかし、3個しか気付かない社員が5人集まれば、重複を除いても7個程度までになります。そうすると、残りの3個を自発的に見つけ出すことは簡単です。ある会社でメンテナンス部門の社員が「畑村塾」のやり方で考え始めたら、「これをやると会社が儲かる」というんです。それは、同じミスをしなくなったからです。先にチェック項目を決めるのではなく、メンテナンスを担当する社員一人ずつが、自分の目で見えるようになるので、問題点に気付くのです。

鶴保 ソフトウェア産業にも、自分で考えることが求められているんですね。技術的なだけを追求してもブレークスルーは難しいですね。

畑村 そうですね。ひ孫までの教育がきちんとできれば、不良率が半分くらいに減るのではないかと思います。そのためにも、SECには、人材育成に役に立つ成果提供を期待しています。その成果が普及すれば、コンピュータシステムの信頼性も高めることができるでしょう。



文：小林 秀雄 写真：越 昭三郎

ソフトウェアエンジニアリング 基礎知識体系の背景と活用

財団法人 京都高度技術研究所 顧問
松本 吉弘

ソフトウェアエンジニアリング基礎知識体系 (SWEBOK : Guide to the Software Engineering Body of Knowledge: ISO/IEC TR19754) に関して、これを国際標準とするに至った背景及び標準化のために辿ったプロセスを省みることによって、その目的を解説し、これが今後、どのように活用されていくかについて展望する。

背景

1.1 「エンジニアリング」に対する解釈

ソフトウェアエンジニアリングが正統な (legitimate) エンジニアリング・ディシプリンであり、かつ社会的に認められた (recognized) 専門職業 (profession) であるという認知を得るまでの道のりはきわめて険しいものであった。Starr (医業の歴史を著し、ピューリッツア賞受賞) [STARR1982]は、専門職業がこのような認知を得るためには、その職業領域において、次の3つの要件を満たす必要があるとしている。

職業人が持つ当該専門分野に関する知識及び適格性に関する妥当性を、同業者コミュニティの中で判定できる環境が作られていること

同業者コミュニティによって妥当であると確認される知識が、合理的で正当な科学的な基盤に基づいていること

職業人が果たす判断、行為、助言が、社会の中で実質的な価値の集積を形成すること

「エンジニアリング」に対する適切な邦訳を見つけることは難しい (日本技術者教育認定機構¹では、これを「技術業」と訳したことがある)。IEEE² Computer Society, The Software and Systems Engineering Standards Committee (S2ESC) の前身である The Software Engineering

Standards Committee (SESC) では、エンジニアリングを、図1に示したモデル [SESC1993] によって解釈し、このモデルに基づいて、モデル各部を構成する部分要素の標準化を進めてきた。

1.2 「ソフトウェアエンジニアリング」に対する解釈

IEEE Computer Society は、ソフトウェアエンジニアリングを次のように定義している (IEEE Std.610.12-1990)。

ソフトウェアの開発、運用及び保守における、システムティックであり、ディシプリンに基づいた、定量的

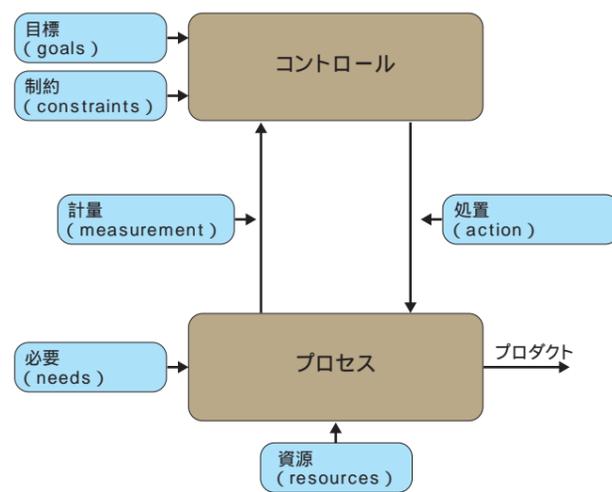


図1 エンジニアリングモデル [SESC1993]

なアプローチの適用である。換言すれば、ソフトウェアへのエンジニアリングの適用である。

で示したアプローチに対する研究である。ここで、ディシプリンとは、方法論に基づいた教育・訓練によって形成された規律であると解釈する。

SWEBOK (Guide to the Software Engineering Body of Knowledge) は、ソフトウェアエンジニアリング知識体系 (Software Engineering Body of Knowledge) への「ガイド」を指し、両者を区別するために、ガイドをソフトウェアエンジニアリング「基礎」知識体系と訳している。「基礎」を付していない「ソフトウェアエンジニアリング知識体系」は、ガイドの中で参照または推奨されている文献、図書で体系化された全知識を指し、SWEBOKは全知識体系の中の「一般的に認められた知識」だけを選定して表示し、読者がその知識へアクセスするための手引きであることに徹している。

文脈によっては、前者を、「Guide to SWEBOK」、後者を「SWEBOK」と呼ぶ場合もある。本解説では、ガイドをSWEBOKと呼ぶことにする。

SWEBOK2004 [ISO/IEC TR19759]によれば、ソフトウェアエンジニアリングは、すでに正当なエンジニアリング・ディシプリンとして社会的に認知されたものであると記されている。その理由として、ソフトウェアエンジ

ニアリングは、エンジニアリングが正当なエンジニアリング・ディシプリンであると認知されるために必要とされる、次の基準を、ここ数年の間に満たしたからであるとしている。しかし、我が国では、必ずしも満たされたといえる状態にはない。

- 教育認定 (accreditation) に合格した初期専門職業教育が行われていること
- 資格認定 (certification) または免許 (license) が実施され、実践に対する適格性を持ったエンジニアを登録する制度が実施されていること
- 専門に特化したスキル開発及び生涯教育を実施する組織が存在し、機能していること
- 専門職業団体が存在し、所属する個人及びコミュニティの活動が支援されていること
- 専門職業団体の中で倫理要綱が規定され、その実施が約定されていること

1.3 SWEBOKが果たす役割

工学には、自然法則に基づいて構成される自然系工学と、数学や認知科学によって形成された普遍的原理・原則に基づいて構成される人工系工学がある。ソフトウェアエンジニアリングは、後者のエンジニアリング系に属する。

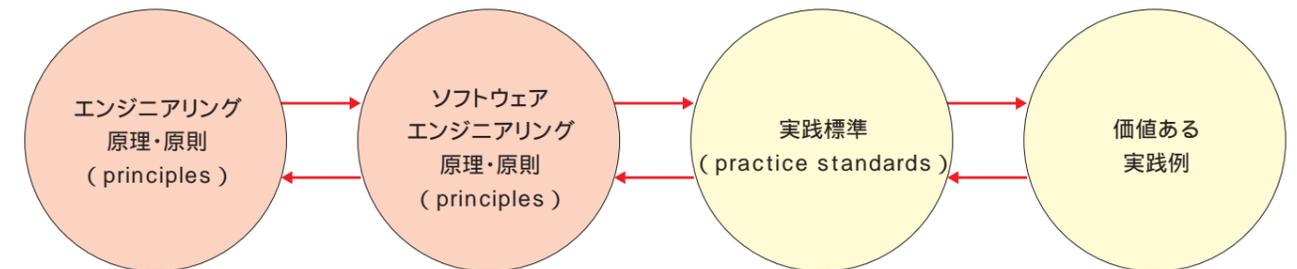


図2 原理・原則 / エンジニアリング / 実践の関係

1 <http://www.jabee.org/>
2 The Institute of Electrical and Electronics Engineers, Inc.

ソフトウェアエンジニアリングが基づく原理・原則 (principles) は、図2に示すように、抽象的な原理・原則だけではなく、標準化され、現場で実現できるものでなければならない。実践標準は、現場でこれを参照し、実践することによって、なんらかの大きさをもった価値を生むようなものでなければならない。右端の価値ある実践例を「ベスト・プラクティス」と呼称することがある。ベスト・プラクティスという用語は、IAAMC³などで公式に使われているが、ソフトウェアエンジニアリングでこの用語を用いることには、何が「ベスト」か、判定するための基準が明確でないとして、疑問視する向きもある [HARRISON2004]。過去半世紀で、ベスト・プラクティスと喧伝された実践方法、言語、ツールの中で、価値を生むことなく、はかなく忘却の彼方に押しやられたものは、枚挙にいとまがないほど数多いことが、疑問視の根拠となっている。

SWEBOKは、次節で述べるように、図2で示した実践標準の1つとして策定された。当初、左右にある「ソフトウェアエンジニアリング原理・原則」と「価値ある実践」との間で、どのようにバランスをとるかが最大の問題となった。前者への依存性を強くし過ぎると、実践者から敬遠され、後者への依存性を強くし過ぎると、時代の経過とともにSWEBOKそのものが陳腐化する恐れがある。このような議論の結果、SWEBOKの中に取り入れるべき知識領域と、その内訳となるトピックスを選定するために従うべき厳しい基準が作られ、策定が行われた。

1.4 IEEE及びISO/IEC国際標準との関連

IEEE以外に、主な国際標準策定機関として、ISO⁴、IEC⁵、ITU⁶が存在する。これらの機関では、標準が具

えなければならない性質を次のように定義している [SESC1993]。

関連するすべての利害関係者（たとえば顧客、製作者）からコンセンサスを得ること

利害関係のあるコミュニティに対する開放性 (openness) を維持すること

標準策定プロセスにおける速度と品質のバランスを適切にマネージすること

標準原案に対する技術上の (technical) 合意形成 (agreement) 及びマネジメント上の合意形成を均等に扱い、それぞれを確立するための努力をし、両合意が形成されてから発効させること

SWEBOKは、IEEE及びACM⁷の協力によって作成されたものであるが、上記の4条件を満たした上で、ISO/IEC JTC1 SC7に対して最短経路 (ファースト・トラック) で提案され、同WG20で審議された結果、ISO/IEC TR19759として発効された。

ソフトウェアエンジニアリング分野では、既にIEEE自身で数多くの標準を発効しているが、SWEBOKの知識領域と、これまで発効されているIEEE標準との対照関係は、SWEBOK2004付録Cに示されている。

1.5 IEEE/ACM Computing Curricula 2005

1.2項で、ソフトウェアエンジニアリングが正当なエンジニアリング・ディシプリンであると認知されるために必要とされる基準の1つとして、初期専門職業教育とその教育認定の存在が必要であることを述べた。初期専門職業教育とは、具体的には大学・学部等における教育を意味する。公に確立された教育認定機構として、我が国では先に述べた日本技術者教育認定機構 (JABEE) があ

り、米国にはABET⁸がある。JABEE及びABETにおけるエンジニアリング教育認定についてここでは述べない。これら教育認定審査では、通常、なんらかのカリキュラムモデルが参照される。情報処理関連分野では、ABETはIEEE/ACMが策定したカリキュラムモデルCC2005 (Computing Curricula 2005) [CC2005]を参照する機会が多い (必須要件ではない)。CC2005は、次のサブモデルが

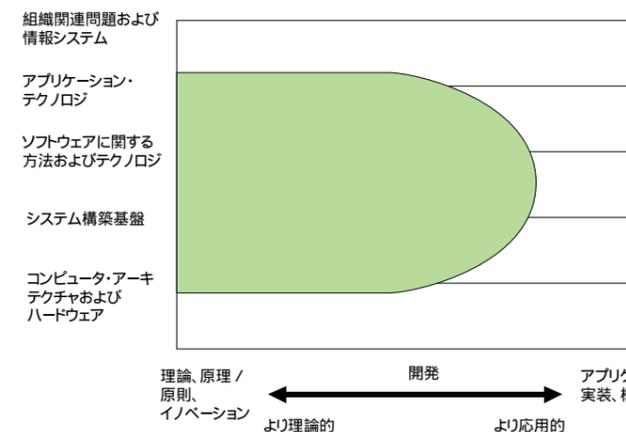


図3(b) カリキュラムモデルCS2001がカバーする範囲 [CC2005]

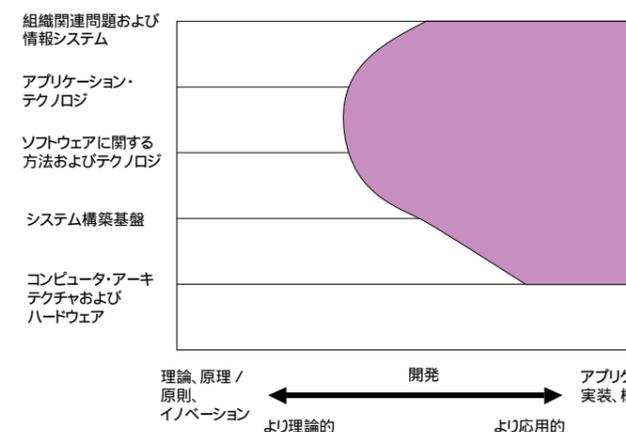


図3(d) カリキュラムモデルIT2006がカバーする範囲 [CC2005]

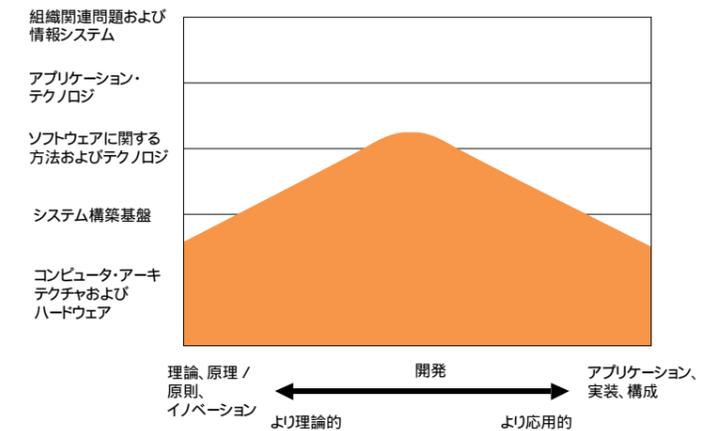


図3(a) カリキュラムモデルCE2004がカバーする範囲 [CC2005]

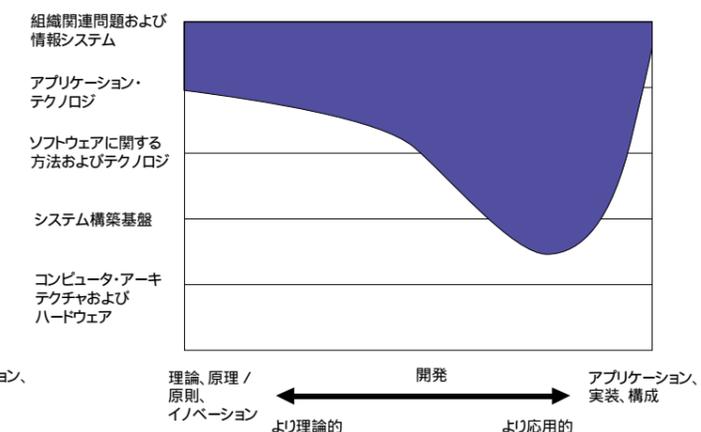


図3(c) カリキュラムモデルIS2002がカバーする範囲 [CC2005]

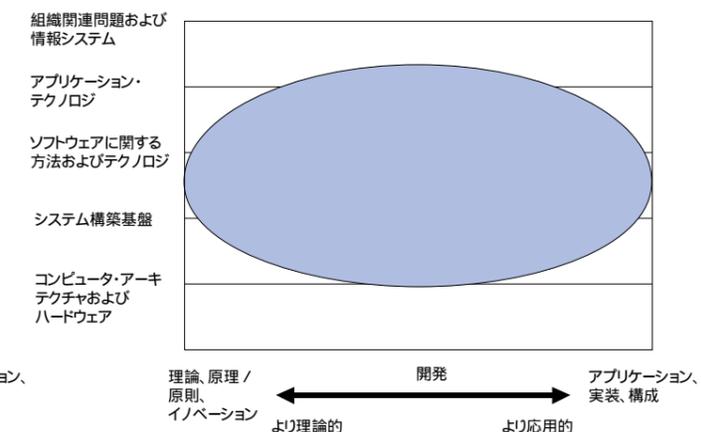


図3(e) カリキュラムモデルSE2004がカバーする範囲 [CC2005]

3 International Association of Association Management Companies (www.iaac.org)
 4 The International Organization for Standardization
 5 The International Electrotechnical Commission
 6 The International Telecommunications Union
 7 The Association for Computing Machinery
 8 The Accreditation Board for Engineering and Technology

ら構成されている。

- Computer Engineering (CE2004)
- Computer Science (CS2001)
- Information Systems (IS2002)
- Information Technology (IT2006)
- Software Engineering (SE2004)

SE2004の策定には、筆者も参加し、情報処理学会で策定したソフトウェアエンジニアリング・カリキュラムモデルJpn1 [JPN1-04]を、国別例としてSE2004の中に織り込んだ。

CC2005の中の各カリキュラム・サブモデル、CE2004、SE2001、IS2002、IT2006及びSE2004がカバーする範囲が、それぞれ図3(a)～(e)によって示されている。この図の縦軸では、下端へ向かっては、コンピュータ・ハードウェアにより近い性質、上端へ向かっては、システムにより近い性質が表示され、横軸では、左端へ向かっては、より理論的な性質、右端へ向かっては、より応用的な性質が表示されている。

ソフトウェアエンジニアリングは、すべてのエンジニアリングに関係し、共通部分を受けもたねばならないという性質がよくわかる。

2 SWEBOKが国際標準になるまでの経緯

2.1 IEEE Computer Societyの活動

ソフトウェアエンジニアリングを、1.1項で述べた3つの要件を満たし、専門職業として社会的認知を得ようとする活動は、1993年にIEEE Computer Societyで始まった。同年、少し遅れて、ACMでも同様の活動が始められ、さらに両者の交流が行われた結果、1994年にIEEE/ACM合同委員会が発足し、(1)事業における意思決定、(2)専門職業人資格認定及び(3)教育認定のための基礎となり、ソフトウェアエンジニアリングを職業として実践するため

の評価基準及び規範になるような標準を策定する作業を開始した。SWEBOKは、この作業によってもたらされた成果物の1つである。

SWEBOKの作成は、SWECC (Software Engineering Coordinating Committee) によって進められたが、エンジニアリングの視点からまとめようとするIEEEと、サイエンスの視点から進めようとするACMの間には、様々な確執が見られ、最終的にIEEEが中心となって素案がまとめられた。2001年にTrial Version、2004年にIronman Version⁹が作成され、公布された。原案は、各章ごとに公募された結果、採用された個人によって書き起こされ、前者の査読には、42か国からおよそ500人、後者に対しては21か国から124人が参加した。

SWEBOKは、図4に示す11の知識領域及び各知識領域に属する複数のトピックスから構成されている。

知識は、「一般的に認められた (generally accepted) 知識」と「先進的、研究的 (advanced and research) 知識」にカテゴリ分けできるという前提に立ち、SWEBOKは、前者、すなわち「一般的に認められた知識」の中でとくに中核を占める知識だけを扱うものであると定義された。また、SWEBOKキックオフ会議で、「一般的に認められた」とは、大学・学部卒業後4年の実務経験を経たソフトウェアエンジニアが合格できるソフトウェアエンジニアリング技術者試験において、準備学習のために作られる図書に盛り込まれるべきレベルの知識であると定義された。このようにSWEBOK記述のための厳格な仕様が定められ、その詳細は、SWEBOK2004の付録Aに記されている。

2.2 ISO/IEC JTC1 SC7の対応

IEEEは、2001年にISO/IEC JTC1 SC7に対して、SWEBOKを国際標準のための提案として提出し、SC7はWG20 (Software Engineering Body of Knowledge) を新たに設けて提案を審議した。我が国の情報規格調査会においても、これに対応する小委員会 (主査：筆者) を2002

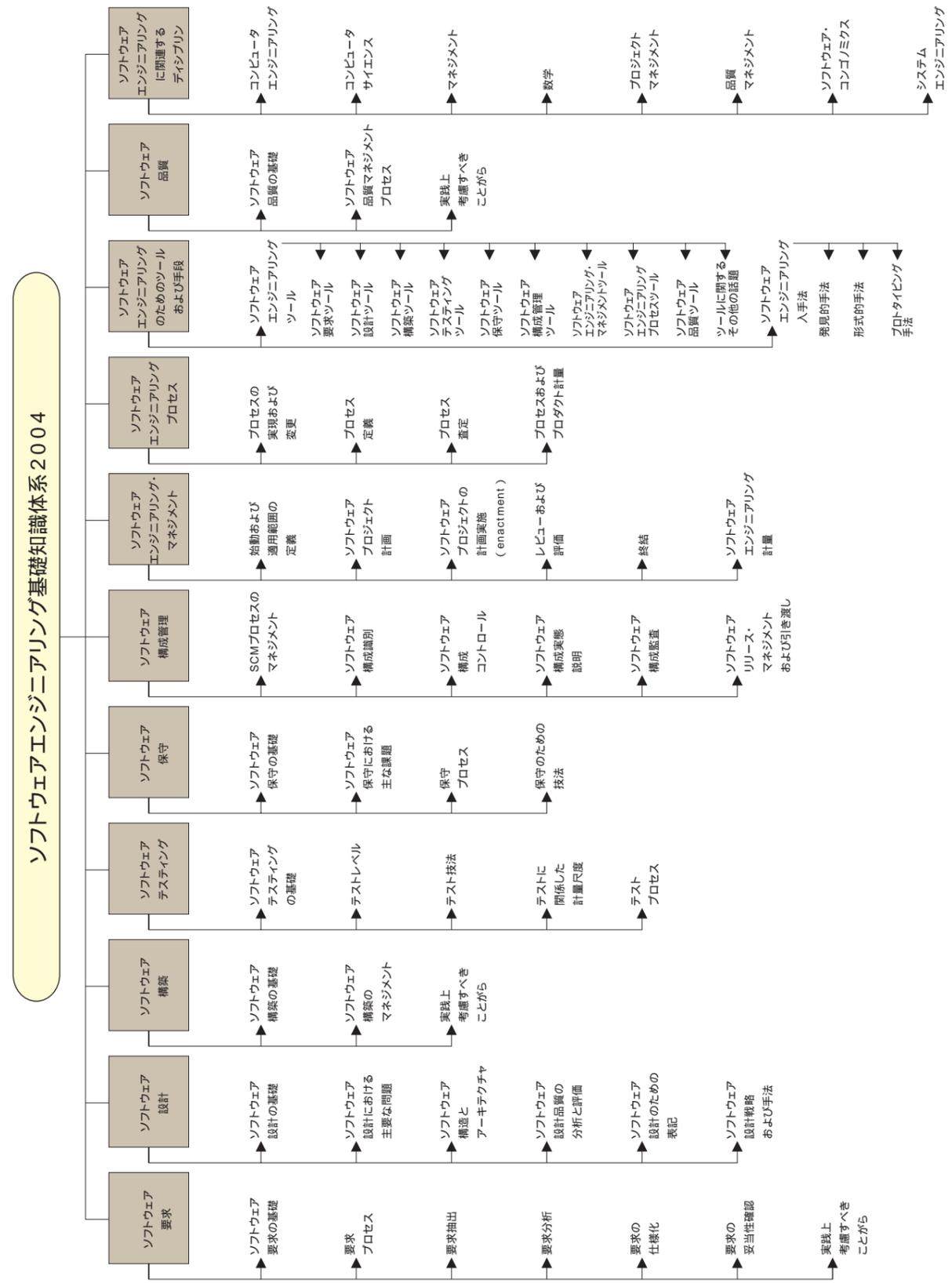


図4 SWEBOK2004の構造 [SWEBOK2004]

9 <http://www.swebok.org/>

年に設置し、審議に参画した。その結果、SWEBOK2004が、ISO/IEC TR 19759:2005 Software Engineering - Guide to the Software Engineering Body of Knowledge (SWEBOK)として登録された。

2.3 これからの計画

SWEBOK2004は、現在、IEEE CSDP (Certified Software Development Professional) 及びIEEE Distance Learningカリキュラムで活用されている。また、ソフトウェアエンジニアリング関係のIEEE標準は、SWEBOKの構造に沿って分類されている。

IEEE Computer Societyは、SWEBOK2004を、4年周期の「タイムボックス」方式で進化させる、すなわち4年ごとに更新しようとしている。周期の最初の1年目に知識領域構造の変更可否を検討し、2年目及び3年目にトピックス内容の変更に対する検討を行い、4年目に文面を変更、参考文献を更新し、発効させようとする。このタイムボックスの中で、ワークショップ開催、草案作り、投票、投票結果の問題解決が、適宜配置される。現在進められているタイムボックスによる更新は、2008年の予定である。

3 ソフトウェアエンジニアリング基礎知識体系

3.1 一般事項

SWEBOKの10の知識領域及び関連するディシプリンは、図4に示したような構造を形成する。各知識領域は、トピックス及びサブトピックスから構成され、各トピックスに対する参考文献及び参照箇所が、それぞれにおいて示されている。

この知識領域構造から、SWEBOKはウォーターフォール・ライフサイクルモデルに従って作られていると誤解

される向きが多いが、他に適当な配置順がないために、このようになっているに過ぎない。SWEBOKにおける知識領域の並び順は、ライフサイクルモデルや方法論に全く関係がなく、アジャイル型から計画駆動型を含むすべてのライフサイクルモデルに適用できるように記述されている。

3.2 知識領域の概要

3.2.1 ソフトウェア要求

この知識領域に関して受ける誤解は、システムやビジネス利害関係者の要求に関する扱いの貧しさである。システム(ビジネスを含む)エンジニアリングに関する知識体系は、INCOSE (International Council on Systems Engineering) のG2SEBoK (oは小文字)(Guide to Systems Engineering Body of Knowledge)において、ビジネスを含む、より広い分野を対象にして、かつ深く論じられている¹⁰(余談になるが、これに対して我が国のトヨタ自動車の生産システムエンジニアリングが強く影響を与えている)。SWEBOKにおける「ソフトウェア要求」は、G2SEBoKをもとにして形成される「システム要求」によって導出されるものであるという原則に立って記述されている。INCOSEは、G2SEBoKを、やがてISO/IEC JTC1 SC7に対して提案する、と聞いている。

3.2.2 ソフトウェア設計

この知識領域では、ソフトウェア設計を、明確に、ソフトウェアアーキテクチャ設計とソフトウェア詳細設計に分けている。この知識領域は、時代の技法推移の影響を最も受けやすいところであるため、記述はごく基本的な内容にとどめられている。

3.2.3 ソフトウェア構築

ソフトウェア構築は、従来、プログラミングと呼ばれ

ていた領域である。プログラミング能力に占める、設計的能力及びテスト能力の割合が、きわめて強くなっているため、「ソフトウェア構築 (software construction)」という名称に改められた。知識領域のトピックスに、構築設計、構築テスト、再利用等が加えられた。

3.2.4 ソフトウェアテスト

テストをライフサイクルの最後に配置する考えは、すでに過去のものとなっているため、テストはソフトウェア品質の中のV&V (Verification and Validation: 検証と妥当性確認)に含めるべきであるという議論が強くなされた。しかし、実務の現場では、ライフサイクルの最終ステージでのテストに重きを置く場合が多いので、独立した知識領域として残された。SE2004カリキュラムモデル策定では、テストを独立させることは認められず、V&Vの中で扱われることになった。

3.2.5 ソフトウェア保守

2000年問題によって、業界は運用後のことを考えながら開発を進めねばならないことの重要性を思い知らされた。また、オープンソース・パラダイムにおける保守のあり方が問題になり、運用後の保守に費やす費用の80%が是正 (corrective maintenance) のためではなく、それ以外(たとえば機能や性能向上)のために使われていることなどの事実がある。このため、従来の「保守」という概念を全く改めた記述、すなわち、リリース以前に行うべき保守に関わる問題を強調した記述となっている。

3.2.6 ソフトウェア構成管理

時代にあまり左右されない知識領域であるため、古典

的構成管理論が、手際よくまとめられている。

3.2.7 ソフトウェアエンジニアリング・マネジメント

SWEBOK-Trial版 [SWEBOK-Trial]では、この知識領域を、組織マネジメント (organizational management) とプロジェクトマネジメントに分けていたが、SWEBOK2004へ向けて改訂を行う際、委員の1人であるR.H. Thayerの主張が通って、組織マネジメントが削除された。削除の主な理由は、組織マネジメントは、ソフトウェアエンジニアリングだけの問題でなく、関連する領域が広いためであるということであった。しかし、組織マネジメントを削除したことには、問題が残る。

3.2.8 ソフトウェアエンジニアリングプロセス

この領域は、時代の動向を最も反映する知識領域であるため、できるだけ基本的な知識だけに絞るよう努力がされた結果、CMM (Capability Maturity Model)¹¹、SEPG (Software Engineering Process Group)¹²及びExperiment Factory¹³だけが、トピックスに加えられた。

3.2.9 ソフトウェアエンジニアリングのためのツール及び手法

具体的なツール及び技法に触れることなく、それぞれを分類するために利用できる標準的な項目だけが示されている。

3.2.10 ソフトウェア品質

IEEE Computer Society及びACMが定めているソフトウェアエンジニアリング倫理綱領によれば¹⁴¹⁵、ソフトウェアエンジニアには、プロセス及びプロダクト双方において、最高の品質、効果及び品位を獲得することが(倫理

10 <http://www.incose.org/practice/guidetosebodyofknow.aspx>

11 <http://www.sei.cmu.edu/cmml/>

12 <http://www.sei.cmu.edu/sep/>

13 <http://www.cs.umd.edu/users/basil/>

14 IEEEの場合は、<http://www.computer.org/certification/ethics.htm>

15 ACMの場合は、<http://www.acm.org/serving/se/code.htm>

綱領では、3.01、3.09、5.01、5.05、8.02の各項において) 義務付けられている。この知識領域は、この倫理綱領を 実践するために必要な知識をまとめている。品質保証と V&Vに焦点を当て、それぞれの中での重要なトピックス を展開している。

3.2.11 ソフトウェアエンジニアリングに関連する ディシプリン

関連するディシプリンとして、コンピュータエンジニアリング、コンピュータサイエンス、マネジメント、数学、プロジェクトマネジメント、品質マネジメント、ソフトウェア・エルゴノミクス、システムエンジニアリングを取り上げ、それぞれに関連する項目を羅列している。

CCSE (Computing Curricula Software Engineering) Steering Committeeには、筆者も委員として参加した。ここでは、まず、SWEBOKが大学等の学部教育に活用できるかどうかの議論がされた。議論の内容は、数年にわたる Conference on Software Engineering Education and Training (CSEE&T) の予稿集に採録されている。SWEBOKは、そのままでは活用できないという結論になり、独自のソフトウェアエンジニアリング教育基礎知識体系 (SEEK: Software Engineering Education Knowledge) が計画された。SEEKは、公開に至らなかったが、その内容に沿ってSE2004が作成された。作成の過程では、コンピュータサイエンスの視点を重視するACMメンバとエンジニアリングの視点を重視するIEEEメンバとの間で、緊迫した議論が展開された。

4.2 ソフトウェアエンジニア資格試験への活用

2.3項で述べたように、SWEBOKの10知識領域に対して、Business Practices and Engineering Economicsを加えた11の知識領域に関して、ソフトウェアエンジニア資格認

定のための試験CSDPが実施されている。図5は、ソフトウェアエンジニアの能力・知力を評価するための基礎となるモデルの一例を示している。最下段には、関係する知識体系が配置され、その上に知恵と技能が生まれる。さらにその上に知的スキル及びモータ・スキル (反射的に意思決定し、行動する能力) が醸成され、最上段には、個人の本質から自然に生まれる資質・人格が形成されるものと考えられる。

CSDPでは、下から3段目にあるインテリジェント・スキル及びモータ・スキル以下をカバーする能力を評価する (モータ・スキルの評価を行うため、限られた時間にすばやく意思決定をしなければならないような試験を行っている)

参考文献

[CC2005] http://www.computer.org/portal/cms_docs_ieeeecs/ieeecs/education/cc2001/CC2005_FinalFeb06.pdf
 [CHARETTE2005] Charette, R.N. : Why Software Fails, IEEE Spectrum, September 2005, pp.36-43
 [HARRISON2004] Harrison, W. : Best Practices: Who Says?, pp.8-10, IEEE Software, January/February, 2004
 [ISO/IEC TR19759] ISO/IEC TR19759-2004 : Guide to the Software Engineering Body of Knowledge, 2004 Version, 2004
 [JPN1-04] <http://blues.se.uec.ac.jp/acc-se/IPSJ-SE-Curriculum.html>
 [SESC1993] SESC Long Range Planning Group : Master Plan for Software Engineering Standards, Version 1.0, Dec. 1, 1993
 [STARR1982] Star. P. : The Social Transformation of American Medicine, Basic Books, 1982
 [SWEBOK-Trial] IEEE Computer Society : Guide to the Software Engineering Body of Knowledge Trial Version, IEEE Computer Society, 2001 (邦訳: ソフトウェアエンジニアリング基礎知識体系SWEBOK, オーム社, 2003)
 [SWEBOK2004] IEEE Computer Society : Guide to the Software Engineering Body of Knowledge - 2004 Version, IEEE Computer Society, 2004 (邦訳: ソフトウェアエンジニアリング基礎知識体系SWEBOK2004, オーム社, 2005)

4 SWEBOKの活用

4.1 大学等における基礎教育への活用

1.5項で述べたSE2004を策定するために組織された

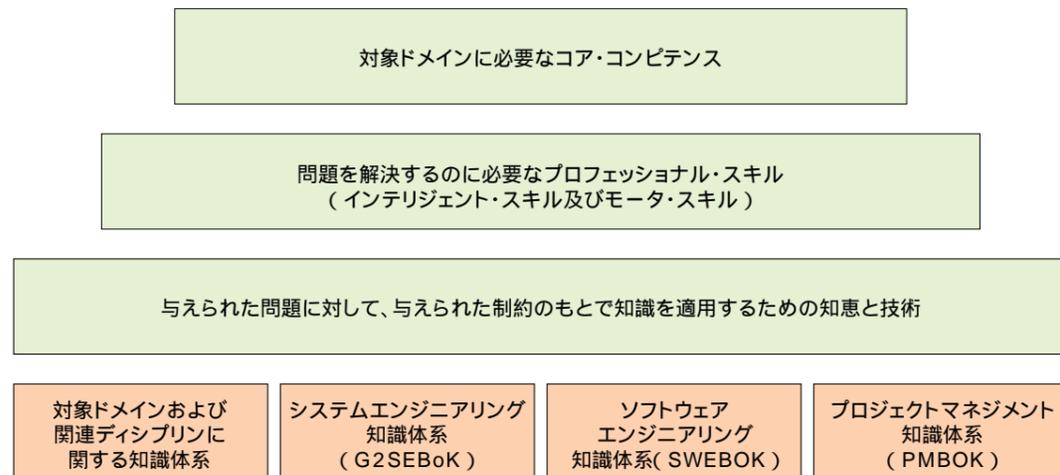


図5 エンジニアとしての資質・スキル・知識を評価するためのモデル例

5 むすび

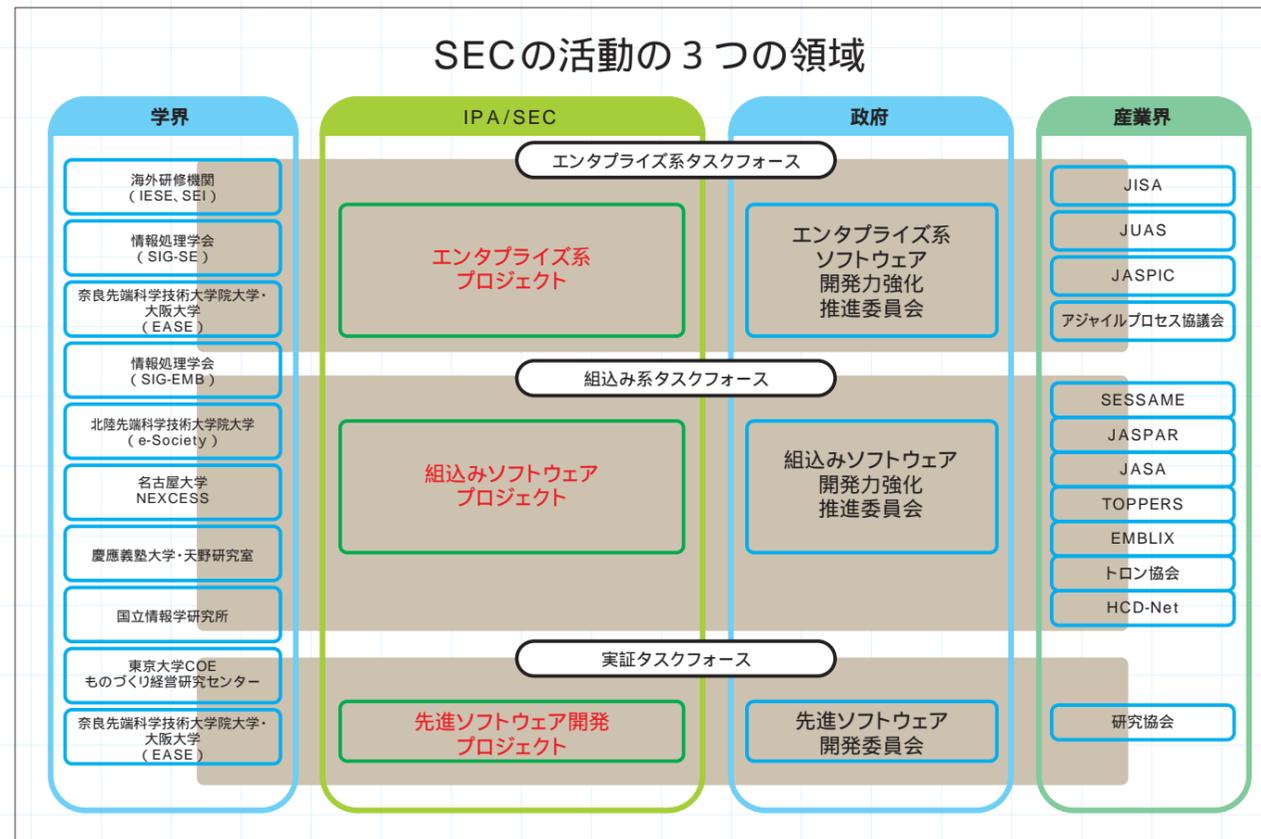
ソフトウェアに対する要求は、質的、量的にとどまるところなく変化・拡大し、不成功に終わるプロジェクトが増加し (例えば、[CHARETTE2005])、市場は脆弱なソフトウェアの介入を許し、ソフトウェア企業の倉庫は、利用 / 再利用されないソフトウェアで満ち溢れ、運用中のシステムは保守・改善困難なソフトウェアに悩まされている。これら市場原理主義の悪の面を反映して生起する人的、経済的資源の浪費を正すための手段の1つは、倫理綱領に根ざした専門職業人 (プロフェッショナル) としてのソフトウェアエンジニアの自覚と良識であろう。SWEBOKは、まだ、策定・試行されてから5年ほどしか経たないため、この視点からは内容的に中途半端な印象を与えている。4年ごとに改訂する機会が与えられているので、SWEBOK2008 (未定) には、倫理、ビジネス法規、エンジニアリング・エコノミクス等を盛り込むよう提案しているところである。

特集

SEC2005年度活動概要

2004年に発足したSECの活動領域には、「エンタプライズ系タスクフォース」、「組込み系ソフトウェアタスクフォース」、「実証タスクフォース」の3領域があります。現在、我が国では、エンタプライズ系及び組込み系ソフトウェアに散発的に発生する不具合によって、残念ながら経済的損失や人的被害が発生してしまっています。SECでは不具合を防止し、高品質・高生産性のソフトウェアを開発するために、ソフトウェアエンジニアリングを日本のソフトウェア産業全体に浸透・普及させることが重要と考えています。エンタプライズ系では定量化と上流工程からのアプローチ及びプロジェクト見える化手法。組込み系ではエンジニアリング領域と組込みスキル領域のアプローチ。双方の成果に基づき、普及活動しています。実証タスクフォースでは、関係団体による成果も取り入れ、実プロジェクトで適用し、その成果を実証しています。ここでは、2005年の活動概要を、共同研究も含めまとめます。

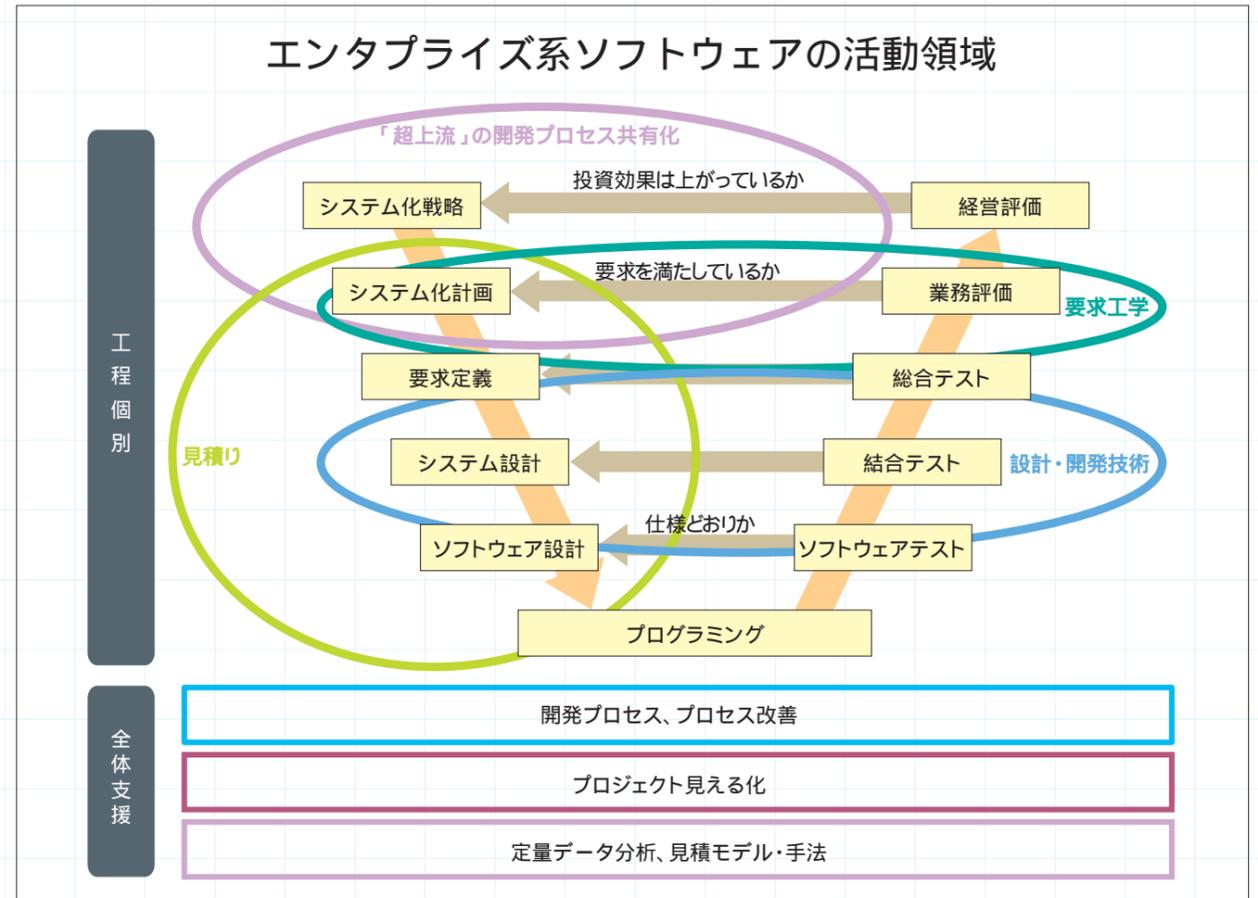
SECの活動の3つの領域



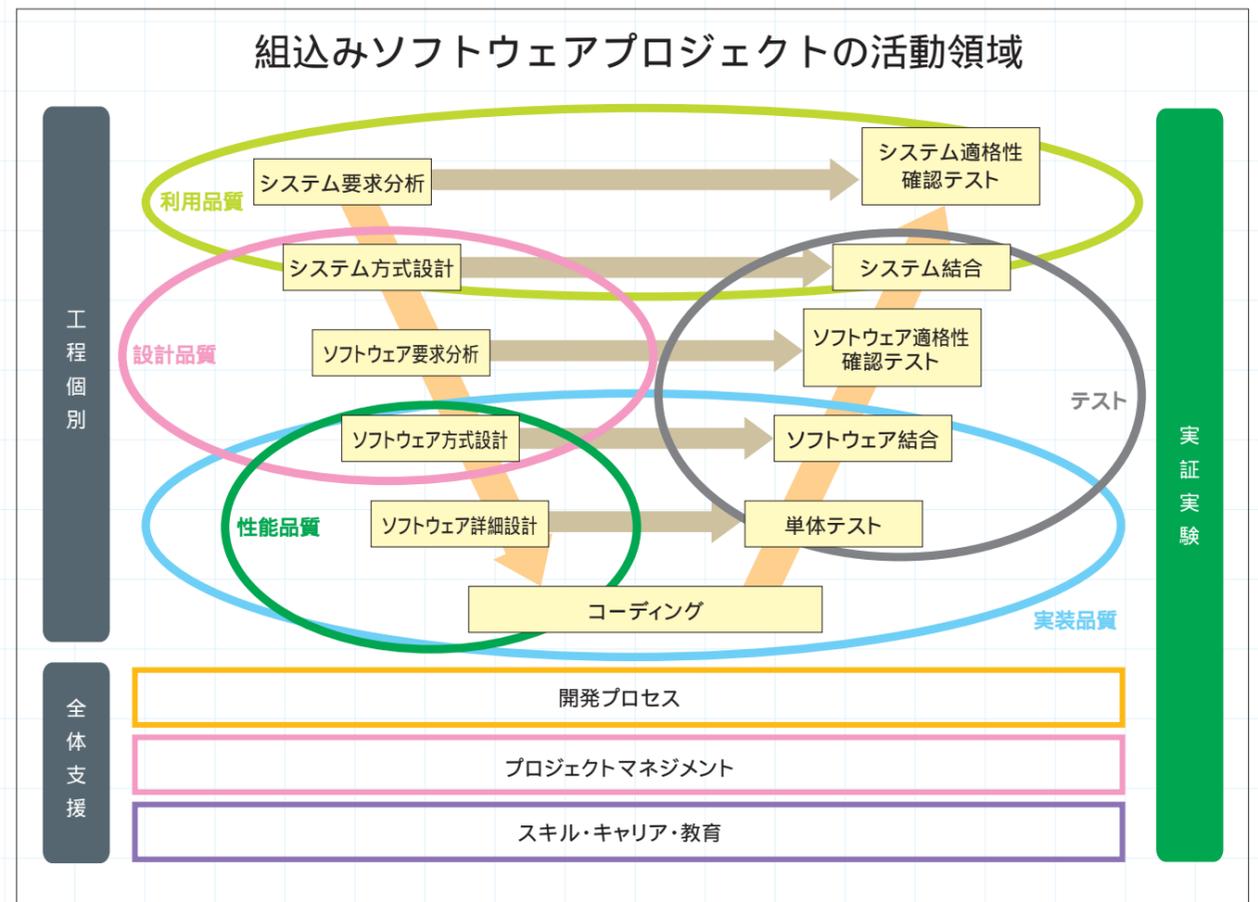
団体略称一覧

- JISA：社団法人 情報サービス産業協会
- JUAS：社団法人 日本情報システム・ユーザー協会
- JASPIC：日本SPIコンソーシアム
- SESSAME：組込みソフトウェア管理者・技術者育成研究会
- JASPAR：Japan Automotive Software Platform and Architecture
- JASA：社団法人 組込みシステム技術協会
- TOPPERS：TOPPERSプロジェクト
- EMBLIX：NPO法人 日本エンベデッド・リナックス・コンソーシアム
- HCD-Net：NPO法人 人間中心設計推進機構
- IESE：フラウンホーファー協会IESE (Institute for Experimental Software Engineering：実験的ソフトウェア工学研究所)
- SEI：Software Engineering Institute
- SIG-SE：社団法人 情報処理学会ソフトウェア工学研究会
- SIG-EMB：社団法人 情報処理学会組込みシステム研究グループ

エンタプライズ系ソフトウェアの活動領域



組込みソフトウェアプロジェクトの活動領域



エンタプライズ系活動

SEC 研究員
石谷 靖

社会インフラやビジネスアプリケーション等、顧客が存在し要望に応じて構築するITシステムに関して、システム障害等の負の側面のみならず新ビジネス躍進等の正の側面の双方から、ますますその重要性が注目を集めている。2004年10月の設立以来、開発に焦点をあて、品質、生産性向上の観点からエンジニアリングの定着を目指し、安心・安全な社会と産業競争力の強化に対する世の中の期待に応えるべく活動をしている。

1 背景と目的

昨年、東証システムをはじめ社会インフラでのシステム障害が多数発生し、社会経済への影響の大きさを見せつけた。エンタプライズ系ソフトウェアでの開発力向上に一層目が向けられており、そのゴール達成が急務である。ゴールは、

- ・ユーザ企業はビジネスでのITシステムの重要性を認識して開発にコミットすること。
- ・ベンダ企業はソフトウェア開発の高生産性と高品質を実現すること。
- ・高品質のシステムが必要なタイミングで入手・稼動する状況が当たり前となること。
- ・信頼性が高く安心できる社会インフラと企業競争力向上を実現すること。

である。

エンタプライズ系ソフトウェア開発はさらにサービスの業的となり、より上流での付加価値を求める傾向が強くなっている。したがって、ビジネスと融合して価値をもたらすために、ユーザとベンダの役割分担と適切な要求

表1 エンタプライズ系のテーマ

分類	個別テーマ(部会)	発 足
要求から 要件・設計・構築へ	超上流における役割分担	2004年10月
	要求工学・設計開発技術	2004年10月
定量的アプローチ	定量データ分析	2004年10月
	プロジェクト見える化	2005年10月
	見積り手法	2004年10月
プロセス改善	プロセス改善	2006年 1月

の定義・実現、環境や規模の変化にも適応可能な最適なアーキテクチャの実現が大きな課題となっている。

2 SECでの取り組み

ゴール達成のためには、ユーザを含むソフトウェア開発の現場へのソフトウェアエンジニアリングの定着が必要であるとの認識の下、SECでは、ソフトウェア開発の基本的な考え方を始め、定着のためのノウハウの収集・分析、ソフトウェア開発手法を研究し、普及を図っている。具体的には、エンタプライズ系の活動として、開発の上流を中心としたテーマに取り組んでいる(表1)。テーマは、大きく3つに分類される。

要求から要件・設計・構築へ

要求を明確にする初期段階の問題から、実際にシステムを作るまでを対象とするものである。ビジネスを成功させるシステムの実現には、問題の本質が事業や業務検討の始まりに一層シフトしていることを再認識し、システム実現のためには誰が何を決定すべきかを示し、システム開発でのユーザ・ベンダ双方の利害関係者の役割と責任の重要性とその実践を改めて世に問いかけている。そして、要求工学・設計開発技術では、実際に要求とアーキテクチャをどのようにつなげシステムとして構築するのか、技術的な観点を中心に取り組んでいる。

定量的アプローチ

開発を定量的に把握し確実にマネジメントするものであり、「見える化」というキーワードでまとめることができる。SECでは「プロジェクト見える化」と「定量的データ分析」の2つの活動を行い、定性的・定量的な両側面からプロジェクトをいかに見えるようにし、プロジェクトを成功につなげるかに取り組んでいる。いずれの活動でも事例、プロジェクト定量データや実際の経験の分析を中心としている。見積り手法は分類と密接なテーマであり、説明力のある見積りモデルの構築を勧め、要求定義の重要性と定量データ分析の重要性の両者を訴えている。

プロセス改善

組織としての能力を高めるための組織全体のプロセス改善である。分類や分類の具体的な技術・手法に基づいて、技術、人、プロセスをいかに効率よく設定し、継続的な改善を成功に結びつけるかを対象とする。本号17頁に、エンタプライズ系活動領域のテーマがカバーしている範囲をいわゆるシステム開発のV字モデルに示した。上流を中心にする過程と作り方のマネジメント及び組織的な取り組みの実現と継続的な改善をカバーして、ゴールを実現するための材料を提供している。

3 活動サイクルと現在の段階

SECの活動サイクルは、「調査・研究」「標準化・開発」「普及・広報」「評価・実態調査」である。各テーマも同様なサイクルを進めており、図1にその詳細を示す。テーマごとに進展の状況に違いはあるが、SECは3月で2年度目を終え、啓蒙・啓発は初年度(2004年度)で対応し、小冊子や定量データ白書が具体的な活動の成果である。次のステップのノウハウ等の提供は昨年度(2005年度)の活動と成果が対応する。

また、実証実験は海外の研究機関や大学との共同研究

を通して行っている。表1に示すとおり「プロジェクト見える化」と「プロセス改善」は発足したばかりだが、2006年夏を目標に最初のステップの成果(考え方の基本、心得等)を発表する予定である。個別のテーマの具体的な内容については、それぞれの稿でご確認いただきたい。

4 今後の方向

2006年度の基本的な展開は、特に新たな重要なテーマである「見える化」手法と効果のあるプロセス改善を探るとともに、実証・普及のフェーズに進む。「超上流における役割分担」「定量データ分析」「見積り手法」については、開発現場への普及を促進する活動を行う。さらに、テーマを統合してトータルな課題解決策を産業に提供する一環として、経済産業省の「信頼性向上ガイドライン」の実践を促進するために、これまでの成果を「信頼性確保」の観点からまとめ上げる活動を開始する。その過程で、共通の言葉として復刊が望まれている「共通フレーム98」の改訂版(「共通フレーム2006」)を策定する予定である。また、具体的な応用先の1つとして、「政府調達改善への支援」をターゲットとして考えている。

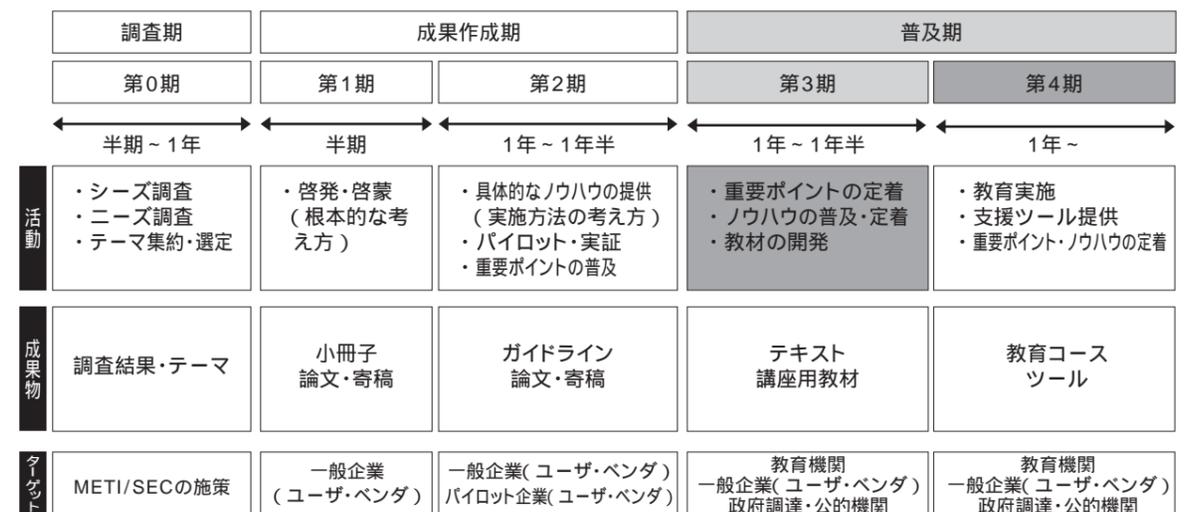


図1 テーマのサイクル

見積り手法

SEC 研究員
石谷 靖

プロジェクトの規模、工数・コスト、工期の見積りの妥当性確保は、ユーザ企業とベンダ企業の双方にとって、必要なシステムを妥当な価格で必要な時期に入手したり、一定の利益を確保する点で、プロジェクトの成否に直接つながる。低すぎる見積りでは、ベンダ企業に必要なアウトプットの達成が困難になり、高すぎる見積りはユーザとベンダとの間の信頼を損なう原因になる。

また、見積りは契約時においてのみ重要なのではなく、見積り結果は、プロジェクトの実施内容や方法まで規定する最も基本的なデータである。プロジェクトの予測として作成した見積り値は、プロジェクトが開始すると目標（ゴール）となり、見積りに応じたプロジェクト成果をいかに達成していくかという課題になる。

さらに、見積り活動は1つのプロジェクトに閉じたものではなく、予測と実績との差異分析を通して次のプロジェクト、さらには組織全体のプロジェクトの見積り精度向上に向けてのフィードバックの基となる。見積り精度の向上は単に見積り技法の問題ではなく、組織としてどのようにプロセスを構築し、取り組んでいるかという点につながる。

1 SECでのこれまでの取り組み

見積り手法は、過去の定量データを活用し、豊富な経験も活用できることから、エンジニアリング的なアプローチを適用しやすい。

このように見積り活動はプロジェクトや組織全体にかかわる重要なものであるが、その場限りの活動の典型である「勘と度胸」に基づいた見積りがなされ、実績が予測から大きく外れることも少なくない。

この現実とのギャップが見積り手法における大きな課題である。

その原因としては、次のことがあると考えている。

- ・ソフトウェア開発の見積りに関して、すでに多くの結果が生み出されているにもかかわらず、ユーザでもベンダでも自社の条件にあった、よりよい見積り手法を

求めている。

- ・予算確保等のため、根拠となる要求等が明確になっていない段階での見積りの必要性も高く、実績と見積りとのずれを大きくする原因となっている。

- ・最初の見積り結果が後に確定していく仕様と合わなくなる場合も、最初の結果をそのままプロジェクト目標としてしまうことも多い。

結果として、関係者間、とくにユーザとベンダの間で見積りに関して、互いに納得が得られていないという状況を生んでいる。

以上の問題意識に基づいて、規模、工数・コスト、工期を対象に、精度の高い（または、妥当な）見積りとは何か、その実現方法は何かを、SECでは追求してきた。1年目は、2005年春に啓発・啓蒙の『ITユーザとベンダのための定量的見積りの勧め』を発行した。2年目の2006年春には、具体的な手法に関する『ソフトウェア開発見積りガイド』を発行し、企業での取り組みを中心に10の見積り手法の事例を紹介する。

精度の高い見積り手法を実現するための重要なポイントは、次の3点である。

自分の組織のデータに基づいた見積りモデルの構築

自組織で過去の実績データを収集・分析することを基本にして、ベースライン及びベースラインからの変動要因の設定により各種見積りに関するモデルを構築する（図1）。

見積りプロセスの確立及び予測と実績の差異分析を通じた継続的な改善

見積り手法の確立は重要であるが、見積り値と実績値の差異分析等を通して、見積り手法の改善やプロジェクトマネジメントの改善につなげ、見積り精度の向上を図ることが併せて重要である（図2）。

要求定義の重要性と早期見積りにおけるリスクの存在と契約によるリスク低減

プロジェクトの早期には、見積りを行いたいニーズと要求は部分的にしか決まっていないというジレンマがあ

る。その解決策としては、見積りの前提（どの情報を使っての見積りか（図3））を明確にし、情報が明らかになる時期をモニタ・コントロールして再見積りを行う等、契約（例：多段階契約）で解決することが肝要である。

2 今後の方向

見積り手法の取組みは、これまでの、SECでの成果を広くソフトウェア産業に伝え、導入・定着を進める段階となる。今後、検討すべきテーマとして保守開発の見積りのテーマがある。それぞれについて、次のような展開を考えている。

導入・定着

普及としては、SECで検討した見積り手法を企業・組織で活用していただくことを考えており、政府調達で

の早い段階での簡便見積り手法の活用が1つの目標である。

実証としては、引き続きCoBRA法等の見積り手法について協力企業で実証と効果の確認等を行う。

保守開発の見積り手法

保守開発、すなわち、開発母体がある場合の開発における見積り手法について、既存手法適用事例を収集し、考え方及び方法についてガイドブック等を整備する。保守開発に対する検討は、見積り手法のみならず企業のニーズが高いことが背景である。

2006年度以降の展開は、これまでの成果の普及を図ることが基本となり、テーマとしての広がりや、他のエンタプライズ系のテーマとの融合を中心に展開していくことになると考えている。

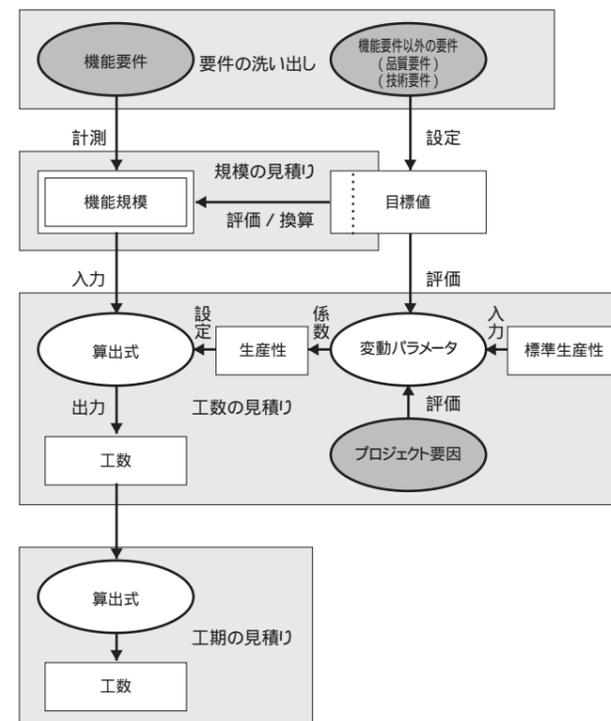


図1 自社のデータに基づいた見積りモデル構築

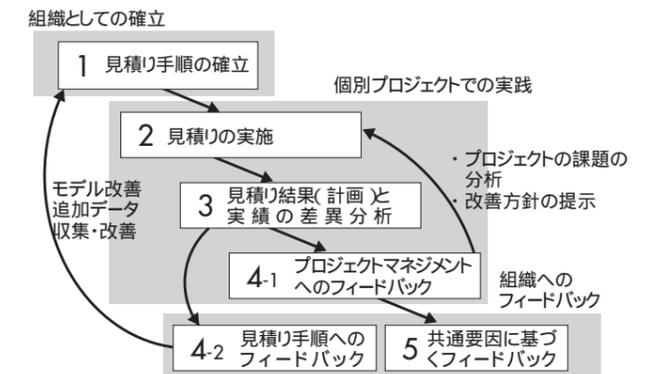


図2 見積り精度の向上の活動

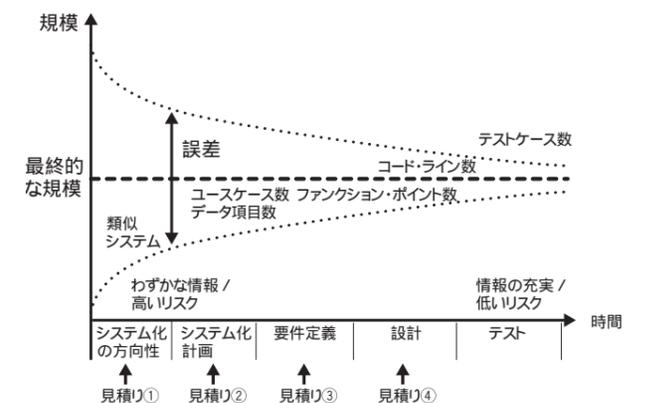


図3 時期ごとに使用できる情報



定量データの定義・収集・分析

-ソフトウェア開発データ白書2006 & データ定義 Ver.2.0のリリース-

SEC 研究員

菊地 奈穂美

ソフトウェア・エンジニアリングの実践には、定量データの収集・分析・応用が基本である。ソフトウェア開発の現場においても、定量データに基づく科学的マネジメントへの進化を目指して、SECでは定量データ分析に2年前から取り組んでいる。第一歩として、プロジェクトの全体像を把握するための定量化を中心に置いた。成果として、発刊2年目になる「ソフトウェア開発データ白書2006」(以下「白書2006」)及びデータ定義書を発行する。

1 定量化に関する期待と課題

国内のエンタプライズ系ソフトウェア領域では次のような課題が認識されていた。

- ・ソフトウェア開発組織では、改善を考えるにあたって、自社や自プロジェクトのポジションを把握し、外部にリファレンスできるような情報を求めている。
- ・ソフトウェア開発にかかわる関係者間で共有認識がしやすく、試算や評価に使える『モノサシ』がない。
- ・プロジェクトの主要な要素の規模・開発期間・工数や品質、及びそれらの間のトレードオフなど、現実には有用な参考情報が国内事例では少ない。

2 方針

このような課題に対し、主として二者間取引に関するITシステムのソフトウェア開発プロジェクトを対象とし、次の方針を掲げた。

- ・プロジェクトの主要な要素である、ソフトウェア規模・開発期間・工数・品質に着目する。
- ・プロジェクト・データを収集し、分析結果を企業間でもリファレンスとして役立つような、実際のデータの分布状況等の情報として提供する。
- ・基盤となる定量データ項目と定義を策定する。
- ・中期的には、データ定義、収集データ精査、分析、結果の活用を含むフレームワークの確立を目指す。

3 成果物の特徴

「白書2006」

プロジェクト実行にあたって必要なことは、関係者がプロジェクトについて状態を定量的に把握・理解できることであり、適切に判断、予測や制御をできることが望ましい。そのため、「白書2006」では、ソフトウェア開発プロジェクトを成功させるために、参考となる基本的な情報を示すことに主眼を置いた。中核となるメトリクスである、ソフトウェア規模(機能規模や行数等) 開発期間、工数、信頼性(稼働後不具合数、不具合密度)等について実績データの分布状況を内容としている。さらに、要素間(規模と工数など)の関係を探った結果を掲載する。規模・工期・工数は、2006年度版からは実績値に加え、計画値についても収集し、計画と実績の予実差も示す。なお、本書では、国内19社からの合計1,400件のプロジェクト・データを収集し、その分析結果を掲載する。

データ項目と定義 Ver.2.0 及び収集フォーム

データ項目と定義は、白書に付録として収録する。定義の電子ファイルはSEC Webサイトからも提供する。収集項目フォームの電子ファイルの提供方法は検討中である。これらにより、各社において「白書2006」と同じ基準でデータを収集し分析に活用できる。

4 「白書2006」のポイント

収集データについて

収録したデータは、国内ベンダ企業19社から提供された1,419件のプロジェクト・データである。

データ収集の重点方針

約80のデータ項目を重点的項目とし、「必須」、「重要」の優先度を付け、それらのデータが収集できているプロジェクトを中心に収集した。本方針は部会で議論し、エンタプライズ系でデータ件数を増やして共有したい項目として選定した。

【本年の重点収集項目】

- ・アーキテクチャ：イントラネット/インターネット、2CS、3CS
- ・業種：金融業、情報通信業、製造業、など
- ・開発言語：Java、VB、C、COBOL
- ・プラットフォーム：Windows系、Unix系
- ・規模：FP、SLOCのいずれかは持つもの
- ・稼働後の不具合数の記録されているもの
- ・工数、規模、工期のデータが把握できているもの
- ・プロジェクトの工数に含まれる作業の種類

【主に収集対象としたプロジェクトのタイプ】

- ・基本設計～総合テストの5工程を実施に含む
- ・時期は、2002年4月～2005年夏頃までに完了
- ・重点収集項目で欠損が極力少ない

データ定義 Ver.2.0のポイント

2006年度版の主な目的は、計画時に参考になる情報提供とした。その際、プロジェクト成否の情報をQCDの観点で分析できることや予実差の分析を目的として、次の新項目をデータ定義に追加した。

- ・QCDそれぞれについて、計画の評価、実績の評価
- ・規模、工数、工期の計画時点の値
- 「白書2006」の分析で改善した点
- ・規模データの精度向上のため、FPは、計測手法別(IFPUGと同一ソフトウェアモデル計測手法のプロジェクト群)で層別を試みた。コード行数の規模は開発言語名を必須で収集し言語で層別実施した。その結果、分析の精度は向上した。
- ・分析のシナリオを拡充し、クロス分析、系列別表示などを工夫して、結果が理解しやすいような表示としている。
- ・2005年度に提出されたデータが多かった未記入(データ無し)項目の対策として、重点項目を設定し、既提出データの見直しと追記も含めたデータ収集を実施した。これにより、未記入が減少し、分析の土台となる有効データが増加、質も向上した。

5 データ分析前の収集・精査のポイント

収集データは、データ提供から、各社内で開発終了後のプロジェクトのデータを一次精査し、確認済みである

ことを前提として、SECへ収集フォームで提供された。データ精度確保について、各社へ確認したところ、例えば、FP規模計測では、各社でFP計測方法の社内教育や計測後データのレビューを行っていることがわかった。データ収集後、SECがデータ精査・確認を行い、データの精度向上に注力した。異常値や誤記と思われるデータについては提供企業に再確認を実施している。この作業を何度も繰り返し、精度を上げることができた。

6 予測や分析に関する先行研究

通常のSEC部会では、部会の参加企業のメンバーなどを中心に、データ項目定義・重点収集方針・仮説・分析結果の解釈レビューの検討等を主に行っている。また、白書掲載のような統計情報を企業の現場で活用する方法を検討している。一方で、ソフトウェア・プロジェクト・データの特徴を踏まえた分析手法や、プロジェクトの予測技術などについては、専門性を持つ大学などの機関と表1のように共同研究の形で先行技術の調査及び検討を進めている。

7 今後の方向

白書の基本形は見えてきたが、さらにデータ間のトレードオフ関係や意義のある仮説等は検討していく。また、企業内での活用の指針も検討事項となろう。技術課題として、企業横断的データを扱うには、各社での計測方法の差によるデータ誤差の扱いの検討があげられる。データの定義・収集・精査・分析のフレームワークのポイントについては、本テーマ担当のSEC研究員が実経験を通しての研究成果をまとめ、知見を発表していく予定である。

さらに、定量的アプローチの進化として、プロジェクトの予測、見積り、実行時の制御での定量データの効果的な活用という目標に向けて進めていく。そのためには継続的な定量データ計測は必須の基盤となるため、各企業において、それぞれの着実な取り組みを願う。

表1 部会検討テーマの先行研究

奈良先端科学技術大学院大学	欠損を含むデータに対する効果的な分析手法の適用と実用性評価
大阪大学	プロジェクト初期段階での、先の混乱予測手法に関する研究



要求工学、設計・開発技術

SEC 研究員

田 尚志

ソフトウェア開発において、要求定義やアーキテクチャ設計といった初期の開発工程がとくに長年にわたって注目され続けている。そして、これらの分野では、要求の獲得、記述、実現を「どのように効率よく高品質に行うか」という研究開発が多く行われてきた。技術的な進歩という意味でも、様々なダイアグラム記法、手法、ツール等が開発され、部分的には、作業の自動化、成果物の再利用等で成果も出てきている。しかしながら、ソフトウェア・システムが社会全体や企業のインフラとなるにつれ、利害関係者が増え、システムを取り巻く環境としてのビジネス、技術、法制度の変化がソフトウェア開発に大きく影響を与えるようになってきており、いまだ課題は多く残されている。

SECでは、要求工学や設計・開発技術の研究部会とともに、これらの分野における技術の実状を調べ、現状課題を把握しながら解決策の検討をすすめてきた。特に産業実態、実践事例、先行技術の3つの視点から理解を深め、先進技術を実践の場と結びつけて生かせるよう技術開発及び普及展開することを目指している。

1 経緯

日本の産業競争力を高められる技術は何か。戦略的に技術開発に取り組めるよう、まずこの視点から、現場で将来利用される技術の先行指標と考えられる研究動向を調べるため、ソフトウェア・アーキテクチャの研究動向調査を実施した。次に、産業で実現でき、望ましい解となる技術はどのようなものかという理解を得るために実践事例をヒアリングする調査を行った。最後に、より非形式的な分野であり、工学的な取組みが他の分野に比べると進んでいない要求工学の分野での研究動向を調べて、将来実践されるであろう技術がどういったものか調査し、整理した。

[第一の調査]

システム品質を支配するともいえる初期の設計判断を

示すソフトウェア・アーキテクチャに関して、主要国際会議の文献（約3,400件）を対象として重要論文を51件抽出するとともに、結果や仮説を整理し、この分野の主だった米国研究者11名へのインタビューを行い、技術の成熟度、注目度、日本のソフトウェア産業への適合性から研究動向を評価した。調査結果では、主要トピックやアーキテクトの役割と開発上の成果物のモデルをベースに次のように整理した。

- ・主要な研究トピックの中でも成熟レベルが比較的高く、実証や実用段階に近いものはソフトウェア・プロダクトラインで、長い期間にわたって多くの論文が抽出されており注目度も高い。
- ・非機能要求としては、個々の品質特性（例えば信頼性のみ）を部分的に対象とするアーキテクチャ設計上の解や設計手法が主であった。非機能要求を全体的に（例えば性能を高めつつ、信頼性や変更容易性を検討する）扱うような例は、ほとんどない。

また、実践では重要だが研究成果を待つだけでは出てきにくい技術がいくつか存在することを把握した。例えば、以下のような技術が該当する。

- ・アーキテクチャ設計を難しいものとしているトレードオフやリスクを理解判断するための技術。
- ・それらを利害関係者間で正確に意思疎通するための文書化やコミュニケーションの技術。

[第二の調査]

第一の調査で理解を深めた先行技術に対して「産業で望まれる解は何か。企業で実践できるアーキテクチャ構築や利用とはどのようなものか。」という視点で9社から実践事例を挙げてもらった。個別のヒアリング調査の結果、うまく実践できている例では戦略的に資産とすることができ設計や人材を配置し、個々のプロジェクトへ展開しており、以下のような特徴があることが確認された。

- ・多くの例では、なんらかの解（例えばJ2EEを用いた

Webシステム）を前提としてシステム全体の構成やアプリケーション・フレームワークを作り込み、その開発と個々のプロジェクトへの展開や運用保守を行う組織的な取り組みを事例としていた。

- ・最初から作り込むのに十分な要求が整理されているわけではなく、不明な点等を適切にリスク要因として洗い出し、システム構想や設計を整理しつつアーキテクチャ設計や実験を行い、本システム開発への影響が大きくなるよう、ある時点まで強弱をつけて連続的に開発している（ウォーターフォールの要求分析、アーキテクチャ設計、詳細設計とは進まない）。
- ・うまく利用できている例では、優秀なアーキテクト（あるいは、そのチーム）を配置していることが多いが、アーキテクトの育成方法やアーキテクトが設計のために利用する手法が整理されている例はない（＝現状は基本的に属人的な解決である）。

[第三の調査]

アーキテクチャ設計よりもさらに上流に位置する要求工学の技術動向を調査した。顧客を含む多様な入力から開発される要求仕様はシステム開発の基盤となる。一方、この分野は非形式的で多様な対象を取り扱うため、実践での工学的な取り組みはあまり進んでいない（あるいは、そもそもどのように対処すべきか整理されていない）。そこで、この調査では、先進研究と実践に求められる技術の把握を目的に、第一の調査と同様の方法で主要4国際会議と1論文誌の文献（約700件）を対象として論文を100件抽出してトピックを整理し、約20名の研究者及び実務者へインタビュー調査を行った。調査結果の一例ではあるが、次のようなことが明らかになった。

- ・シナリオベース、ゴール指向といった手法や形式手法のように、研究コミュニティが活発で多くの研究成果が出ているテーマが存在する。実践での利用のためのツールや手法、ガイド等現場での適用と経験が望まれる。

- ・一方で妥当性の確認、優先順位決定、リスクやコストの認識と対処といった開発現場で望まれる人の絡んだ部分に突っ込んだテーマは、研究の対象外としているか実験的なプロジェクトや環境を対象としている場合が多い。今後、実際のプロジェクトを対象に実証されていく、あるいは、実践者の立場から整理するようなアプローチが必要とされている。

2 今後の方向

様々な技術が、先進的な方向へ進むものから実証段階へ進むものまで多く存在している。SECでは、要求の技術、設計の技術というような単なる技術的視点だけでなく、昨今のシステム障害等から求められている高い信頼性を実現するといった課題や現場の開発者が置かれている状況にもフォーカスを置きつつ、引き続き開発上の課題と技術的な解の整理を進める。また、それと同時に、今後は、産業でも利用していける解となりそうな技術を選定し、具体的な活動を開始していく。

候補となる技術がいくつもある中で、最初のトピックとして非機能要求に着手する。まずは、現場で悩みも多くアーキテクチャ設計を難しいものとしている、トレードオフやリスクを理解判断するための手段、利害関係者間で正確に意思疎通するための文書化、コミュニケーションの手段を検討することを狙う。具体性を確保する意図もあり、当面は、エンタプライズ系ソフトウェアを開発している企業でアーキテクトとして活躍している人を中心に、設計者の立場から要求として必要なものは何か、どういったことを要求して意思疎通していくべきかということを検討整理しようと考えており、2006年度は、これを整理した結果を公開する予定である。

プロジェクト見える化とは

- 下流工程編 -

SEC 研究員

長岡 良蔵

プロジェクトを成功に導くためには、プロジェクトの実施過程において、状況のモニタリング、コントロールと対策のサイクルを通して、プロジェクトの異常を検知し、分析し、対策をとり、最適な状況を維持しなければならない。これらの活動は製造業で確実な地位を築いている「見える化」という活動でまとめることができる。

一方、ITシステム開発プロジェクトは製造物の生産プロジェクトと異なり、「要件が目に見えない」、「開発プロセスが目に見えず、進捗が見えない」、「成果物の正当性が目で確認できない」ため、課題が自然と見えてくることはない。製造業ですら、課題が「見えづらい」といわれている中で、ITシステム開発プロジェクトには課題が見えるようにするための工夫がより一層必要なのは明らかである。「見えないものを見えるようにしたい」「見えたものを見えたまま知らせたい」そして「問題があれば良くしたい」という要望には、切実なものがある。「プロジェクト見える化」への取り組みは、この要望に少しでも応えることを目的としている。

1 SECでの取り組み

取り組みの1つとして、ITシステム開発で最も重要なソフトウェア開発の可視性を高め、「失敗しそうなプロジェクトを救う」観点から解決策を探っている。プロジェクトは、マネジメントをする者が「見えるようにしなければ目に見えない」。このことが情報システムの品質に大きく影響を与える要因になっている。この「見えない物をどのように見えるようにするかの工夫」がソフトウェア開発プロジェクトの成功を握る大事な鍵である。

プロジェクトの異常を見つけて直すためには、一般にプロジェクトマネージャ自身がプロジェクトの状況をチェックするとともに、定期的にPMO（プロジェクトマネジメントオフィス）等のプロジェクト外部の評価機関がチェックを行う。また、外部からみてプロジェクト異常の兆候が見られる場合にも外部からのチェックが行われる。

これらのチェックで異常を見つけた場合、さらに詳細調査が実施される。SECではこのような作業フローに対して問題把握から問題解決までの手法を探り、これを「見える化」「言える化」「直せる化」というキーワードのもとに整理した。

プロジェクトの状況を可視化するのが「見える化」、異常を検知し、調査の結果を分析して、プロジェクトの問題とレベルを明確にするのが「言える化」である。そして、問題とレベルに応じて、プロジェクトに種々の改善を実施するのが「直せる化」である。

改善には、プロジェクト内で実施できるものもあれば、社内調整をしてプロジェクト外からの支援が必要なもの、顧客との調整をしてから実施するものもある。また、改善には当面の問題を解決する暫定対応策と同じ問題を再発させないための根本対策がある。

2 これまでの成果

SECでは2005年の10月から「プロジェクト見える化」のテーマに取り組み、これまでに主に下流工程について検討し、解決書としてまとめている。下流工程から開始したのは、実際の問題が非常に明確に現れる工程であり、また具体的な対策が上流工程に比較してより具体的であることによる。また、SECで考える手法を実際に適用して実証する場として選ばれた「SEC先進ソフトウェア開発プロジェクト」がちょうど「テスト工程」に入っていたこともその理由の1つである。

3 今後の方向

2005年度は下流工程をターゲットにしたが、2006年度は、上中流工程に「見える化」の検討範囲を拡大する予定である。また、ここでもその検討範囲が広いことに鑑み、具体的に解決すべき問題設定として、ソフトウェア開発プロジェクトの「見切り発車」が日常的に行われている事態に着目し、以下を検討することとした。「見切り発車」とは、契約前に設計に着手したり、開発要員がま

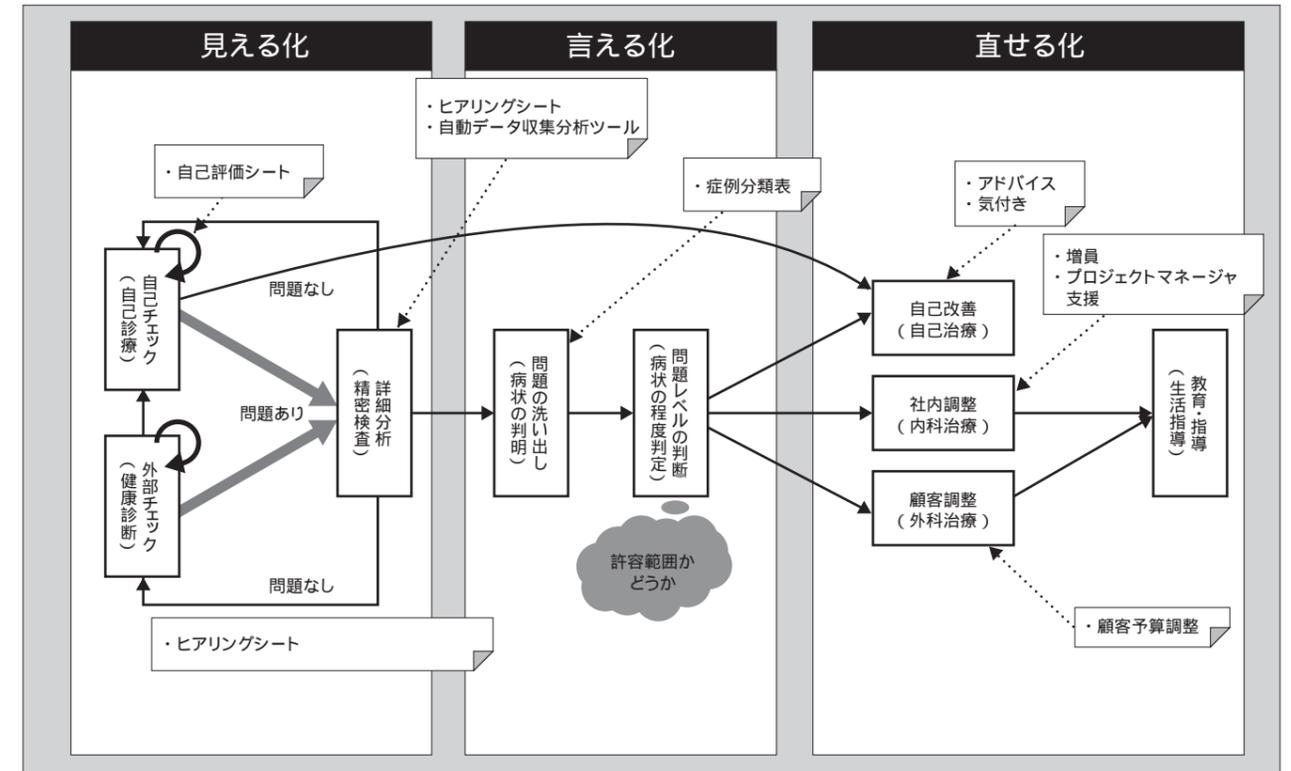


図1 プロジェクト「見える化」「言える化」「直せる化」の作業フロー

だ足りない状態から開発に進んだりすることである。

- ・「見切り発車」状況を乗り切るスーパープロジェクトマネージャの知見を形式知化する。
- ・「見切り発車」を想定してその状況を定性・定量的に可視化（見える化）する手法を案出する。
- ・上記を整理し、上中流での「プロジェクト見える化手法」を構築する。

優れたプロジェクトマネージャはリスクを予知し、ネガティブインパクトを最小にする。その背後に何らかの「見える化」手法が隠れている。

「見える化」がカバーすべき範囲は広い。現在の取り組

みはその一部に過ぎないが、SECのこのプロジェクトによって「プロジェクト見える化」するノウハウを形にし、それを着実に広く普及していくことで日本のソフトウェア産業の発展に貢献したいと切望している。



組み込みソフトウェア・エンジニアリング

SEC 研究員
平山 雅之

SECが発足して約1年半が経過した。この間にも、我々の身の回りにある機器は様々な進化を続け、数多くの組み込みソフトウェアが作られ、組み込まれ続けている。組み込みソフトウェア・エンジニアリング領域の活動は、このように増加の一途をたどる組み込みソフトウェアの品質を向上させ、効率的な開発を実現し、製品ユーザに対して安心を提供するための技術や手法の整備を追い続けている。

1 組み込みソフトウェアとは

(1) 組み込みソフトウェアの定義

この1年半のSECの活動を通じて最も多くいただいた質問の1つは、「組み込みソフトウェアとはどのようなものか」というものである。これについてSECでは「組み込みソフトウェア=製品や機器に組み込まれて動作するソフトウェア」と考えて活動している。この定義に従うと、情報家電、自動車、携帯機器や様々なOA、FA機器等もこの範疇に含まれることとなる。例えば、携帯電話等の場合、それを構成する液晶ディスプレイ等のデバイス制御、通信プロトコル制御、電話帳やブラウザといった情報処理機能は、携帯電話という製品に組み込まれる時点で、すべて組み込みソフトウェアという扱いになる。

機器や製品に組み込まれて動作するという事は、それらの機器や製品が利用される環境の多様性や製品で利用されるプロセッサやメモリ等の制約の影響を受けるといった特質がある。また、これらの製品や機器はユーザに多様性があり、結果として、ソフトウェアの仕様の多様性や不確定性の度合いも高くなる傾向にある。

(2) 組み込みソフトウェアは特別か？

このように「組み込みソフトウェア」は、「製品に組み込まれる」という点で、他のソフトウェアとは特性が異なる部分を有している。しかし、こうした相違点は、ソフトウェアとしてもつ特性の中でもごく一部と考えても構わず、その他多くの部分はソフトウェアとしての本質的

な特性に支配されていると考えることができる。

その意味では、「ソフトウェアを作る」という行為を支えるソフトウェア・エンジニアリングに関しても、組み込み/非組み込みを問わず、共通して利用できる部分が少ない。強いて非組み込みソフトウェアと組み込みソフトウェア開発の違いを考えるのであれば、昨今の組み込みソフトウェアの急激な規模拡大や開発期間の短期化を背景に、組み込みソフトウェア開発では、ソフトウェア工学に基づく適切な開発技術の普及・定着が遅れていることが最も大きな違いであると考えられる。このような問題意識を背景に組み込みソフトウェア・エンジニアリング領域では、

- ・ソフトウェア開発の原点に立ち返り、組み込みソフトウェア開発の基本動作を確実に根付かせていくための工夫をする。
- ・組み込みソフトウェアがもつ前述の様々な多様性や制約を意識して、実際のソフトウェア開発現場で適用可能な“柔らかさ”をもつ支援技術を整備していく。

という2点を強く意識して活動している。

2 主たる活動テーマ

(1) 活動の基本戦略

上記のような活動コンセプトに基づき、組み込みソフトウェア・エンジニアリング領域では、我が国の組み込みソフトウェア開発力の強化という視点で当初3年間の活動目標を下記のように設定した。

2004年度：組み込みソフトウェア開発力の基礎体力向上

2005年度：組み込みシステム開発の設計・管理精度向上

2006年度：組み込みシステム開発の開発プロセス最適化

2004年度から2005年度までの目標は、組み込みソフトウェア開発におけるソフトウェア・エンジニアリングの導入加速を進めることで、開発の基礎体力的な部分の向上や設計・管理精度の向上を目指してきた。

具体的には、後述する各部会で小冊子等を作成し、これを活用して種々の基本動作や基本技術の啓蒙を進め開発の基礎体力向上を目指した。更に組み込みソフトウェア

開発の特徴を考慮した適切な道具を整備し、提供することで、設計や管理の精度向上を目指すという基本戦略にのっとり活動を展開してきた。

(2) 活動領域

組み込みソフトウェア・エンジニアリング領域では「ソフトウェアの開発を、より円滑かつ効率的に、高品質のソフトウェアを作り上げる」ための技術の整備を大きな目標としている。この目標を達成するためには、

- ・「開発の過程でどのような作業を行うか」という開発プロセスの視点
- ・「その作業をどのような手法で実施するか」という開発技術の視点
- ・「実施した開発作業が適切かどうかを制御する」ための管理技術の視点

の3つの視点から取り組む必要があると考えている。

また、これらの視点に関しては、この数年、様々な新技術の開発提案が続いているが、正しい理解がなされないまま誤って一般の開発者に伝わってしまっている技術も少なくない。これらの新技術に関して正しい理解をする助けとなるような技術の咀嚼もまた、SECの役割と考え、活動している。

(3) 活動テーマ

活動の基本戦略を実現すべく組み込みソフトウェア・エンジニアリング領域では、現在、6つのテーマについて作業活動を展開している。これらのテーマは前述の活動部会を組織して領域の考え方に従い3つのカテゴリに分けることができる。

開発プロセスカテゴリ：このカテゴリは、組み込みソフトウェア開発で実施すべき作業を開発プロセスの視点で整理開発作業を円滑に進めることを目的としている。

開発技術カテゴリ：このカテゴリは、この開発プロセス領域で定めた作業を実際に進めるために必要となる、設計や実装、テスト等のための具体的な手法やツールを整備し、普及推進することを目的としている。

管理技術カテゴリ：このカテゴリは、開発プロセスで定めた作業が実際の開発の中で円滑に実施されるようにコントロールするためのマネジメント手法を整備し、普及推進することを目的としている。

以下では、各カテゴリの考え方やテーマ等を紹介する。

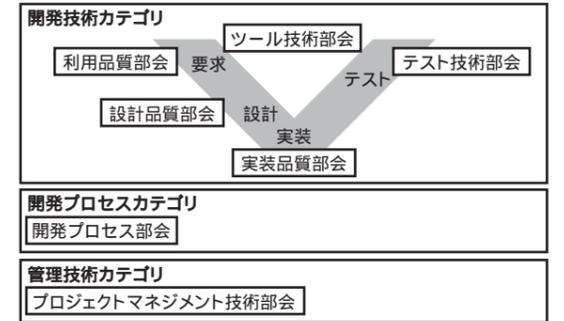


図1 活動領域とテーマ

開発プロセスカテゴリ

組み込みソフトウェアの開発プロセスの問題については、「開発プロセス部会」において集中的な議論を重ね、必要なソリューションの整備を進めている。このテーマについては、下記の2つのステップを考えており、このうちStep 1を先行させている。

Step 1 組み込みソフトウェア向け標準プロセスの整備

このステップは、組み込みソフトウェア開発で必要となる作業を標準的なプロセスとして整理することを目標としている。現在、これについては本号33頁に詳細を示すように、ほぼ、めどが付き「組み込みソフトウェア開発プロセス標準」の整備の最終段階にきている。

Step 2 組み込みソフトウェア向けSPIの整備

標準的なプロセスを整備することは組み込みソフトウェアの開発作業を適正化するための第一歩である。現実には、定められたプロセスが確実に実行されるように、現実のプロセスを確認し、問題箇所を是正していく活動が必要になる。一般には、こうした活動をプロセス改善(SPI: Software Process Improvement)と呼ぶが、様々な既存手法も参考に、組み込みソフトウェアに適したSPIの考え方を整備していく必要があると考えている。

開発技術カテゴリ

このカテゴリについては、開発のV字型モデルを参考にすると、上流の要求・設計等の技術、中流の実装技術、下流のテスト技術等技術の広がりが大きい。このため、これらの広がりを連続的に網羅するように、設計品質技術、実装品質技術、利用品質技術の3テーマを中心に検討している。また、ツール技術準備、テスト技術についても検討の準備を進めている。以下、概要を紹介する。

(i) 設計品質技術

実フィールドで発生するソフトウェアの不具合は、設計段階での誤りが原因であるものが少なくない。その意



組込みソフトウェアの組織的な開発の実現

開発プロセスガイドと開発計画書ガイドの策定

SEC 研究員
室 修治

近年の組込みソフトウェア開発の現場では、開発規模や開発プロジェクト規模の急増に伴い、種々の問題が生起している。とくに開発に係る作業やプロジェクト遂行面でのプロジェクトマネジメント等で様々な課題が発生する場合が少なくない。SECでは組込みソフトウェア開発を組織的に行うことで効率的な開発を実現すべく、組込みソフトウェア開発に適した開発プロセス技術及び開発管理技術の整備を進めている。

開発プロセス技術の整備と促進については、その第一歩として、組込みソフトウェア開発向けの開発プロセスガイドの作成を進めている。また、開発管理技術については、きちんとしたプロジェクト計画に基づく予実管理を定着させるために、まず、プロジェクトの開発計画書作成に焦点をあて、その参考としていただくためのガイドを策定している。

1 開発プロセスガイドの策定

(1) 開発プロセスに関する課題

現在、多くの組込みシステム開発では、その開発プロセス（作業）面で次のような課題が散見され、結果的に製品の品質問題を誘発させる要因となっている。

- ・組織における開発プロセスが明確に定義できていない。
- ・組織で採用している開発プロセスにヌケやモレ、曖昧なところ等がある。
- ・開発プロセスの一部しか考えられておらず、全体としての作業が網羅できていない。

このような状況を解決し、高品質の組込みソフトウェアを効率的に開発するための作業指針として「開発プロセスガイド - 最適なプロセスを構築するために -」（以下、開発プロセスガイドと呼ぶ）の策定を進めている。

(2) 開発プロセスガイドの特徴

情報システムの開発プロセスに関する国際標準（ISO/IEC12207、15288）を参考に、我が国の組込みソフトウェア開発の現場で実践されてきたソフトウェア開発

作業に関する知見を整理しており、下記のような特徴を持っている。

特徴1：ソフトウェア、システムの開発について具体的な作業のレベルで開発プロセスを整理している。

特徴2：組込みソフトウェアの開発において実施すべき作業項目を十分に網羅している。

特徴3：個々の作業の入出力を明示するとともに、作業内容についても具体的に理解の容易な表現を用いている。

(3) 開発プロセスガイドの構造

組込みソフトウェア開発に関わる作業を「プロセス」、「アクティビティ」、「タスク」、「サブタスク」という4階層で整理してある。

プロセス
組込みソフトウェア開発を進める上で、実施することが求められる作業をいくつかの塊（作業群）でとらえたもの。

アクティビティ
プロセスを実施する際の、より具体的な作業をまとめた作業群。

タスク
個々のアクティビティを実施し、達成する上で実施が求められる作業をいくつかのグループに分けて整理したもの。

サブタスク
個々のタスクを実施する場合に、実施作業をそれぞれ作業ステップに近いレベルで整理したもの。

図1に示すように、本ガイドではプロセスとして、ソフトウェア・エンジニアリング・プロセス、サポート・プロセス、システム・エンジニアリング・プロセスの3つのプロセスを対象としている。この中で、例えばソフトウェア・エンジニアリング・プロセスについては、ソフトウェア要求定義、ソフトウェア・アーキテクチャ設計、ソフトウェア詳細設計、実装及び単体テスト、ソフ

味で、ソフトウェアの構造と振る舞いを設計作業の中で確実に検討する必要がある。設計品質技術として、その第1歩にあたる設計モデリング技術に着目し、その啓蒙や設計品質の視点からのモデリングのあり方を「組込みソフトウェア開発における品質向上の勧め（設計モデリング編）」として整理し、啓蒙のための小冊子の準備を進めている。

(ii) 実装品質技術

コーディング等のソフトウェア実装については、実装レベルの標準化等を促進することを目標として、コーディング作法の整備を進めてきた。昨春にVer.0.8を公開し、多くの皆様からパブリックコメントをいただいた。これらのコメントを参考にコーディング作法Ver.1.0の整備がほぼ完了し、「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）」として整備を進めている。

(iii) 利用品質技術

組込みシステムの機能増大の中でその使い勝手が重要視されるようになってきている。利用品質技術については組込みシステムの利用者からみた満足度という視点からシステムの利用性をとらえ開発の中で反映していく方法について議論を進めている。

このテーマについても、この技術の重要性を認識していただくために「組込みソフトウェア開発における品質向上の勧め（ユーザビリティ編）」を編集した。

(iv) ツール技術及びテスト技術

組込み開発環境として様々なソフトウェア開発ツールが用意されている。しかし、これらを有効活用するという視点において、ツールの選択や活用方法等は十分に整備できておらず、開発者がツールに翻弄される局面も少なくない。ツール技術については、こうした問題意識を背景に適切なツール導入の考え方や手順を整備する方向でテーマ選定を進めている。

また、テスト技術は、ソフトウェアの高品質化のために必須の技術である。しかし、一方で、様々な手法等が数多く提案されており、これらを実際開発の中でどのように利用していくかを整理することが求められている。テスト技術については、実際の開発現場での課題等もヒアリングしつつ、様々な手法を開発現場に普及するための道筋について議論しており、その中から、方向性やテーマ選定を進めている。

管理技術カテゴリ

近年の規模の大きな組込みソフトウェア開発プロジェクトにおいて円滑なプロジェクト運営を実現するためには、プロジェクトマネジメント技術の導入が不可欠である。我々はこの問題を考えるために「プロジェクトマネジメント技術部会」を組織し、検討している。この中で、まず、注目したのは、適切なプロジェクトマネジメントを進める上での基礎となる「開発計画の策定」というテーマである。実際の現場での状況を見ると、開発プロジェクトの最終段階で、Q（Quality）、C（Cost）、D（Delivery）が当初の計画どおりに達成できないプロジェクトが半数以上にのぼっている一方で、そもそも計画の立て方に問題があったプロジェクトも数多く存在している。このためプロジェクトマネジメント技術として、まず、プロジェクト計画書に盛り込むべき内容を議論し、プロジェクト計画書の雛形策定を実施した（この結果については、本号34頁を参照）。管理技術カテゴリとしては、この先、この標準的なプロジェクト計画書をベースにしたプロジェクトの予実管理方式やプロジェクト混乱予測等についても、今後検討していく予定である。

3 まとめ及び今後の方向

以上、SEC組込みソフトウェア・エンジニアリング領域の活動概要について紹介した。現在、この領域で取り組んでいるテーマは、SEC発足の前年度に実施したソフトウェア開発力強化推進委員会準備会での答申をベースにして実行計画レベルに落とされた。この点で、既に準備会での議論から2年以上経っており、この間の周辺状況を踏まえた若干の軌道修正も必要と考えている。とくに、この間で顕著になってきた点として、システム事故や障害が相変わらず多発する傾向が続いており、経済産業省から出された「信頼性ガイドライン」にも載るように国全体でこの問題に真剣に取り組んでいくことが求められている。組込みソフトウェアとて例外ではなく、SECとしても、組込みソフトウェアの信頼性・安全性を向上させるための技術や手法、考え方を早急に整備していく必要があると考えている。2005年度まで準備フェーズとして進めてきたテーマの本格検討等も含め、組込みソフトウェア開発に関する技術を、面でサポートする体制を築き上げていきたいと考えている。

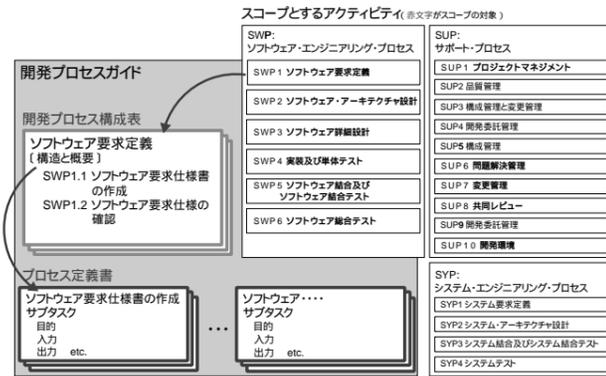


図1 開発プロセスガイドの構成



図2 開発プロセスガイド プロセス定義書

ソフトウェア結合及び結合テスト、ソフトウェア総合テストといった6つのアクティビティから構成される。さらにこの中で「ソフトウェア要求定義」というアクティビティについては、「ソフトウェア要求仕様書の作成」「ソフトウェア要求仕様の確認」という2つのタスクがあり、「ソフトウェア要求仕様書の作成」の「サブタスク」としては制約条件の確認、ソフトウェア機能要求事項の明確化、ソフトウェア非機能要求事項の明確化、要件の優先順位付け、ソフトウェア要求仕様書の作成という5つのサブタスクを定義している。とくにサブタスクレベルの定義については、図2に示すように、個々のサブタスクの入出力とその作業内容及び組み込みソフトウェア特有の注意点等を具体的に整理している。

(4) 開発プロセスガイドの利用者、利用方法

個々の組織や部門、あるいは個々のプロジェクトの特性を考慮し、それぞれに適した開発プロセスを整備する際に利用することを想定している。開発プロセスの整備が必要なシーンとしては、

ソフトウェアの開発プロセス等が未整備で、新規に組織や部門の開発プロセス標準を決める場合。

既に部門や組織の標準的な開発プロセスがあるが、実際の作業等とのずれが生じていて見直しが必要な場合。従来の開発プロセスでは対応しきれないような開発をする場合。

等が考えられる。いずれの場合にも、実際の開発プロジェクトや組織に関わるリーダーやマネージャ、あるいは、それらを開発プロセス面から支えるSEPG (Software Engineering Process Group) メンバ等が中心になって、本ガイドを参考に最適な開発プロセスの整備と実行を目指していただければと考えている。

2 開発計画書作成ガイドの策定

(1) プロジェクトマネジメント面での課題

近年の組み込みソフトウェア開発では急速に大規模化が進行しているが、その中で開発プロジェクトのマネジメントが重要な役割を持つようになってきている。その中でも、プロジェクト全体の動きをデザインするプロジェクト計画書を、いかに適切に用意できるかがプロジェクトの成否の大きな分かれ目となっている。しかし、実際の開発プロジェクトでは、プロジェクト計画書が曖昧だったり、ずさんであったりして、適切なプロジェクトマネジメントが機能せずに失敗を繰り返すプロジェクトも少なくない。このためSECでは、実際の開発プロジェクトにおいて円滑なプロジェクト運営の基礎となるプロジェクト計画書を正しく作成し、プロジェクトを適切にマネージするための参考として「開発計画書作成ガイド」の整備を進めている。

(2) 開発計画書作成ガイドの特徴

実際の組み込みソフトウェア開発プロジェクトでプロジェクト計画書を作成することを想定し、そのための指針を、計画書フォーム例、記載例、項目の目的、何故その項目が必要なのか、記載時の注意事項等を示しながら紹介している。「開発計画書作成ガイド」は、以下のような特徴をもっている。

特徴1：実際のプロジェクトでそのまま使用できるフォームを提示している。

特徴2：組み込みソフトウェアの開発において計画すべき項目を十分に網羅している。

特徴3：サンプルプロジェクトでの例を使い、計画立案のポイントをわかりやすく解説している。

このような特徴を生かすことで、体系立てた計画書を持たない組織であっても、プロジェクトを円滑に進めるためのベースとなるプロジェクト計画書を容易に作成できるようになる。

(3) 開発計画書作成ガイドの構成

開発計画書として、そのまま利用することもできる計画書フォームとその説明を中心とし、計画立案のためのテクニック、サンプルプロジェクトを使用している計画書作成例、計画チェックリスト等、計画にかかわる必要事項を網羅している。

(4) 開発計画書作成ガイドの利用者、利用方法

想定する利用者

通常、プロジェクト計画書はプロジェクトを任されたプロジェクトマネージャや組織長等が作成する。このため「開発計画書作成ガイド」の第一のユーザは、こうしたマネージャ層を対象としており、実際に計画書を作成する際の参考として利用していただくことを想定している。

また、複数のプロジェクトを同時に抱える場合、組織内でプロジェクト計画書の様式や記載内容を揃えておく必要もある。このため、組織内でプロジェクト計画書の様式を揃える役割を持った支援グループのメンバの方等にも参考にしていただくことを想定している。

1. プロジェクトの概要

1.1 プロジェクトの目的	1.5 プロジェクトの前提条件
1.2 プロジェクトの目標	1.6 プロジェクトの成果物
1.3 目標達成のための方針・手段	1.7 スケジュールと予算
1.4 プロジェクトの範囲	1.8 計画の更新

2. プロジェクトの参照・定義

2.1 参照
2.2 定義

3. 体制

3.1 プロジェクトの体制	3.3 内部体制
3.2 外部インタフェース	3.4 役割と責任

4. スタートアップ計画

4.1 見積り計画	4.3 設備、機器等調達計画
4.2 人員計画	4.4 プロジェクトの人員研修計画

5. 作業計画

5.1 作業計画	5.4 人員リソース割当て
5.2 作業項目	5.5 予算リソース割当て
5.3 スケジュール割当て	

利用方法

(i) 開発計画書を作成する場合

「開発計画書作成ガイド」の中心をなす「フォームと説明」では見開き左ページにフォーム及び記載例と、その項目が何のために必要であるかの「目的」、「記述すべき内容」、「記述の際の注意/考慮すべき内容」等がまとめられている。それぞれの項目の意図・目的を「目的」欄で確認し、「記述すべき内容」「記述の際に注意/考慮すべき内容」に沿ってフォームに記載していけば、必要な事項がもれなく記載された計画書を完成させることができるようになっている。また、計画を決めるための具体的な手法にあたるPERTやWBC等の作成テクニックについては、「計画のテクニック」に詳述している。作成後はチェックリストに従い、それぞれの項目の記載確認も行うことができる。

(ii) 計画書に基づきマネジメントを行う

実際のプロジェクトを運営していくためには完成した「計画書」をベースに行っていくが、そのポイントについては解説編の「計画書を使ってマネジメントする」にまとめられている。

このようにして「開発計画書作成ガイド」を活用することで、プロジェクトの特性にあった適切な開発計画書を策定し、それに基づいたプロジェクトのマネジメントを実践していただければと考えている。

3 まとめ

両ガイドとも上述したように「現場で実際に使ってもらえるもの」を目指し策定作業を行っている。2006年春にWeb上で公開して寄せられるコメントを勘案し、同年秋に出版する予定である。



図3 開発計画書作成ガイド (フォームと説明)



組込みソフトウェア実装品質向上

組込みソフトウェア向けC言語コーディング作法ガイドの策定

SEC 研究員
大野 克巳

組込みソフトウェアの実装は、C言語が多用されている。しかし、C言語は組込み特有のメモリ操作等が容易な反面、記述の自由度が高く、技術者の経験によって品質面での差異を生じやすい特性を持っている。こうした組込みソフトウェア実装面での品質安定を実現するため

には、コーディング規約等を整備することが有効と考えられる。SECでは、企業やプロジェクトでコーディング規約を作成・運用する人に対して、コーディング規約作成の支援を目的としたコーディング作法ガイドを作成した。

1 コーディング作法ガイドの特徴

作法は、「JIS X 0129-1 ソフトウェア製品の品質第1部：品質モデル」に準拠した主品質特性をもとに、作法概要、作法詳細に分類・階層化し、さらに、それぞれの作法にC言語に対応したルールと、その必要性を提示している。JIS X0129-1の主品質特性としては、信頼性、保守性、移植性、効率性、機能性、使用性の6つの特性が規定されている。このうち、機能性、使用性の2特性については、より上流の設計段階以前に作り込むべき特性と考え、このコーディング作法ガイドでは信頼性、保守性、移植性、効率性の4特性を大分類として採用している。

2 コーディング規約策定のためのガイド

コーディング作法ガイドは、プロジェクトや組織のコーディング規約として直接利用するものではなく、コーディング規約を策定するためのガイドとして整理した。このコーディング作法を活用したソースコード品質向上の手順としては次の2段階になる。

コーディング作法ガイドを参考に自部門のコーディング規約を整備する

策定したコーディング規約を利用してソースコード作成やチェックを行う

このコーディング規約整備については、対象製品の特性・プロジェクトの性質を考慮し、作法・ルールの取捨選択やルールの具体化を行う。このようにして策定したコーディング規約は、コーディングの際の指針としたり、ソースコードレビューの際の指針として活用することで信頼性の高い、保守しやすいコードを作成するための一助となることが想定される。

3 コーディング規約の整備

コーディング作法ガイドを用いてコーディング規約を整備する場合には、新規に整備する場合と既存の規約を見直す場合が考えられる。ここでは前者の手順を示す。

(1) コーディング規約の作成時期

コーディング時に参照するルールのみならず、関数名の命名規約等プログラム設計に関わるルールもあるため、プログラム設計以前に作成する必要がある。

(2) コーディング規約の作成方法

Step 1 作成方針の決定

プロジェクトで作成されるコードの書き方を示すコーディング規約の作成方針を決定する。例えば、安全性を重視し、便利であっても危険な機能は使用しないという書き方にするのか、危険な機能であっても注意して使用する書き方にするのか等が、この方針にあたる。

Step 2 ルールの選択

決定した規約作成の方針に従い、ルールの選択を行う。例えば、組込みソフトウェアの場合、異なるプラットフォームへの移植等を行う場合があるが、このような場合には移植性を重視し移植性に関するルールを多く選ぶ等の工夫をすることも必要になる。選択では各ルールの品質特性やルールの「選択指針」を参照とする。

Step 3 プロジェクト依存部分の定義

コーディング作法ガイドのルールには、以下の3種類がある。

- 規約としてそのまま使えるルール
- プロジェクトで規定する必要のあるルール
- プロジェクトの特性にあわせて、どのルールとするかを選択する必要のあるルール

例えば、の例としては、組込みソフトウェアでとくに重要なメモリ操作に関して、メモリ制約等を考慮し、次のi) ii) のルールのいずれかを選択できる。

- i) 動的メモリは使用しない。
- ii) 動的メモリを使用する場合は、使用するメモリ量の上限、メモリ不足の場合の処理及びデバッグ方法等を規定する。

Step 4 ルール適用除外の手順を決定

実現する機能により、コーディング時に注目すべき品質特性が異なる場合がある（保守性よりも効率性を重視しなければならない等）。この場合、定めたルール通りに

保守性4 4.他人が読むことを意識する。 [作法]保1-1 使用しない記述を残さない。		
4.1	使用しない関数、変数、引数、ラベル等は宣言(定義)しない	選択指数 規約化
[不適合例] void func(int arg){ /arg未使用 */		[適合例] void func(boid){
[備考] 使用しない関数、変数、引数、ラベル等は宣言(定義)は、削除し忘れたのか、記述を誤っているのかの判断が難しいため、保守性を[関連ルール] 7.16		

選択指針：コーディング規約を作成する際のルールの選択指針。
規約化：対象ルールが、プロジェクトごとの指針によって詳細を定める必要があるかないかを示す。また、選択や文書作成を指示するルールもこの欄で示す。

図3 コーディング作法・ルールの例

記述すると、目的が達せられない等の不具合になる可能性がある。このような場合に対応するべく、部分的に、ルールを適用除外として認めることを手順化しておく。

4 コーディング規約の活用方法

ここでは、本ガイドを用いて整備したコーディング規約の活用方法について紹介する。

(1) ソースコードを記述する際に利用

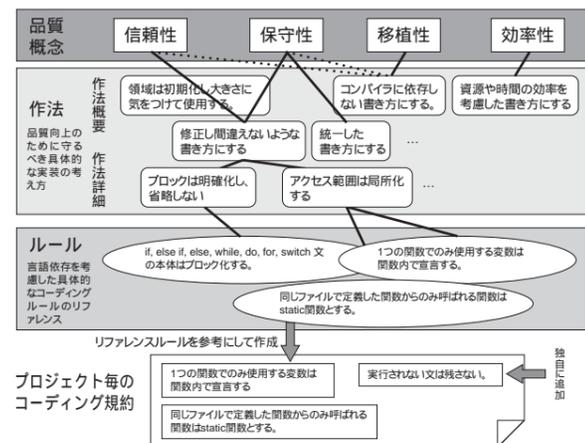
技術者がソースコードを記述する際に定められたコーディング規約を守ることで、ソースコード作成時の属人性を排除できる。また、これにより定められたソフトウェア品質を確保し、ソースコードの信頼性や保守性等を向上させることができると考えられる。これは、不具合を減らすという意味では非常に重要な利用の仕方といえる。

(2) ソースコードレビュー時の基準として利用

小規模なソフトウェアだけでなく、大規模なソフトウェア開発では、一般にコードレビューやインスペクションを行う時にレビュー基準が必要になるが、レビュー基準としてコーディング規約を利用することで、ソースコード品質の安定、向上を実現することも可能と考えられる。また、ルール以外にもコーディングミスに分類されるような凡例についても紹介を行っている。

5 まとめ

コーディング作法ガイドは、コーディング規約策定者のもとより、ベテラン開発者には、C言語の持つ特性を俯瞰し、新たな気づきのきっかけとして、初心者には、先人が築いたノウハウを体得できる実践的な教科書として、役立つと確信している。



- ・作法：ソースコードの品質を保つための慣習、実装の考え方であり、個々のルールの基本概念を示す。作法概要、作法詳細に階層化して示している。
- ・ルール：守らなければならない、具体的な1つ1つの決めごとであり、コーディング規約を構成する。このガイドではリファレンスとして示している。なお、ルールの集まりもルールと呼ぶことがある。作法、ルールの多くは、複数の品質特性と関連するが、最も関連の強い特性に分類している。

図1 品質概念、作法、ルールの関係

	品質概念	作法詳細	作法概要数	ルール数
1	信頼性1	領域は初期化し、大きさに気を付けて使用する	3	7
2	信頼性2	データは範囲、大きさ、内部表現に気を付けて使用する。	8	21
3	信頼性3	データが保証された書き方にする。	7	12
4	保守性1	他人が読むことを意識する。	15	29
5	保守性2	修正し間違えないような書き方にする。	4	6
6	保守性3	プログラムはシンプルに書く。	5	11
7	保守性4	統一した書き方にする。	7	20
8	保守性5	試験しやすい書き方にする。	3	4
9	移植性1	コンパイラに依存しない書き方にする。	5	12
10	移植性2	移植性に問題のあるコードは局所化する。	1	3
11	効率性	資源や時間の効率を考慮した書き方にする。	1	4
	合計		59	129

図2 コーディング作法ガイドの作法・ルールの構成



組込みスキル標準(ETSS)

SEC リサーチフェロー
大原 茂之

我々の身の回りは組込みシステムを用いた様々な機器で溢れている。組込みシステムの中でも組込みソフトウェアは各機器固有の機能を実現するソフトウェアである。組込みソフトウェア開発技術は、2004年版及び2005年版組込みソフトウェア産業実態調査報告書から、我が国のあらゆる産業に横断的に使われている技術であることが明らかになった。同時に、この報告書からは次のような問題点も浮き彫りとなった。

- 組込みソフトウェア開発技術者が不足していること
- 欧米企業に比べて我が国の組込みソフトウェア開発技術者スキルの標準化が遅れていること等

すなわち、我が国の製造業の共通基盤ともいえる組込みソフトウェアの開発力が、実は技術者の熟練度と勤勉さに頼り、技術者に高い負荷がかかっているという、ある種の脆弱性が見えてきたといえよう。

1 組込みソフトウェアの特性

上記の脆弱性は、技術全体のどこに位置づけられるかを考えるには、組込みソフトウェアによる製品の開発工程をみておく必要がある。図1に示すように、組込みソフトウェアの拡大に伴い、製造工程は軽くなり開発工程に負荷がかかるようになってきているのである。この開発工程に組込みソフトウェア技術者という人材が必要で

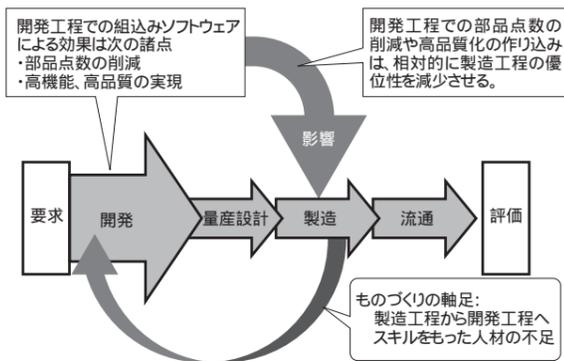


図1 製品開発製造と組込みソフトウェア

あり、今後、さらに重要度を増す開発工程に脆弱性が存在しているのである。

したがって、こうした脆弱性の問題は、我が国の機械機器の種類や開発特性に合わせて産業横断的に解決する戦略が重要であり、国が取り組むべき問題である。

SECでは、この問題解決に向けて2つの観点から取り組んでいる。1点目は、組込みソフトウェア開発スキル領域に取り組んでいるもので、組込みソフトウェア開発技術を担う人材に関するものである。2点目は、組込みソフトウェアエンジニアリング領域が組込みソフトウェア開発技術を整備・体系化において取り組んでいる。

2 2005年度の取り組み

人材のスキルに関する強化策は、次の3つの取り組みに分割できる。

組込みソフトウェア開発技術者が技術革新を進めることができるようにし、かつ産業横断的に使用できるスキル標準の体系化である。これは、キャリア開発や教育カリキュラムのベースになる。

組込みソフトウェア開発の専門性と責任性によって職種を定めたキャリア基準を策定する。

産業横断的に使える組込みソフトウェア技術者向けの教育研修基準を策定する。

これらはそれぞれ、スキル基準部会、キャリア開発部会、教育部会において検討した。

さらに2005年度スキル基準部会において策定したスキル基準Version 1.0の有効性を確認するために実証実験WGを設置し、スキル基準導入実証実験を実施した。また、2005年度教育部会において策定した組込みシステム開発未経験者向け教育カリキュラムを題材として、研修コースの開発プロセスを策定するために、研修コース開発プロセスWGを設置した。



3 技術とスキルの峻別

スキル標準として、スキル基準部会、キャリア開発部会、教育部会の各部会で検討を行うにあたり、「技術とスキル(技能の関係)」を再度確認し、各部会での検討を進めた。

技術とスキルの関係を図2に示した。基本的に技術は知識であり、知識を形にした例としてツールがあり、このツールを使いこなす能力がスキルとなる。技術は個人に属するものではないが、スキルは個人に属するものである。

技術者とは、こうした工程もしくはサブ工程を設計・開発・実現できる人材のことであり、この工程にかかわるPDCAを回せる人材ということになる。

そして、この工程にかかわるPDCAの大きさと回転周期を技術開発力とみることができる。

4 主たる活動成果

2005年度の各部会の活動成果は次のとおりである。

- スキル基準部会の成果
 - ・スキル基準 (Version 1.1)
- キャリア開発部会の成果
 - ・キャリア基準 (Version 1.0)
- 教育部会の成果
 - ・教育研修基準 (Version 1.0)
 - ・教育ガイドライン(案)

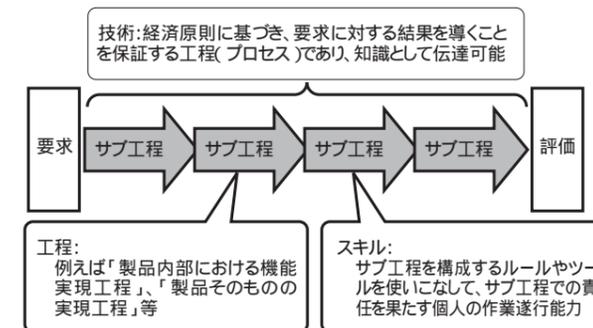


図2 技術とスキルの関係

研修コース開発プロセスWG

- ・教育カリキュラムデザインガイド

実証実験WG

- ・スキル基準導入実証実験報告書

なお、実証実験WGに関しては、2004年度に策定したスキル基準 (Version 1.0) の導入による実証実験を行ない、スキル基準 (Version 1.1) に反映することを最大の目標とした。ただし、実証実験を行うに際し、扱う情報の多くが実験に協力いただいた各企業の秘密情報に触れる部分があるため、部会活動ではなくSEC研究員のみでの活動とした。収集したデータについてもその扱いには慎重に対応した。

実証実験に協力いただいた企業は、RTECH会(ルネサステクノロジ特約店による任意団体)、沖通信システム株式会社、株式会社KSK、株式会社CSKシステムズ、東芝情報システム株式会社、日本電気株式会社、株式会社ベリサーブ、ヤマハ株式会社である(五十音順)。

5 2006年度の活動予定

2006年度の各部会の詳細な活動は、概ね次のような方向を考えている。

- スキル基準部会
 - ・組込みスキル基準 (Version 2.0)

これまでは、組込みソフトウェアを主体にスキル基準を策定してきたが、2006年度はSoCといったハードウェア技術に拡張する。

キャリア開発部会

- ・キャリア基準導入ガイドの作成

教育部会

- ・組込み技術者教育指導要領の作成

6 おわりに

各部会の成果物をすべてVersion 1.0以上にすることができた。実証実験等では多くの企業の皆様のご協力をいただいた。関係各位に、この場をお借りして感謝申し上げる次第である。



ETSS スキル基準 Version 1.1

SEC 研究員
佐藤 和夫

スキル基準部会は「スキル基準 Version1.0」をさらに使いやすいするための検討及び実証実験の成果を反映させることを課題として活動した。

1 スキル定義の見直し

スキルとキャリアの関係を明確にすべく以下のように修正した。

(1) スキルの定義について

スキルについて、経験が必須であるのかという点について検討を行い、スキルとは、あくまでも作業の遂行能力を指すものであり、キャリアとは独立に評価できなければならない。すなわち、キャリアは経歴書で評価をし、スキルは「～ができる」ことを評価する。実務には職歴が含まれるので、スキルとキャリアを明確に区別するために、スキルの定義から実務経験を削除した。

(2) レベルの概念について

スキル定義の明確化に伴い、スキルレベルについてもキャリアを含むような文言を検討し、次のように修正した。

- レベル4（最上級）：新たな技術を開発できる
- レベル3（上級）：作業を分析し、改善・改良できる
- レベル2（中級）：自律的に作業を遂行できる
- レベル1（初級）：支援のもとに作業を遂行できる

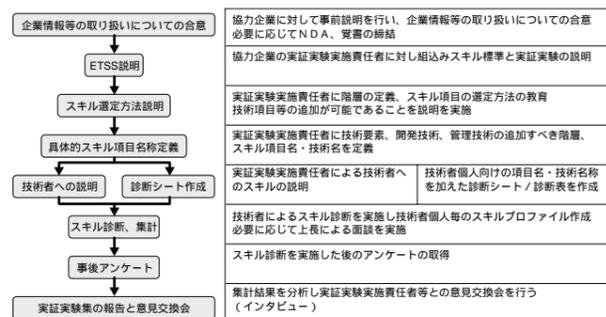


図1 導入手順



ただし、これらはスキルレベルの定義自体の変更ではなく、レベルの概念を表す文言の変更である。

(3) スキルの評価について

技術要素のスキル評価について「作れる」と「使える」をさらに明確に区別できるようにするために、「実現することができる」と「組み込むことができる」とした。

また、開発技術、管理技術のレベル4（最上級）についての検討を行い、使える技術のレベル4が作れる技術であるとした。このほか、手法とツールのレベル4には異なる判断基準が存在するので、開発技術と管理技術のスキル評価要件を区別した。

手 法：新たな手法を開発できる

ツ ール：ツールの改善点を提案できる

2 実証実験

実証実験では、組込みソフトウェア開発関連企業においてスキル基準を導入し、その実現性、妥当性、効果等の検証を行った。

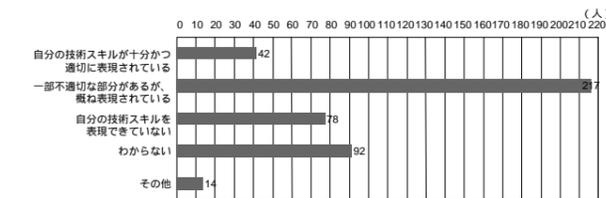


図2 技術者 スキル診断シートについて

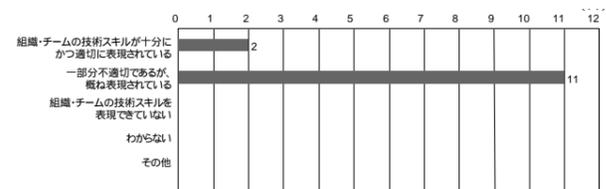


図3 管理者 組織・チームに対するこれまでのスキルに関するイメージと比較して

(1) 導入手順

企業ごとにスケジュールは異なるが、基本的な導入手順は図1の通りである。

(2) 実験結果

実証実験に協力いただいた企業の技術者445名、管理者13名からアンケートを回収することができた。以下はアンケートから判断できた内容である。ただし、管理者については人数が少ないため、以下の判断は、すべて技術者のアンケートに基づくものであり、管理者からの回答については、参考としてのみ取り扱う。

スキル基準で技術者の保有スキルが表現できていると判定できる

スキル評価シートについての設問（図2）では、259名から保有スキルが表現できているとの回答を得た。

自分の技術スキルについて問う設問（図4）では、「今後強化すべき技術スキル（短所）を明確にできた」（174名）、「自分の技術スキルの分布や傾向を把握できた」（161名）の回答が多数を占め、技術者自身のスキルを把握するのに有効であると評価された。参考であるが、管理者もスキル基準で技術者の保有スキルを把握できると評価していることがわかる。

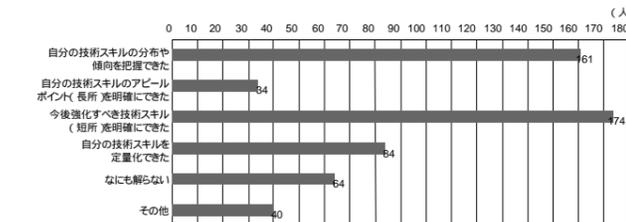


図4 技術者 自分の技術スキル

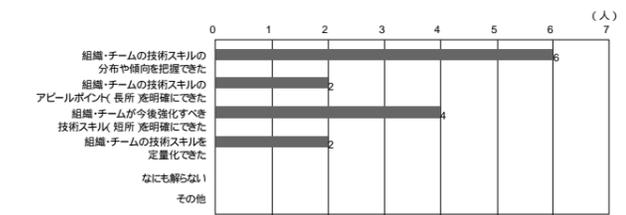


図5 管理者 組織・チームの技術スキルについてどのようなことがわかりましたか

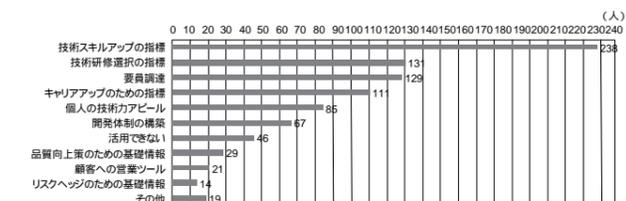


図6 技術者 スキル分析はどのような局面で活用できそうですか

技術者のスキルアップガイドとして使える

スキル分析をどのような局面で活用できそうかという設問（図6）では、「技術スキルアップの指標」（238名）、「技術研修選択の指標」（131名）、「要員調達」（129名）、「キャリアアップの指標」（111名）と、教育、調達の指標として使えるとの回答を多数得た。参考であるが、管理者も技術者のスキルアップの指標として使用できると評価している。

スキル基準フレームワークは有効

推進者へのインタビュー及び技術者・管理者のアンケートにより、スキル基準は技術者の保有スキルを表現することができ、技術者のスキルアップガイドとして有効であることから、スキル基準フレームワークが企業にとり有効であることが明確になった。

今後の課題

スキル分析の改善点を問う設問（図8）では、「スキル評価レベルの定義が曖昧である」の回答が多かった（170名）。参考であるが、管理者の回答でも「スキル評価レベルの定義が曖昧である」の回答が多かった（7名）。スキル診断シートを作成する場合、記載するスキル評価の文言を明確にしていくことも重要な課題である。

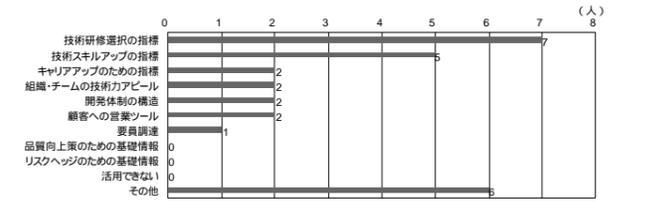


図7 管理者 スキル分析はどのような局面で活用できそうですか

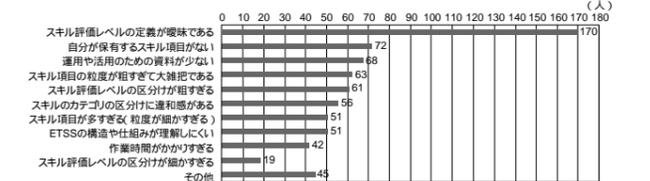


図8 スキル分析で改善すべきと考えられる点

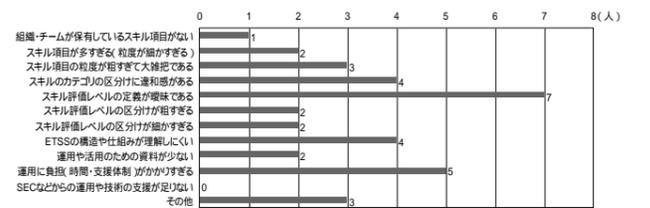


図9 スキル分析で改善すべきと考えられる点



ETSS キャリア基準Version 1.0

SEC 研究員
渡辺 登

キャリア基準部会は、組込み技術にかかわるキャリアVersion1.0を定義することを目的に活動を行った。

1 キャリアの定義

組込みソフトウェア開発を目的とする人材を育成及び活用するには、専門的能力の育成と専門的能力を生かす職場の提供が必要である。

キャリア基準は、組込みソフトウェア開発における職種を定義（名称、職掌や責任等）する。これは、エンジニアリング観点での専門性を考慮したものである。

IT系で導入が進んでいるITスキル標準と同様の『キャリアフレームワーク』を用いて、組込みソフトウェア開発（一部、組込みシステム開発）における職種を10職種定義している（図1）

2 職種の定義

プロダクトマネージャ

経営的観点のもとに、製品の企画・開発・製造・保守等にわたる製品ライフサイクルを統括する責任者。

プロジェクトマネージャ

製品開発プロジェクトの構築ならびに遂行にあたり、プロジェクトを計画・指揮・監督する責任者。

ドメインスペシャリスト

特定の技術・製品分野について高度で専門的な知識や開発経験を有する専門技術者。

システムアーキテクト

システムの利用・開発等の要件を満すシステム構造ならびに開発プロセスを設計する技術者。専門分野として、組込みプラットフォーム開発と組込みアプリケーション開発の2つを定義。

ソフトウェアエンジニア

ソフトウェアの各開発工程において開発・実装・テスト作業を担当する技術者。専門分野として、組込みプラットフォーム開発と組込みアプリケーション開発の2つ

を定義。

ブリッジSE

組織的・地理的に分散するプロジェクト組織間の調整作業を担当する技術者。

開発環境エンジニア

プロジェクトで使用するツール・設備等、開発環境の設計・構築、運用を担当する技術者。

開発プロセス改善スペシャリスト

開発プロセスとその実施状況をアセスメントし、改善の推進を担当する専門技術者。

QAスペシャリスト

プロジェクトの全工程において品質確保・維持・向上の推進を担当する専門技術者。

テストエンジニア

テスト設計、テスト実行等のテスト作業の実施を担当する技術者。

3 キャリアレベルの定義

図2に示すように、ビジネスやプロフェッショナルとしての価値創出に応じたレベルをキャリアレベルとして、7段階を定義している。

職種	プロダクトマネージャ	プロジェクトマネージャ	ドメインスペシャリスト	システムアーキテクト	ソフトウェアエンジニア	ブリッジSE	開発環境エンジニア	開発プロセス改善スペシャリスト	QAスペシャリスト	テストエンジニア
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										

図1 職種/専門分野とキャリアレベル

4 職種ごとの責任

キャリア基準（Draft）では、職種ごとの活動領域を例示したが、開発業務における活動領域は状況や組織によって異なる。今回のキャリア基準では、職種ごとに要求される責任を明示し、活動領域の例示はしていない。責任を果たすべき、必要なスキルを保有し、必要な活動領域で貢献するといったポリシーとしている。

5 技術以外のスキル

キャリア基準のキャリアレベルは、ビジネスやプロフェッショナルとしての価値創出に応じたレベルとしている。これは、技術スキルのみでは達成できるものではなく、技術以外のスキルが必要となる。したがって、スキル基準の技術スキル以外に、キャリア基準では「パーソナルスキル」と「ビジネススキル」を定義する。

これにより、ビジネスやプロフェッショナルとしての貢献を実現する職種の定義を行っている。これらのスキ

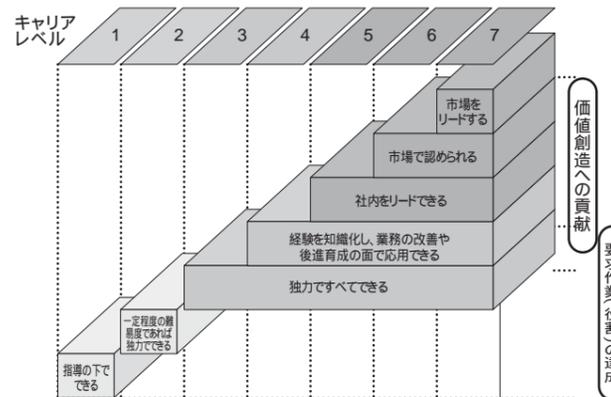


図2 キャリアレベルの定義

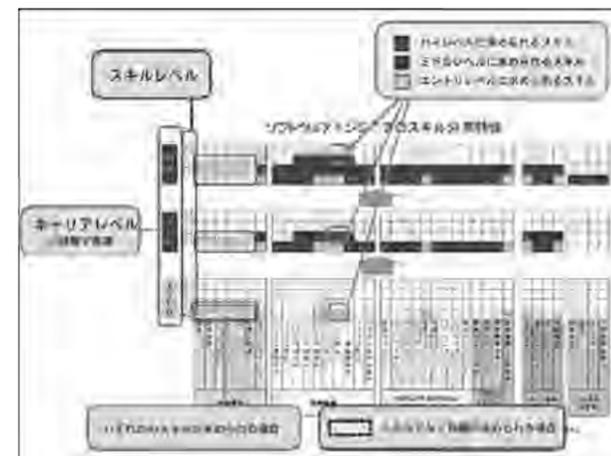


図3 スキル分布特性

ルを定義することで、組込みソフトウェア開発に従事する開発者が、技術のみに偏ることなく育成されることを期待している。

6 スキル分布特性

職種ごとに求められるスキルを定義するもので、これは職種のレベルごとに、責任を果たす上での必要条件としてのスキルを提示し、人材育成と人材活用を実現するためのベースになるものである（図3、表1）

キャリアレベルは、エントリー・ミドル・ハイの3分類ごとに、スキル基準で定義される「技術要素」「開発技術」の第一階層ごと、「管理技術」は第二階層ごとに、求められるスキルレベルを定義している。また、前述の「パーソナルスキル」「ビジネススキル」は第一階層ごとに求められるスキルレベルを定義している（表2）

実際の開発においては、具体的に作業をする（手を動かす）ことは求められないが、知識や情報をもとにした判断が必要な場合も多く見受けられる。これらについては、スキルのみでなく、求められる知識なども表現することで、人材育成にも適用可能としている。

表1 職種ごとの責任

職種名称	責任	
	責任の範囲	責任の例
プロダクトマネージャ	商品開発の事業	収益、貢献
プロジェクトマネージャ	プロジェクト	品質、コスト、納期
ドメインスペシャリスト	技術の展開	プロダクト（商品）開発の効率性
システムアーキテクト	システム構造・実現方式	開発の効率性・品質
ソフトウェアエンジニア	ソフトウェア開発の成果物	品質、生産性、納期
ブリッジSE	外部組織との共同作業	品質、コスト、納期
開発環境エンジニア	開発環境の品質	使用性、作業効率
開発プロセス改善スペシャリスト	組織の開発プロセス改善実施	プロセス改善効果
QAスペシャリスト	プロセス品質 プロダクト品質（企業による）	出荷後の品質問題
テストエンジニア	システムの検証	品質、テスト効率性、テスト納期

表2 パーソナルスキルとビジネススキル

スキルカテゴリ	第一階層	説明
パーソナルスキル	1 コミュニケーション	話す、聞く、書く等
	2 ネゴシエーション	質問、調査、主張等
	3 リーダーシップ	能力開発、時間管理、動機付け等
	4 解決	着眼・発想、問題発見・分析、論理思考等
ビジネススキル	1 経営	分析、戦略、評価等
	2 会計	財務分析、経理等
	3 マーケティング	分析、市場調査、戦略等
	4 HCM	人事戦略、要員管理、能力開発等

HCM: Human Capital Management

ETSS 教育研修基準Version 1.0

SEC 研究員
関口 正

スキル領域の教育部会では、組込み技術教育に関する定義を行う「教育研修基準」を作成するために、次のような活動を実施した。

- 教育カリキュラムフレームワークの策定
- ETSSのスキル基準やキャリア基準と連携した組込みシステム開発分野の人材育成を実現するため教育や訓練の仕組みの定義
- 教育カリキュラムデザインガイドの作成
- 昨年度の成果物である「組込みシステム開発未経験者教育カリキュラム」等を、実際に現状に即した教育カリキュラムを構築・運用するためのプロセスの明示
- 教育カリキュラムガイドラインの策定
- 昨年度の成果物である、「組込みシステム開発未経験者教育カリキュラム」を適切に活用するための指針を策定

1 教育カリキュラムフレームワーク

人材育成を目的とした、教育カリキュラムを構成する構造や仕組みを、組込みスキル標準（ETSS）では教育カリキュラムフレームワークと定義する（図1）。

ETSSにおける、教育カリキュラムフレームワークの構成要素を以下に示す。

- 教育対象**
教育カリキュラムの実施対象となる受講者の人材像。ETSSのスキル基準や、キャリア基準のフレームワーク等を用いて定量的に可視化する。
- 教育目標**
教育カリキュラムが目標とする人材像。教育対象と同様に、ETSSのスキル基準や、キャリア基準のフレームワーク等を用

- いて定量的に可視化する。
- 教育カリキュラム**
教育対象とする人材（受講者）を、目標とする人材像（あるべき姿）へ育成するために、必要となる科目を適切な順序で組み合わせたもの。
- 科目**
特定の技術分野に対する知識やスキルを習得するために、必要となる教育項目の組み合わせ。
- 教育項目**
教育や訓練で習得すべき技術項目。ETSSの教育カリキュラムフレームワークでは、スキル基準やキャリア基準によって定義されるスキルカテゴリに準ずる形式で表現を行う。

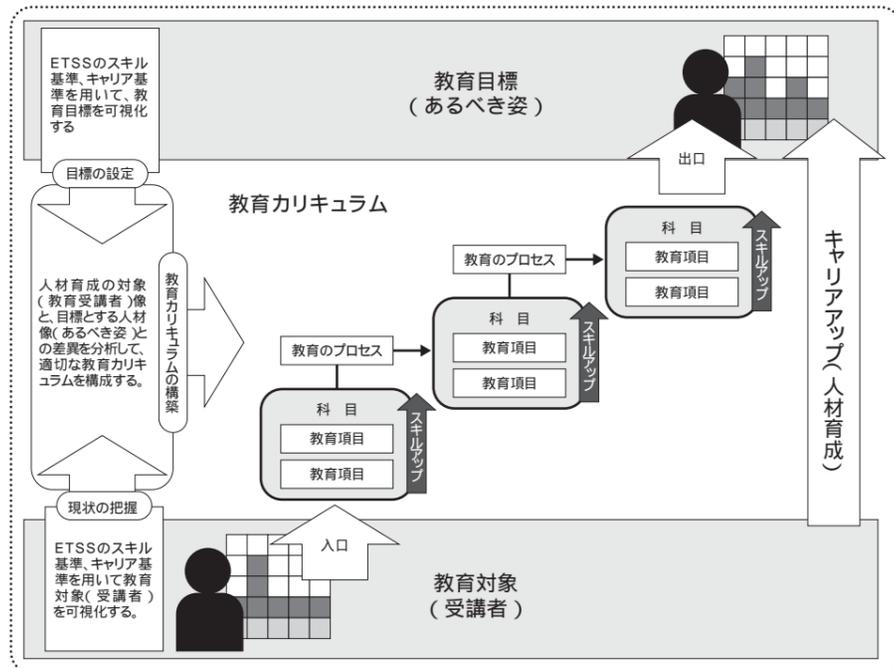


図1 教育カリキュラムフレームワーク

2 教育カリキュラムとスキル基準

前述のとおり、教育カリキュラムは科目と適切な履修順序によって実現される。

その科目が、どのような技術項目に関係するのかをETSSのスキルフレームワークを利用して関連付けを行う。

この関連付けにより、教育カリキュラムを受講するために必要な知識やスキルや、受講することによって習得することが期待できる知識やスキルを、定量的に明示できる。

これは、スキルアップを目的とした人材育成計画や教育の効果測定や評価との連携を実現し、人材育成計画や教育カリキュラムへの結果を反映することが可能となる。

3 教育カリキュラムとキャリア基準

教育カリキュラムの教育目標や教育対象を、ETSSのキャリア基準により提示されたスキル分布特性を用いることで、キャリアアップやキャリアシフトを実現するための教育カリキュラムを構成することができる。

図2は、キャリア基準で提示されたスキル分布特性を利用した、特定の職種におけるキャリアアップを目的とした教育カリキュラム構築のアプローチイメージである。

4 教育カリキュラムデザインガイド

組込みソフトウェア開発分野に即した、教育カリキュラムを適切な形で開発・実施するためには相応の手順が必要となる。

教育カリキュラムを開発・運用する知見や経験のない組織では、これらの手順を試行錯誤しながら模索することになる。教育カリキュラムを開発するために必要となる手順や留意点をガイドする「教育カリキュラムデザインガイド」を中小企業の人材育成担当者向けに作成することとした。

教育カリキュラムデザインガイドでは、教育カリキュラムを実装する工程（プロセス）を、6段階のステップで提示している。

- 人材育成計画立案
- 教育計画立案
- 科目設計
- 教材制作・調達
- 実施
- 評価

5 教育カリキュラムガイドライン

組込みスキル標準（ETSS）の教育研修基準で定義された内容に準拠した教育カリキュラムが開発・普及していくことが予想される。

その際に、ETSSの教育研修基準に準拠した、実際の教育カリキュラムのあり方や、教育カリキュラムの信頼性と品質の向上を目的としたガイドラインが必要との検討がなされた。

このような組込み技術教育のガイドラインを提供するために、実証実験や各種調査等を実施し、検討をすすめていく。検証結果がまとまり次第、公開する。

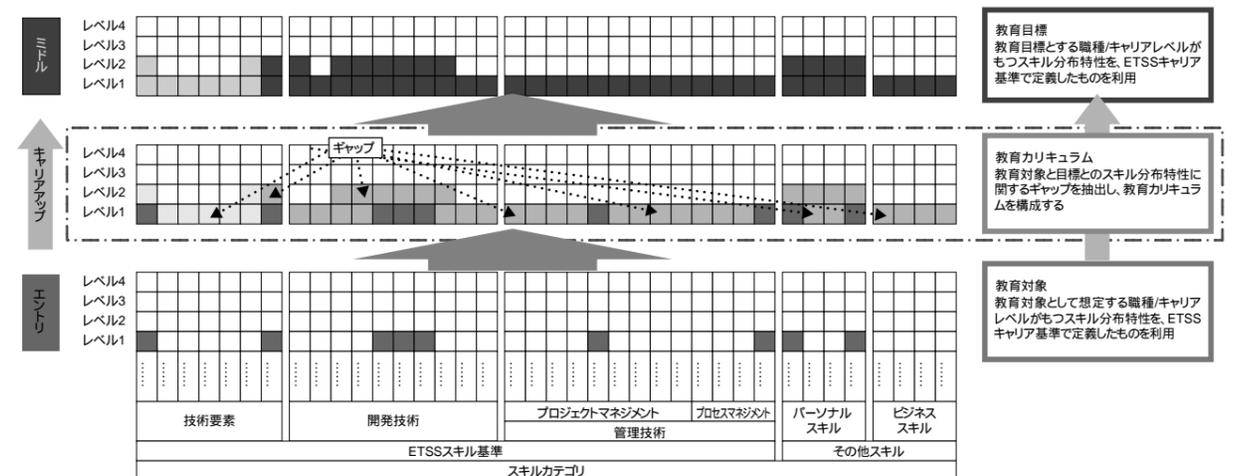


図2 教育カリキュラムとキャリア基準との連携



先進ソフトウェア開発プロジェクト

SEC 研究員
樋口 登

SEC journal No.5のProject Report[HIGUCHI2005]では、先進ソフトウェア開発プロジェクトの活動内容について述べたが、本稿では、2005年度の成果と課題をまとめ、今後の方向性について述べる。

1 2005年度の活動成果

先進ソフトウェア開発プロジェクトの2005年度の活動の主な成果を中心にまとめると、以下のようになる。なお、活動内容の詳細については、SEC journal No.5[HIGUCHI]を参照されたい。

EPM[MATSUURA2006]によるデータ収集と分析
構成管理ツールや障害管理ツール等の情報を分析することにより、以下のような状況や予兆を検出することができた。

- ・修正規模の大きな障害があった
- ・テスト工程に入ってから仕様の変更があった
- ・ファイルの構成が変更された
- ・障害が一時的に多発している
- ・長期間解決されていない障害がある
- ・障害の解決が一時的に滞留している
- ・障害が多発しているコンポーネントがある
- ・特定の要因による障害が多い
- ・修正工数の多い障害がある
- ・構成管理ツールへのソースコードの登録が一括して行われている
- ・障害管理ツールへの障害の登録、変更が一括して行われている
- ・修正漏れの可能性がある
- ・常に同時に更新されているファイルがある
- 競合他社にはソースコード等の詳細部分を開示しない状況にあって、上記の分析結果のフィードバックはプロジェクトマネジメントにとって有用であった。
- コードクローン分析
- ソースコードに含まれるコードクローン（類似コード

片）を分析することにより、以下のような可能性や予兆を検出することができた。

- ・共通化できる部分がある
 - ・ほぼ同じソースコードが別のコンポーネントで使われている
 - ・同様の処理部分をコピーして作成したコードがある
 - チェックシートによる「見える化」アプローチ
 - チェックシートによる自己診察とヒアリングを通して、以下の項目を提示することができた。
 - ・プロジェクト遂行上で問題になりそうな項目
 - ・プロジェクトマネジメントにおける改善項目
 - ・改善のための具体例
 - ソフトウェアエンジニアリングへの認識
 - 上記のような活動を実施することにより、プロジェクト参加者に対して以下のような効果が見られた。
 - ・定量データを収集する意識付けが行われた
 - ・プロジェクトマネジメントの必要性を再認識した
 - ・リスク管理の重要性を認識した
- 活動に関連した発表
先進ソフトウェア開発プロジェクトにおける様々な活動内容に関連して、表1のように、広く紹介された。

2 課題

データを収集・分析し、分析結果を開発者に説明して議論を行い、必要に応じて改善策を検討するという一連の活動を通して、いくつかの課題が明らかになった。

- ・EPM導入時の負荷が大きい
- ・データ項目の定義が曖昧で入力データが間違っていることがある
- ・手作業による分析のためにフィードバックまでに時間がかかる
- ・EPMの運用ルールが守られていないため入力データの精度が低く、分析ができない場合がある

3 今後の方向

前述の課題に対応するため、及び新たな分析を試みるために2006年度は、以下の取り組みについて検討している。

EPMの容易な導入を実現
2005年度はEASE（Empirical Approach to Software Engineering）プロジェクト[EMPIRICAL][EASE]の支援によってEPMを導入したが、より手軽にEPMを導入できるように、IPAでは、委託先を公募[FUKYUTOOL]し、EPMのパッケージ化を行う。

このパッケージにより、プロジェクト単位等、EPMの専門家がいない場合やより小規模な単位でのEPMの導入を可能にし、定量データ収集環境の普及を図る。

運用ルールの周知徹底
プロジェクトの途中から参加する開発者もいるため、運用ルール等のドキュメントについては、開発工程が変わるとき等、適宜説明会を開催し周知する。

EPMの運用を支援
入力データの精度を上げ、運用ルールが徹底されないことによって分析ができなくなることを防ぐために、入力データと運用方法を頻繁に確認し、徹底するように支援する。

より速い分析とフィードバック
より有効なフィードバックを実現するために、できるだけ分析を自動化し、分析結果を短時間で出せるようにする。これにより、データ収集からフィードバックまでの時間を短縮し、問題が予見される場合には早期に対策を検討できるようにする。

仕様書を分析
2005年度はテスト工程での分析が中心だったが、2006年度は上流工程も含めた分析を行う。

仕様書のレビュー状況や変更状況を分析することにより、上流工程の品質を見極める。

要員のスキルとプロジェクト状況の分析
プロジェクトに参加している開発要員のスキルを評価し、プロジェクトの状況との関係について分析を行う。

情報の発信
上記活動によって得られた成果等については、SEC内で、他の部会と連携する等により有効に活用するとともに、積極的に公開し、各種手法の紹介や普及に努める。

他の部会との連携

先進ソフトウェア開発プロジェクトは他の部会で開発・検討されている様々な手法の実践・実証の場でもある。

2006年度も引き続きCOSE（Consortium for Software Engineering、ソフトウェアエンジニアリング技術研究組合）プロジェクトを中心に、プロジェクト見える化部会や定量データ分析部会とも連携してソフトウェアエンジニアリングに関する有効な手法の確立を目指して活動を行う。

参考文献
[HIGUCHI2005]樋口登：先進ソフトウェア開発プロジェクト，SEC journal No.2, 2005
[MATSUURA2006] 松浦清，神谷芳樹，樋口登：先進ソフトウェア開発プロジェクト Part 1，SEC journal No.5, 2006
[EMPIRICAL]http://www.empirical.jp/research/epm.html
[EASE]http://www.empirical.jp/
[FUKYUTOOL]http://www.ipa.go.jp/software/fukyutool/

表1 活動に関連した発表

2005年5月18日 IPAX2005 近藤 弘志：次世代 ITS のソフトウェアアーキテクチャとプラットフォーム http://sec.ipa.go.jp/event/20050518a.php
2005年6月23日 エンピリカルソフトウェア工学研究会 2005年度 第1回 神谷 芳樹：先進ソフトウェア開発プロジェクトについて http://www.empirical.jp/research/katsudo.html
2005年8月24日 NASSCOM Quality Summit 2005 鳥居 宏次：The Software Industry in Japan and Empirical Software Engineering http://www.nasscom.org/qualitysummit/
2005年9月9日 第24回ソフトウェア品質シンポジウム 松本 健一：エンピリカルソフトウェア工学の現状と展望 http://www.juse.or.jp/software/symposium_20050908_2.html
2005年10月3日 情報化月間記念特別行事 神谷 芳樹：実証プロジェクト + 可視化 + 定量データ報告 http://sec.ipa.go.jp/event/20051003.php
2005年11月2日 第5回クリティカルソフトウェアワークショップ 肥後 芳樹：コードクローン分析ツール Gemini を用いたコードクローン分析手順 http://www.jsfws.info/jaxa-ista/uhrsw/text/program_jp.html
2005年11月8日 IWFST2005(International Workshop on Future Software Technology 2005) 神谷 芳樹、菊地 奈穂美、松村 知子、Mike Barker、松本 健一、岩村 聡：A Research Framework for Empirical Software Engineering Collaboration and Its Application in a Software Development Project http://www.sea.jp/Events/ws/20051108.html
2005年11月16日 ISESE2005(4TH INTERNATIONAL SYMPOSIUM ON EMPIRICAL SOFTWARE ENGINEERING) 神谷 芳樹、菊地 奈穂美、松村 知子、岩村 聡、Mike Barker、松本 健一：An empirical trial of multidimensional in process measurement and feedback on a governmental multi-vendor software project http://attend.it.uts.edu.au/isese2005/program.htm
2006年3月16日 エンピリカルソフトウェア工学研究会 2005年度 第4回 ・松本 健一：EASE プロジェクトの状況について ・楠本 真二：コードクローン分析、現場へのフィードバックの事例報告 ・松村 知子：障害修正工数に関する要因分析事例報告 http://www.empirical.jp/research/katsudo.html



大阪大学大学院 (http://www-ise4.ist.osaka-u.ac.jp/) との共同研究

ベイズ識別器に基づく混乱予測に 利用するメトリクスの統計的選択

大阪大学大学院情報科学研究科
助手

水野 修

大阪大学大学院情報科学研究科
博士後期課程1年

安部 誠也

大阪大学大学院情報科学研究科
教授

菊野 亨

SEC 研究員

菊地 奈穂美

SEC 研究員

平山 雅之

ソフトウェア開発プロジェクトの安定的な実施は、多くのソフトウェア開発企業における最重要課題の1つである。我々はベイズ識別器によるプロジェクト成否の予測に取り組んでいる[MIZUNO2005]。

2006年度は、まず、上期にはメトリクス・仮説検討、妥当性検討として、適用による効果が期待される対象や問題、そこで利用できる予測技術の検討、適用方法等に関する仮説を立てることを目標とした。下期には、事例への適用検討として、パイロット事例を取り上げ、検討した予測手法を試験的に適用することを目標とした。

1 成果のポイント

本研究では、開発現場から収集されるメトリクスデータを利用して、品質、コスト、工期のそれぞれについて、成否の予測を行った。データとして利用したのは、企業横断的に収集されたものであったが、このデータでの予測はあまり良い結果でなかった。そこで一企業からのデータを収集し、予測の実験を再度行った。

各視点の予測に使用するメトリクスの選び方によって予測の結果が変わるため、本研究では、メトリクスを選択する2つの方法を考えた。方法1は専門家による選択であり、方法2は統計的検定による選択である。そして、各方法で選択したメトリクスを使用してベイズ識別器に基づくプロジェクト成否の予測実験を行い、比較評価の結果、統計的検定に基づく手法を利用することで精度の高い予測が可能になることを確認した。

2 成果の各論

(1) メトリクス、仮説、妥当性の検討

メトリクスと仮説の妥当性を検討するために、SECから提供された1,009件のプロジェクトデータに対してベイズ識別器が有効に機能するか否かを判断する。実際には一定の条件を満たす505件のデータを利用した。ベイズ識別器により、プロジェクト成否評価、生産性、プロジ

ェクト日数予実差、発生不具合数についての予測を行った。しかし、予測実験の結果はどのメトリクスについてもよい結果は得られなかった。この実験の結果より、以下のような問題点を導き出した。

1. 企業横断的に集められたデータの信頼性

1. 悪い結果を示すデータが非常に少ないデータの偏り

この結果、できるだけ一企業内のプロジェクトに絞ってデータを収集した方がよい結果を残せようだという結論に至った。

(2) 事例への適用研究

SEC研究員の協力の下、新たにデータの収集を行った。ここでは、専門家によるメトリクスの選択と統計的な手法によるメトリクスの選択の2手法がベイズ識別器による予測精度にどの程度の影響を及ぼすかを調べた。

まず、豊富な開発経験をもち、メトリクスに精通した複数の専門家がメトリクスの選択を行った。品質、コスト、工期の視点ごとに選択されたメトリクスを利用して、ベイズ識別器による予測実験を行った結果、品質の予測精度は0.708、コストと工期についてもそれぞれ予測精度は0.519、0.500と計算された。

次に、Wilcoxonの順位和検定に基づくメトリクスの選択を行った。品質の予測精度は0.875、コストと工期についても予測精度はそれぞれ0.704、0.769と計算された。この結果、品質、コスト、工期のすべての視点において、統計的検定によって選択されたメトリクスを用いた方が、予測精度が高いことがわかった。

本研究の内容は論文としてまとめ、2006年5月に上海で開催されるソフトウェア工学国際会議(ICSE '06)のFar East Experience Trackにて発表する予定である[ABE2006]。

参考文献

[MIZUNO2005]水野修, 安部誠也, 菊野亨: プロジェクト混乱予測システムのベイズ識別器を利用した開発, SEC journal, No.4, pp.24-35, November 2005
[ABE2006]Abe, S., Mizuno, O., Kikuno, T., Kikuchi, N., and Hirayama, M.: Estimation of Project Success Using Bayesian Classifier, In Proc. of International Conference on Software Engineering, Far East Experience Track, (to appear)



大阪大学(http://sel.ist.osaka-u.ac.jp/)との共同研究

先進ソフトウェア開発プロジェクト におけるコードクローン分析

大阪大学大学院情報科学研究科
教授

楠本 真二

大阪大学大学院情報科学研究科
博士後期課程3年

肥後 芳樹

大阪大学大学院情報科学研究科
博士前期課程1年

吉田 則裕

大阪大学大学院情報科学研究科
教授

井上 克郎

ソフトウェアの保守を困難にする要因の1つとして、コードクローンが注目されている。コードクローンとは、「ソースコード中に存在するコード片で、他のコード片と一致または類似しているもの」と定義される。コードクローンの存在はデバッグ作業や保守における機能変更において、しばしば障害となる。本稿では、先進ソフトウェア開発プロジェクトで開発されたソースコードに対して、コードクローン分析ツールGemini[HIGO2005]を用いた分析を行い、プロジェクトマネージャに対して分析結果のフィードバックを行った内容について簡単にまとめる。

1 分析の概要

分析対象のソースコードは5種類で、8,000行から35,000行程度の規模であった。主な分析としては、コードの重複度(ファイル全体のトークン数に対するいずれかのコードクローンに含まれるトークン数の割合)の算出、散布図上で目立ったコードクローン、メトリクス値として際だった値をもつコードクローンに対する詳細分析を実施した。図1に1つのソースコード群に対する散布

図を示す。散布図では、原点を左上隅として、水平・垂直方向にソースコード中のトークンを出現順に配置し、水平方向のトークンと垂直方向のトークンが等しい箇所に点をプロットする。散布図中ではクローンの対は点の連続した線分として出現する。図1では、A~Gの部分抽出した。一方、メトリクス値を用いた分析では、あるコードクローンのソースコード断片のディレクトリ階層内での広がり、コードクローンを共通ルーチンとして書き換えた場合のソースコードの減少量の予測値等の値に注目し、分析対象ソースコード群において大きな値を持つコードクローンを抽出した。

抽出したコードクローンに対して、ソースコードで内容を確認し、特徴及び集約すべきクローンであるかどうかを調査した。主な結果は、以下の通りである。

コードの重複度は30% ~ 60%程度。
ある特定の処理(ログ収集、日付操作等)の簡単なコードクローンが多く検出された。また、扱う情報は違うがロジックが同じであるコードクローンも多く検出された。
コピーにより生成し、変数名、関数名等の変更漏れと思われるものも検出された。

2 今後の方向

プロジェクトマネジメントの観点からは、品質やコストに対して影響を与えるコードクローンの識別が必要となる。品質やコストを含めたクローン分析を実施することを今後の課題と考えている。

参考文献

[HIGO2005]肥後他: コードクローン分析ツールGeminiを用いたコードクローン分析手法, 信学技, SS2005-30, 105, 228, pp. 37-42, 2005

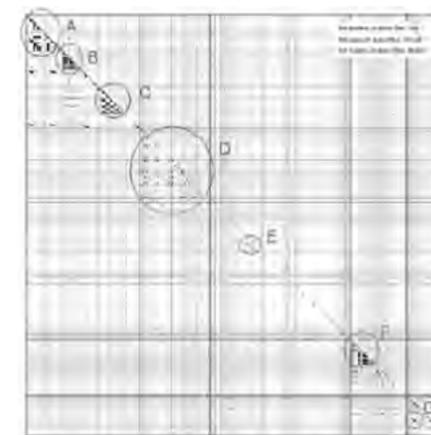


図1 散布図の例



慶應義塾大学(<http://www.am.ics.keio.ac.jp/top/index-j.shtml>)との共同研究

組み込みアーキテクチャとそのプログラミングについての調査

慶應義塾大学理工学部 教授
天野 英晴

組み込みシステムの応用分野は近年飛躍的に拡大し、目的に応じた処理を従来とは桁違いの高性能で実行することが要求されるようになった。このため、単純なプロセッサとメモリで構成された組み込みプロセッサ分野に、コンフィギャラブルプロセッサ、マルチプロセッサ、タイルプロセッサ、リコンフィギャラブルシステム、動的リコンフィギャラブルシステム等、新しいアーキテクチャが次々に登場した。しかし、実行すべきプログラムは、相変わらず従来の単純なシステムで主としてメモリ効率を上げることを目的に設計されており、効率の良い移植が困難である。そこで、新しいアーキテクチャを分類、整理し、現状のプログラミング方式をまとめ、移植が容易な新しいプログラミング作法を探った。

図1に示すように新しいアーキテクチャを整理、分類し、それらのプログラミング方式をまとめたサーベイを行った。これに基づき、新しいプログラミング作法としてストリームモデルに基づく書き方を提案した。さらに、組み込みベンチマークMiBenchの中から選んだいくつかのプログラムをこの手法で書き直して並列化を行い評価した。

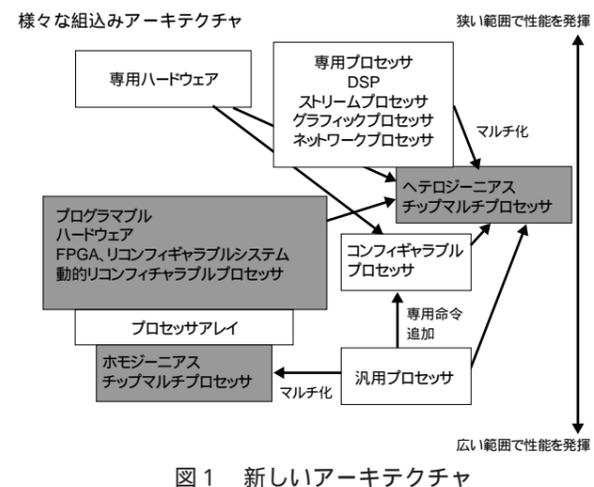


図1 新しいアーキテクチャ

1 成果 - ストリームモデル

ストリームモデルは、Dallyにより2003年に提案された。この方法では、まず、プログラムの各段階をタスクとして定義する。そして、ストリームデータをグローバルな配列データとして定義し、入出力を明確に記述する。それぞれのタスクでは、入力されたストリームデータに対して書き込みを行うことが可能だが、出力データは別の領域を確保して書き込む必要がある。この際、ストリームのサイズを明確にすることが望ましい。ストリームモデルで記述したJPEGコードのモデルを図2に示す。それぞれのタスクに相当するサブルーチン間のデータとその量のやりとりが明確化されている。このような形で記述を行うことで、並列化、ハードウェア化が容易になる。

2 今後の方向

ストリームモデルは、メディア処理の多くに適合するが、さらに広い範囲に適用可能かどうかをより詳しく調査し、プログラミング作法を定める必要がある。また、並列化MiBench、ハードウェア記述用MiBench等を公開し、提案した作法を広めて行きたい。



図2 ストリームモデル



国立情報学研究所(<http://www.nii.ac.jp/>)との共同研究

ETSS向け教育プログラム評価手続きの策定

国立情報学研究所
アーキテクチャ科学研究系 教授・研究主幹
本位田 真一

国立情報学研究所/
株式会社三菱総合研究所
糸野 文洋

本共同研究は、ETSSに準拠した教育プログラムに対する評価方法を策定し、組み込み分野における教育プログラムの作成及びその保守・運用の支援に役立てることを目的としている。

1 背景

国立情報学研究所では、ソフトウェア工学分野の研究成果である最先端のツール類を開発の現場で活用できるスーパーアーキテクトを育成するためのプロジェクト(トップエスイープロジェクト)を2004年から実施している。本プロジェクトでは、産官学による連携のもと、15件の教材開発を行い、講座を開講している。この特徴の1つに、テーマとして「情報家電」を取り上げている点がある。組み込みシステムとしての家電機器がネットワ

ークを介してサーバと連携する情報家電を題材とすることにより、組み込み系とエンタプライズ系の両知識の習得を狙っている。カリキュラムの構成も今後想定される情報家電の課題に基づいたものとなっている(表1)。

本共同研究では、本プロジェクトでの教材開発及び講座で得られた知見と日本技術者教育認定機構(JABEE: Japan Accreditation Board for Engineering Education)等の教育プログラム評価認定制度の動向調査に基づき、ETSS向け教育プログラムの評価フレームを策定する。さらにトップエスイープロジェクトで開発した教材・カリキュラムへの評価試行を実施し、評価フレーム自体の改善を進め、評価フレームの公開・普及を進めていく予定である。本共同研究は、図1に挙げたスケジュールと実施体制で進めている。

2 成果と今後の方向

国内においては、JABEEが技術者育成向け教育プログラムの審査・認定を実施しており、2004年度まで認定を受けた教育プログラムは、情報系を含め、186件にのぼっている。本研究においてもJABEEによる評価フレームをベースとする方針であるが、その審査対象が学部教育であるため、ETSSのスキルレベル(1以上)を対象とした教育カリキュラムへの評価フレームに利用できるかどうかは、今後の検討が必要である。JABEEにおいても産

業界からの強い要請により、大学院評価認定の策定を開始している。こうした動向や2006年5月に公開予定のETSSの次期バージョンを踏まえ、評価方法の策定を検討する予定である。

表1 カリキュラム

工程	課題	セキュリティ	信頼性・効率性	変更容易性
要求分析		要求分析		
	安全要求分析		形式仕様記述(基礎編)	コボ-ネットへの懸
設計			形式仕様記述(応用編)	
	形式仕様記述(セキュリティ編)		設計モデル検証(基礎編)	アジャイル開発
			設計モデル検証(応用編)	アズペ外指向開発
			並行システムのモデル化と検証	ソフトウェアバグ
			性能モデル検証	
実装		テスト・プログラム解析		
		実装モデル検証		

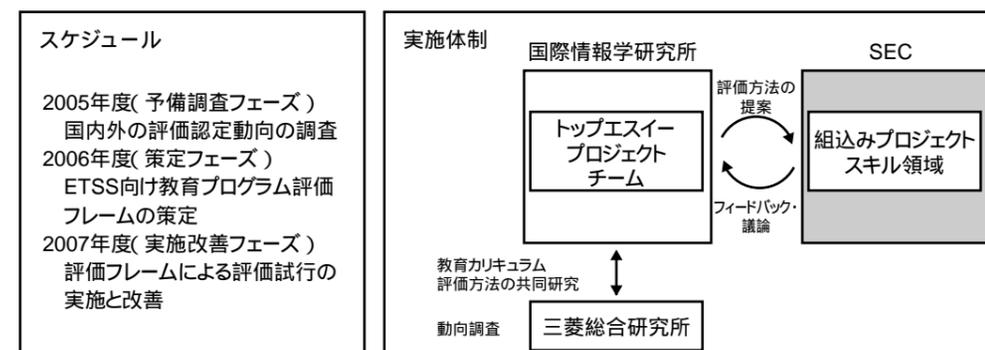


図1 本プロジェクトのスケジュールと実施体制



東京大学COEものづくり経営研究センター(<http://www.ut-mmrc.jp/>)との共同研究

デジタル製品向け組込みソフトウェア

- 開発力と競争力について -

東京大学COEものづくり経営研究センター (MMRC)
特任助手

立本 博文

日本の製造業は、組込みソフトウェアと無縁ではいられない。1970年前半には、早くも、自動車のエンジン制御や白物家電のシーケンス制御に利用され始め、80～90年代はVTR、CDのデジタルサーボ制御、90年代後半からはデジタル家電が急速に普及した。

このような背景の中で、MMRCではデジタル製品の中に含まれる組込みソフトウェアに注目して、その製品開発プロセスと最終的な競争力との関係パターンを、SECと共同して調査することとした。なお、デジタル製品では、ソフトウェア開発が重要なプロセスであるとともに、クリティカルパスになりつつことも注目した理由である。

1 経緯

調査の枠組みとして、1980年代に行った自動車産業の日米欧の国際製品開発力比較に用いた概念を参考に、「表の競争力(市場での製品競争力)」、「裏の競争力(製品開発力)」とに分けて測定・分析をし、最後に両者の関係について考察することとした。

組込みシステム製品は広範囲に及ぶため、今回の調査の対象をデジタル家電(DVDレコーダ等)に絞った。

研究の前半では、「表の競争力」を調べるために、デジタル家電(4製品)の市場価格と製品の属性との間の関係をヘドニック法により測定した。ヘドニック法は、製品の店頭寿命が短い家電製品等に対して、製品の価値を製品が持つ属性(機能)の集合とみなし、1つ1つの属性の寄与率を推定し、各属性から製品の理論価格を測定する方法である。卸売物価指数等に適用される方法である。さらに、今回の分析では、その結果から説明力の高いと思われる製品属性をハードウェア(HW)要因、ソフトウェア(SW)要因、その他に分けて考察を行った。その結果、HW要因が製品価格の6割程度、SW要因が3割程度を説明するケースが多かった。一方、企業ごとのバラツキをみると、HW要因では、ほとんどバラツキが無い一方、SW要因と考えられる属性には、企業ごとのバラツ

キが大きく製品の競争力の差になっていることが推測された。

研究の後半では、「裏の競争力」の測定を行った。製品開発における組込みソフトウェアの開発プロセスのプロセス管理の仕方(メカニズム)や、そのパフォーマンスがどのような関係にあるのか把握することが目的である。

その際には、組込みソフトウェア開発プロセスは、製品開発プロセスの1サブシステムであるので、システム全体(製品開発全体)から測定する視点とサブシステム(組込みソフトウェア開発)の視点の両者を設問項目に取り入れた。メカニズムとしては、前後工程の調整のあり方、部門間調整のあり方、社内共通部品の持ち方等を取り上げた。パフォーマンスとしては、各工程のリードタイム、投入人数、不具合発生件数等を調査項目とした。

この結果作成されたアンケート項目を「2006年版組込みソフトウェア産業実態調査」の質問項目に追加し、現在集計・分析作業を行っている。

2 今後の方向

今後、他の製品分野(車載ソフト等)や、製造業を支える製造装置に含まれる組込みソフトウェアへの応用を図っていく計画である。

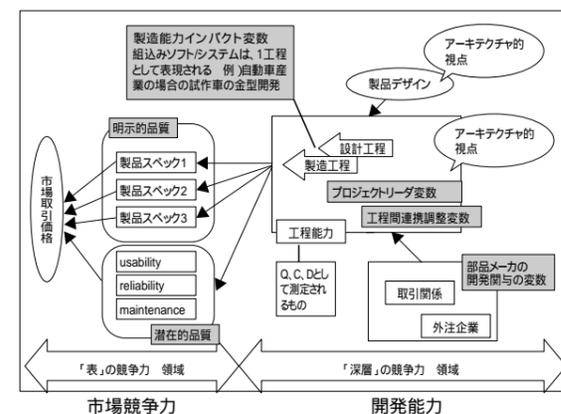


図1 組込みソフトの競争力とその要因を調査するためのフレームワーク



名古屋大学大学院情報科学研究科 附属組込みシステム研究センター NEXCESS
(<http://www.nexcess.itc.nagoya-u.ac.jp/>)との共同研究

組込みソフトウェア教育における体験型学習の調査

名古屋大学大学院情報科学研究科
附属組込みシステム研究センター NEXCESS

山本 雅基

社会人向けの組込みソフトウェア技術者教育を実施しているNEXCESSでは、人材育成の立場でSECとの共同研究を実施している。組込みソフトウェアなどの教育では、知識を教え込むだけではなくスキルを育成させることが要求されている。一般的に、スキルは体験させる演習(体験型学習と呼ぶ)を通じて育成され、座学の講義よりも多くの時間を必要とし教育投資が増加する。本研究では、教育の投資対効果を向上させることを目的に、組込みソフトウェア教育における体験型学習の実態調査等を通じ、組込みソフトウェア技術者のスキル習得に必要な効果的な体験型学習のあり方を求めた。

1 調査結果

(1) 社会人向け教育における体験型学習の時間比率

社内教育及び公開講座として実施されている社会人向け組込みソフトウェア教育の9社16科目について、サンプリングし、教育時間に関する調査を行った。また、実機ボードを使用するeLearningコンテンツ5科目を大学院生に受講させ、実際に要した時間を計測した。この結果、調査した社会人向け組込みソフトウェア教育では、体験

型学習に全教育時間の約50%の時間をかけていることがわかった。

(2) 体験型学習における受講者の行動分析

体験型学習としての実機演習では、プログラミング演習に比較的長い時間を要する。演習の進行状況を分析し、無駄なく演習が行われているか分析する。今回、プログラミング演習における受講者のキー打鍵回数を記録する実験を行った。NEXCESS受講者の了解を得た上で、キー操作を記録するアプリケーションを実習用のPCに常駐させた。簡易分析により、1時限の中でも受講者によりキー打鍵回数に大きな差があること(図1)、単位時間ごとのキー打鍵数を時系列で並べると、演習時間内で打鍵数が大きく変動していること等がわかった。

2 今後の方向

体験型学習時の受講生の振舞いの分析を深め、最も効果が期待できる演習問題の難度、粒度及び演習時間などを求める。将来的には教室内だけではなく、職場における指導の指針としても活用できる成果を目指す。

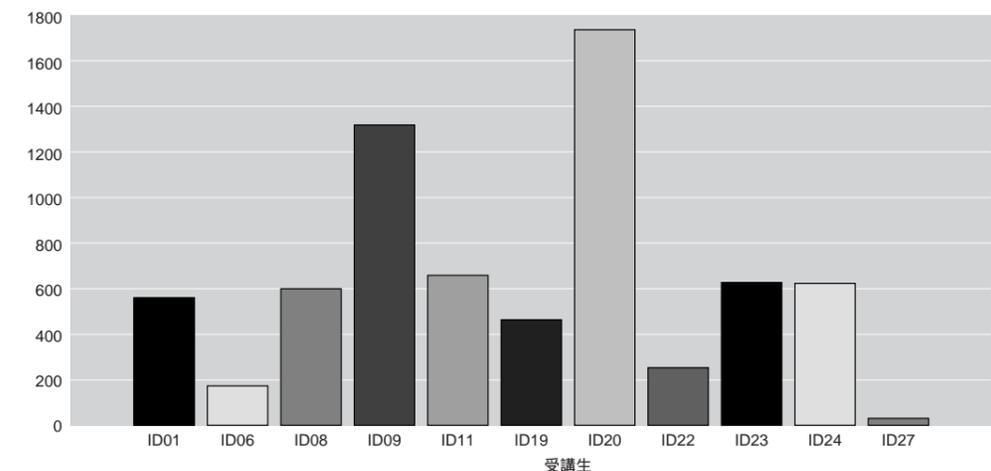


図1 1時限(90分)における受講者のキー操作数累計



奈良先端科学技術大学院大学 (http://se.naist.jp/) との共同研究

協調フィルタリングを活用した工数見積り

奈良先端科学技術大学院大学 情報科学研究科 教授 松本 健一
 奈良先端科学技術大学院大学 情報科学研究科 助教授 門田 暁人
 同志社大学文化情報学部 助教授 宿久 洋

ソフトウェア開発プロジェクトの実績データは、多くの場合、未記録の値(欠損値)を含んでいる。本研究では、欠損値を含むデータセットを入力として工数見積りを行うことが可能な「協調フィルタリングに基づく工数見積り手法(EASE協調フィルタリング法:EASE-CF法)」の実データへの適用と評価を行った。また、欠損値を含むデータセットに適したデータ間の依存関係の分析手法について検討し、分析の試行を行った。研究の遂行にあたっては、奈良先端科学技術大学院大学の杉直樹研究員、松村知子研究員、角田雅照氏、柿元健氏、SECの菊地奈穂美研究員に尽力いただいた。

本研究では、SECが15社から収集した1,009件のプロジェクト、約400種類の特性変数を含む実績データ(SECデータ)を用い、詳細設計終了時を想定し、その時点で収集が終了していると思われる97個の特性変数を用いて、開発総工数を見積もった。相互検証法によって見積り精度を評価した結果、EASE-CF法は他の手法よりも高い精度を示し、実績工数に対する見積り値の相対誤差の平均値が0.642、Pred25(相対誤差が0.25以下のプロジェクトの割合)が30.1%であった[OHSUGI2006]。また、データ間の依存関係の分析については、統計的検定や平行座標プロット(Parallel Coordinate Plot)を用いて、生産性に影響する要因、及び、要因間の関連の分析を行った[TSUNODA2006]。

1 成果

詳細は割愛するが、EASE-CF法の適用結果について述べる。詳細は文献[OHSUGI2006]を参照いただきたいが、工数見積り手法として、EASE-CF法、重回帰分析、対数重回帰分析、ニューラルネットワークを比較した。比較にあたっては、SECデータから開発総工数と開発規模(FP)が計測されている378件のプロジェクトのデータを抽出し、精度評価に用いた。

評価結果を箱ひげ図で表したものを図1に示す。図中

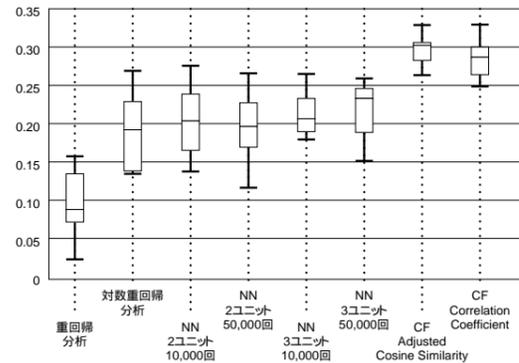


図1 Pred25の評価結果

では、EASE-CF法をCF、ニューラルネットワークをNNと略記した。図では、横軸に各手法を列挙し、縦軸がPred25の値を表す。図に示されるように、EASE-CF法、とくにAdjusted Cosine Similarityを類似度計算アルゴリズムとして用いた場合に、ばらつきが小さく、高い精度で見積りを行えた。この結果は、過去の類似のプロジェクトのデータに基づいて類推見積りを行うという協調フィルタリングのメカニズムが有効であったことに加え、実験データに含まれる欠損値が他の手法の精度を低下させたためと考えられる(他の手法では、欠損値を当該変数の平均値で埋める処理を行っている)。

2 今後の方向

EASE-CF法は、他の手法と比較すると高い精度で見積りが行えるものの、見積りが大きく外れるプロジェクトは依然として多数存在する。EASE-CF法は、過去の類似プロジェクトに基づいて見積りを行う手法であるから、類似のプロジェクトが少ない場合には見積りを行わないといった割り切りも可能であり、今後の検討課題である。

参考文献

[OHSUGI2006] 大杉直樹, 角田雅照, 門田暁人, 松村知子, 松本健一, 菊地奈穂美: 企業横断的収集データに基づくソフトウェア開発プロジェクトの工数見積り, SEC journal, No.5, pp.16-25, 2006
 [TSUNODA2006] 角田雅照, 門田暁人, 宿久洋, 菊地奈穂美, 松本健一: 外部委託率に着目したソフトウェアプロジェクトの生産性分析, 電子情報通信学会技術報告, ソフトウェアサイエンス研究会, 2006



人間中心設計推進機構 (http://www.hcdnet.org/) との共同研究

組込みソフトウェアにおけるユーザビリティ

特定非営利活動法人 人間中心設計推進機構 理事長・事務局長 鱗原 晴彦
 小樽商科大学社会情報学科 助教授 平沢 尚毅
 東海大学情報理工学部 経営システム工学科 助教授 辛島 光彦
 特定非営利活動法人 人間中心設計推進機構 稲垣 和芳

組込みソフトウェアの場合、バグにより作動しなくても、操作ミスや使い方がわからなくて作動しなくても、ユーザにとっては要求が達成されないという点で、同じ不具合ということになる。こうした利用品質の低下を回避するためには、システム要求分析の段階でユーザビリティ(Human centered design)を導入するとよいことが知られており、組込みソフトウェアの開発現場へ適用可能な「ユーザビリティ技術」の手法・技法を確立するとともに、そのスキル定義とキャリア(職種)定義を目指すことが急務となっている。SECの目的である「高品質なソフトウェア開発」を実現するにあたり、第一段階としての本調査では、アンケート調査及びインタビュー調査により、組込みソフトウェアにおけるユーザビリティ活動の現状分析を行い、その取り組み実態を明らかにした。

アンケート調査は、2005年11月16日~18日にパシフィック横浜で開催されたEmbedded Technology 2005(ET2005)において、エンジニア約600名に対して行った。インタビュー調査は、先行企業2社の組込み系システムのソフトウェア設計のプロジェクトマネージャ及びプロジェクトマネージャ統括責任者に対して行った。

1 成果

ET2005のアンケートでは、6項目以上を行ったが、回答者の98%が、ユーザビリティの重要性を認識していた。しかし、図1に示すように、実際には、ユーザビリティ

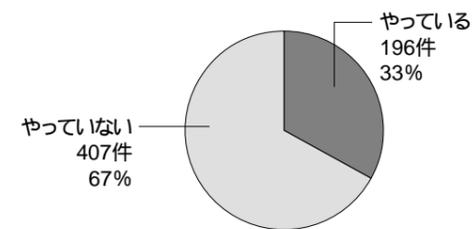


図1 「ユーザビリティ(利用品質)の課題に関する改善活動をやっていますか?」アンケート結果

の改善活動を行っている回答者は、33%と少ない。

もちろん、ここでいう改善活動は、高次のユーザビリティから単なるクレーム処理に至るまで幅広いはずなので、SECが目指す利用品質の向上に値する改善活動はもっと少ないものと推測することができる。こうした情報からユーザビリティが知識として紹介されていても、開発に生かせる知見情報の伝達不足や、実開発には盛り込めない何らかの障壁があることも容易に予測できる。

アンケート調査及びインタビュー調査の結果、明らかになったユーザビリティ及び人間中心設計プロセスへの理解と取り組みの現況は次のとおりである。

各エンジニアやプロジェクトマネージャは、ユーザビリティというタームや、その必要性及び製品価値の向上への寄与は認識しつつある。一方、プロジェクトとして人間中心設計プロセスの現状の設計プロセスへの組み入れについては、一部を除き消極的である。その理由として、設計プロセスの改革は企業にとって大きな問題であり、経営層の判断が求められるが、費用面での効果が不明瞭であることが考えられる。これが今後の人間中心設計プロセスを現状のソフトウェア設計プロセスに融合させていくための課題として浮き彫りとなった。

2 今後の方向

今後は、この結果を踏まえ、利用状況情報をもとにした「ユーザビリティ改善提案活動」を継続的に推進するための必要十分な課題を抽出し、利用品質の向上を目指すための指針を提言していきたい。



北陸先端科学技術大学院大学(<http://kt-www.jaist.ac.jp/index-j.html>)との共同研究

形式的手法の実用化を目指して

北陸先端科学技術大学院大学
情報科学研究科 特任教授
岸 知二

組込みソフトウェアの信頼性が大きな関心事となっており、形式的手法のような科学的手法の実用化に期待が高まっている。本共同研究は、形式的手法の実用化のために必要な技術の評価、手法の洗練、さらに産業界への認知拡大を加速することを目的としている。共同研究は、SECの研究者と北陸先端大学側とが定期的に打ち合わせを行いながら進めた。

今年度は形式検証技術に焦点をあて、以下の観点からの検討を進めた。

形式検証技術の利用方法の検討：形式検証技術の適用事例の調査等を踏まえ、形式検証技術の有効な活用方法を議論・検討し、既存の検証技術との使い分け等を整理した。

例題に基づく検証技術の評価：形式検証技術の有効性

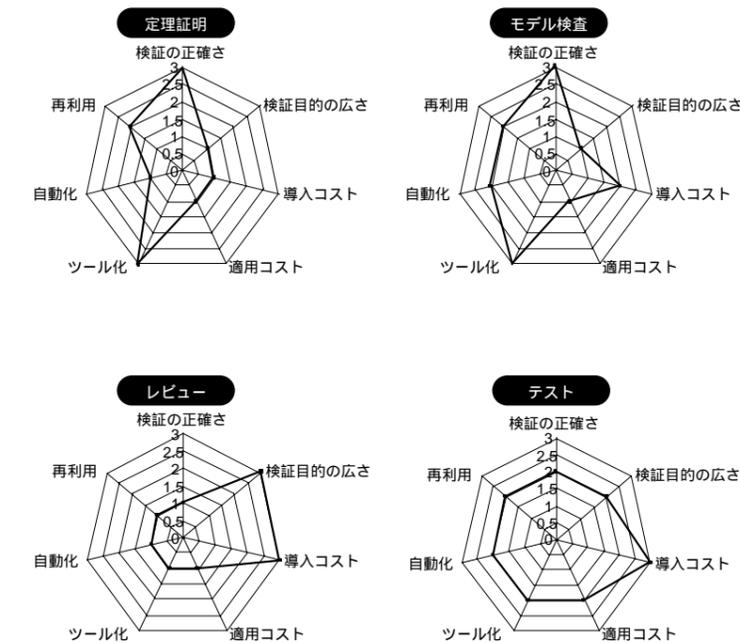


図1 各種検証技術の特性（外側程優れていることを表す）

を実証的に確認するために、CDプレーヤのサンプル設計に対して検証の試行を行い、状態遷移設計の検証等への有効性を確認した。

1 成果

ソフトウェアの検証には、従来からレビューやテスト等の技術が使われてきた。図は形式検証技術であるモデル検査技術と定理証明技術及び既存技術との特性を定性的に示したものである。形式検証技術は検証の正確性等には優れるが、一般にピンポイントの検証となり、また適用コストも高い。形式検証技術を活用するためには、それぞれの検証技術の特性を理解し、適切な使い分け、使いこなす必要があり、本研究では、典型的な活用パターンの検討をすすめた。

一方、CDプレーヤの状態遷移図のサンプルに対して、モデル検査技術を適用した。ここでは、状態遷移図において一般的に検証すべき性質と、CDプレーヤというアプリケーション特定の観点から検証すべきものに分けて、検証する性質を抽出して検証を行った。その結果、いくつかの誤りを検証し、有効性を確認できた。一方で、状態遷移図からモデル検査を適用可能な記述への変換、検証する性質の抽出法等に関しては、工夫が必要であることも判明した。

2 今後の方向

形式的手法は有効な技術であるが、既存の検証技術とは違った特性を持っているため、実際に適用するには活用方法の検討と、ノウハウの蓄積が必要である。今後、よりリアルな事例での適用を行い、モデルケースとして提示するとともに、適用技術の洗練を行っていききたい。



IESE (<http://www.iese.fhg.de/>)との共同研究

見積り手法の実証評価 - CoBRA法とOSR法 -

SEC 研究者
石谷 靖

大学・研究機関等との共同研究の一環として、2004年11月にドイツフラウンホーファ協会IESE¹とは包括的共同研究契約を結び、具体的なテーマとして先進的な見積り手法（CoBRA法とOSR法）の日本国内での適用可能性を取り上げ、約1年間共同研究を行った。ここでは2005年度の成果を中心に、共同研究のその後の成果を紹介する。

なお、IESEとの共同研究は、SECにおける実証・普及の中の1つの項目として位置づけられる。見積り手法が最初に取り上げたのは、ニーズが高いことが最も重要な要素であるが、さらに具体的な手法があること、また、定量的な評価・実証が比較的容易であることがある。

また、CoBRA法の特徴は、経験豊富なプロジェクトマネージャの知識を形式化することにある。経験豊富な人間の直感が正確であるとの経験則に基づいている。OSR法の特徴は、大量のプロジェクトデータからデータマイニング手法を用いて見積りモデルを作成するものである。

1 成果

今回の共同研究は、大きく2つのフェーズに分かれる。2004年度では、国内企業で手法を適用して見積りモデル

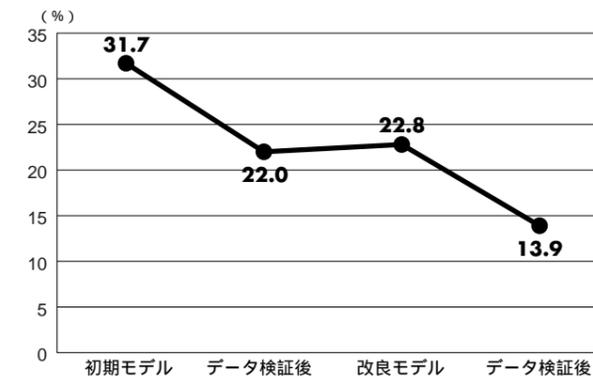


図1 平均見積り誤差

を作成し、2005年度で、そのモデルをリファインした。結論からいうと、2005年度CoBRA法におけるモデルのリファインは成功を収め、図1に示すように見積り精度の向上を実現することができた。具体的には、最初のモデルでは平均見積り誤差30%強であったが、要因モデルの見直しや過去プロジェクトデータの検証等を行った結果、同じ評価方法で、平均誤差14%を実現した。この試行を通してCoBRA法のモデルの改善への強みを実感し、導入方法の知見を得ることができた。

同様に、OSR法も2004年度の試行に対してリファインを行い、大量のデータから見積りのベースとなるプロジェクトデータを集約する方法としての有効性とデータマイニング手法共通の課題を確認することができた。

以上の成果は、2006年2月24日及び27日にIESEと共同で開催したコスト見積りワークショップ²において、IESE、試行企業及びSECから発表している。また、とくにCoBRA法については、『ソフトウェア開発見積りガイドブック』(2006年4月刊行)の中で取り上げ、解説している。SECでは今回の成果を背景に、現場でのエンジニアリングの実践を促進する1つの手段として、CoBRA法を中心に見積りモデル作成の実証を進め、知見を蓄え、さらに普及を図っていく予定である。

2 今後の方向

IESEとは、次の実証・普及のテーマとして、プロセス改善等の活動の導入とその評価フレーム等について共同研究することを検討している。これは、見積り手法に続いて、実践を促進する手段を提供するために、ニーズの高さとエンジニアリングとしての定量的な評価実証の必要性から選択したものである。

1 Institute für Experimentelles Software Engineering : 実験的ソフトウェア工学研究所。 <http://www.iese.fhg.de/>

2 1st International Workshop on Efficient Cost Estimation Approach (WESoC 2006)

新潟県での情報産業振興の取り組み 「N-SECプロジェクト」 ～IT産業クラスター形成を目指して～

http://www.nico.or.jp/

財団法人にいがた産業創造機構
産業創造グループ情報戦略チーム チーフ

高畑 悦武

財団法人にいがた産業創造機構（NICO）は、新潟県内産業の活性化を目的として平成15年4月財団法人新潟県中小企業振興公社を改組し活動を開始した。

NICOの標榜する“産業創造”を象徴するプロジェクトとして、「IT産業クラスター形成促進プロジェクト」は、県内情報関連産業、教育機関等との連携により「高度IT人材育成研修事業」を中心に活動してきた。

本稿では、4年目を迎えたNICO「IT産業クラスター形成促進プロジェクト」のこれまでの軌跡と、新たに活動を開始したN-SECプロジェクトを紹介したい。

1 県内ソフトウェア産業の状況

経済産業省編「平成16年特定サービス産業実態調査（速報）」によれば、新潟県内には132事業所に4,271人が従事しており、年間売上高は618億円となっている。これらの数値は、いずれも対前年比で5%以上下回っている状況となっている。また、平成16年7月に新潟県IT産業ネットワーク21が実施した「平成16年度新潟県ITサービス企業実態調査」によれば、県内ITサービス産業は資本金5千万円未満の企業が64.4%、年間売上高5億円未満が67.8%、従業員数50人未満が71.3%となっており、中小規模の企業が圧倒的に多いことがうかがえる。主要顧客は「情報サービス業」が55.2%と最も多く、さらに「コンピュータメカ」も20.7%であり、全国の業界実態と同様に業界の二次請け構造が反映されている。

2 新潟県の情報産業振興施策

新潟県は、洋食器・ハウスウェア、利器工器具、建築用金物等の金属加工製品や絹織物、ニット等の繊維製品等の地場産業が発達し、近年は工作機械、電気機器、電子部品等地域に蓄積された加工技術を生かした産業が集積し、県内経済を牽引してきた。しかし、バブル経済崩

壊による国内経済の低迷と海外製品の流入、製造業の海外移転等により、県内産業は大きな打撃を被ることとなった。これに対し、新潟県は「付加価値向上による地場産業の再生」、「新産業の創出による雇用の確保」、「技術革新に対応した人材の育成」を産業振興の方針として打ち出し、産業振興関連の事業を検討することになった。

そこで、情報通信技術を活用した産業群の形成を図るため、平成12年度後半より県内外の有識者を委員として招聘し、政策の検討を重ね、平成13年度に「にいがたe-Port構想」を策定し、県内の情報関連産業振興の方針を定め、具体的な施策体系の構築に着手し、平成14年度にはIT関連産業の育成振興と地域産業のIT活用促進を目的とした「IT対応促進総合支援事業」を実施し、IT産業クラスター形成促進事業を開始した。

3 財団法人にいがた産業創造機構（NICO）

平成15年度に新潟県における産業振興の中核的支援機関として「財団法人にいがた産業創造機構（NICO：Niigata Industrial Creation Organization）」が設立された。

NICOは、新潟県内の産業活性化を推進するために、新潟県の外郭団体である財団法人新潟県中小企業振興公社を母体として、新潟県が実施してきたベンチャー関連支援事業、技術・製品開発支援補助金、製造技術の高度化

に関する研究事業等を付加し、事業の企画立案、技術・サービス等競争力の強化、販路・取引の拡大、設備増強等企業活動を一貫して支援できる事業体系を整備した。また、公社職員に加え、県庁から職員を派遣するとともに、民間企業、大学等から人材を登用し、産学官が一体となり事業を推進する体制を構築した。

情報関連産業への支援は、株式会社新潟ソフトウェアセンターの役割を引き継ぎ、情報化人材を育成する新事業支援機関として、高度IT人材育成に必要な研修設備を整備した。県内IT関連産業振興の中核的拠点としての機能を強化するとともに、NICO設立当初より、ITを活用した産業群形成を目指して「IT産業クラスター形成促進プロジェクト」を開始し、クラスター形成の基盤となる高度なITスキルを有する人材を育成するため「高度IT人材育成研修」を実施してきた。また、県内情報関連企業の業界団体である「新潟県IT産業ネットワーク21（以下IT21）」及び県内ITコーディネーターで組織する「ITC新潟」の事務局業務を担い、さらには平成17年度からは県内自治体、情報関連産業、大学等教育機関で組織する「新潟県IT&ITS推進協議会」の地域情報化委員会事務局業務を新潟県総務管理部情報政策課とともに担っており、情報関連分野における産学官連携の名実ともに要的存在

新潟県紹介

新潟県は、日本海側のほぼ中央に位置し、面積はおよそ12,582km²（東京都の約6倍）で全国第5位の広さを持ち、人口は約243万人である。信濃川、阿賀野川等、豊富な水量を誇る大河がもたらす肥沃な平野が広がり、県境の急峻な山脈と長い日本海の海岸線に囲まれた県土は、四季折々の豊かな自然や伝統文化に恵まれ、多くの産業を育んできた。

そして新潟は、北前航路や多くの街道等、日本海側の海陸交通の要衝であり、首都圏との人的、物的交流も以前から盛んに行われてきた。昭和60年の上越新幹線の開業をはじめ、関越・北陸・盤越・上信越・日東道の5高速道路、海外8路線を持つ新潟空港、中核国際港湾である新潟港・直江津港等、充実した交通ネットワークが整備され、産業や文化の交流拠点として発展している。

新潟県は、恵まれた自然、風土を持ち、日本一おいしいコシヒカリをはじめ、逸品といえる野菜や果物等、様々な食素材を育んできた全国有数の食材の宝庫である。また、新潟県はモノづくりも盛んで、県内各地には、織物・ニット等の繊維製品、作業工具・金属洋食器・金属ハウスウェア等の金属製品、清酒・味噌等の食品、木工等、数多くの地場産業が展開している。

となっており、関連業界・専門家との連携強化及びその活動を支援してきた。

4 IT産業クラスター形成促進プロジェクト

平成15年度からNICO産業創造プロジェクトとして実施してきた「IT産業クラスター形成促進プロジェクト」は、新潟県内に情報通信技術（IT）を活用した産業群の形成を目指している。

NICOのIT産業クラスター形成促進プロジェクトでは、情報通信技術（IT）を地域産業の共通した基盤的技術としてとらえ、それを提供する企業（情報関連産業）とそれを活用する企業（地域産業）及びそれらを支援する機関（行政・大学・産業支援機関）の密接な連携体制の構築、すなわちITを基盤とした産業クラスターの形成を実現するために、まず地域内において需要が急速に高まりつつある高度な情報通信関連技術者を育成体制（高度IT人材育成体制）の構築に取り組んだ。

技術革新が急速に進展する情報通信技術においては、単に研修設備を整備し、研修を実施すればよいというものではなく、継続的な人材育成事業を行うためには、事業の目標を明確に定め採算性も考慮しながら、現在の技

さらに、豊富な石油・天然ガス等の利用を目指して発展してきた化学工業や一般機械器具製造業、近年発展してきた電気機械器具製造業等、産業市場を担う業種も多岐にわたり、情報通信技術（IT）やバイオテクノロジー、ロボット技術等、先端技術の研究開発と実用化が産学連携で取り組まれ、発展が期待される。



術の進展、産業界の動向等に対応した研修カリキュラムを提供できるかが課題となった。そこで、IT21を中心に、高度IT人材育成研修に関する説明とともに、社内における戦略的な人材育成の重要性についてITSSのフレームワークを用いた説明会を開催することにより、人材育成に対する気運醸成に努めた。

また、研修カリキュラムの内容については、業界代表者、教育機関、情報活用側企業の代表及びIPAのITスキル標準センターの方々を委員としたカリキュラム検討委員会を開催し、研修内容に関する評価及び要望等を研修事業へ反映した。

その結果、情報システムの上流を担う技術者養成を目指し、研修対象者も企業の中堅以上の技術者を対象とした研修カリキュラムにもかかわらず、平成15年度265名、平成16年度206名、平成17年度426名の受講実績があり、プロジェクトの1つの評価指標である4年間で800名の技術者育成という目標を3年間で達成できた。これは、県内情報関連産業との信頼関係及びIPAとの連携により可能となったものであり、本プロジェクトの初期の目的で

ある高度IT人材を育成する仕組みの構築に関しては、軌道に乗りつつあると思われる。

5 IT産業クラスター形成促進プロジェクトの新たな展開

IT産業クラスター形成促進プロジェクトを契機として、IT21が主体となり、県内5地域（新潟・長岡・上越・柏崎・十日町）における情報通信技術活用状況や業界の状況、市場分析等を行い、情報通信技術活用による地域活性化と業界活性化に向けたアクションプラン「新潟県ITサービス産業白書2004」（IT21白書）をまとめた。

6 N・SECプロジェクト

IT21白書の提案を受け、NICOでは平成17年度からIT産業クラスター形成促進を一層加速させるプロジェクトとして「N・SEC（NICOソフトウェア・エンジニアリング・センター）プロジェクト」を開始した。N・SECプロジェクトは、座学研修主体の「高度IT人材育成研修」では獲

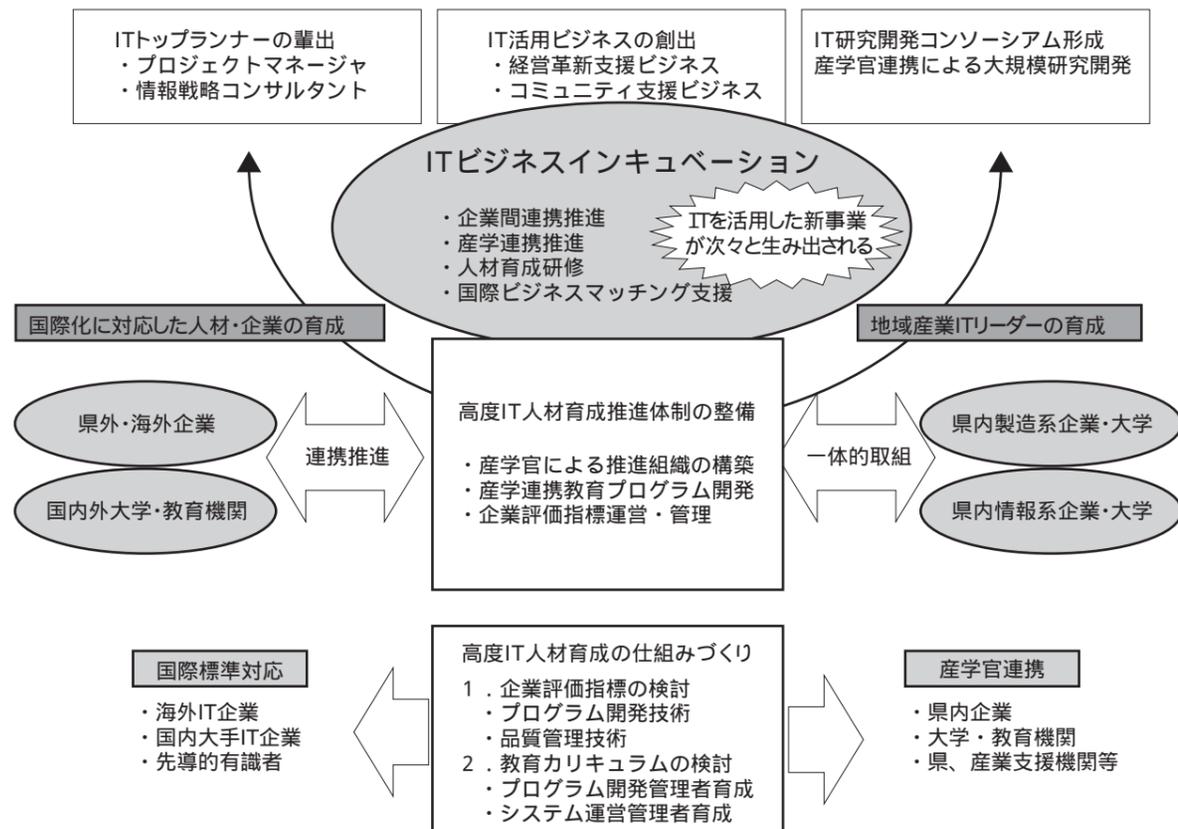


図1 IT産業クラスター形成促進プロジェクト

得が難しい「実践できる能力」を習得するための研修手法について検討を行ってきた。具体的には、IPA / SECで開発・検討されているソフトウェア工学に基づいた品質・生産性向上に関する研究成果を県内の企業・大学等と連携し、実証プロジェクトを実施するというものであり、エンタプライズ系では「OSSシステム適用研究会」を開催し、組込み系では「組込みシステム研究会」を開催し、平成18年度以降の具体的な活動を検討した。

7 今後の展開

NICO IT産業クラスター形成促進プロジェクトは、平成18年度で4年目を迎える。

N・SECプロジェクトへ事業が拡大したことにより、新潟県IT産業ネットワーク21、社団法人新潟県電子機械工業会から「組込みシステム開発力強化」等に関する連携事業実施等の要望が高まったため、平成18年度は、これら業界団体、NICOそしてIPAとの間で、情報関連人材育成及び産学官連携による技術開発等に関する包括的な協定の締結も視野に入れながら、一層の連携強化を図って

いく予定である。

今回ご紹介したN・SECプロジェクトの他に、県内産業のIT活用による経営革新を支援する「新潟IT経営応援隊事業」を県内のITCと協力しながら実施し、地域の情報化推進や個別企業の情報活用技術の向上にも取り組んでいる。しかし、組込みソフトに見られるように、情報通信技術の利活用領域が広範化、細分化され、技術も一層高度化が進展するとともに、企業の活動も広域化、国際化していることから、NICOも海外を含めた県外他の地域・企業・大学等と積極的にネットワークを広げ、案件によっては県域を越えた産業支援体制の構築にも取り組んでいきたい。

参考文献

- 新潟県IT産業ネットワーク21, 新潟県ITサービス産業白書2004, 2004 財団法人にいがた産業創造機構 <http://www.nico.or.jp/>
- 新潟県HP <http://www.pref.niigata.jp/>
- 高度IT人材育成支援サイト <http://eduport.nico.or.jp/>
- 新潟県IT産業ネットワーク21 <http://www.nico.or.jp/it21>
- 社団法人新潟県電子機械工業会 <http://www.neia.or.jp/>
- 新潟県IT&ITS推進協議会 <http://www.n-it-its.jp/>

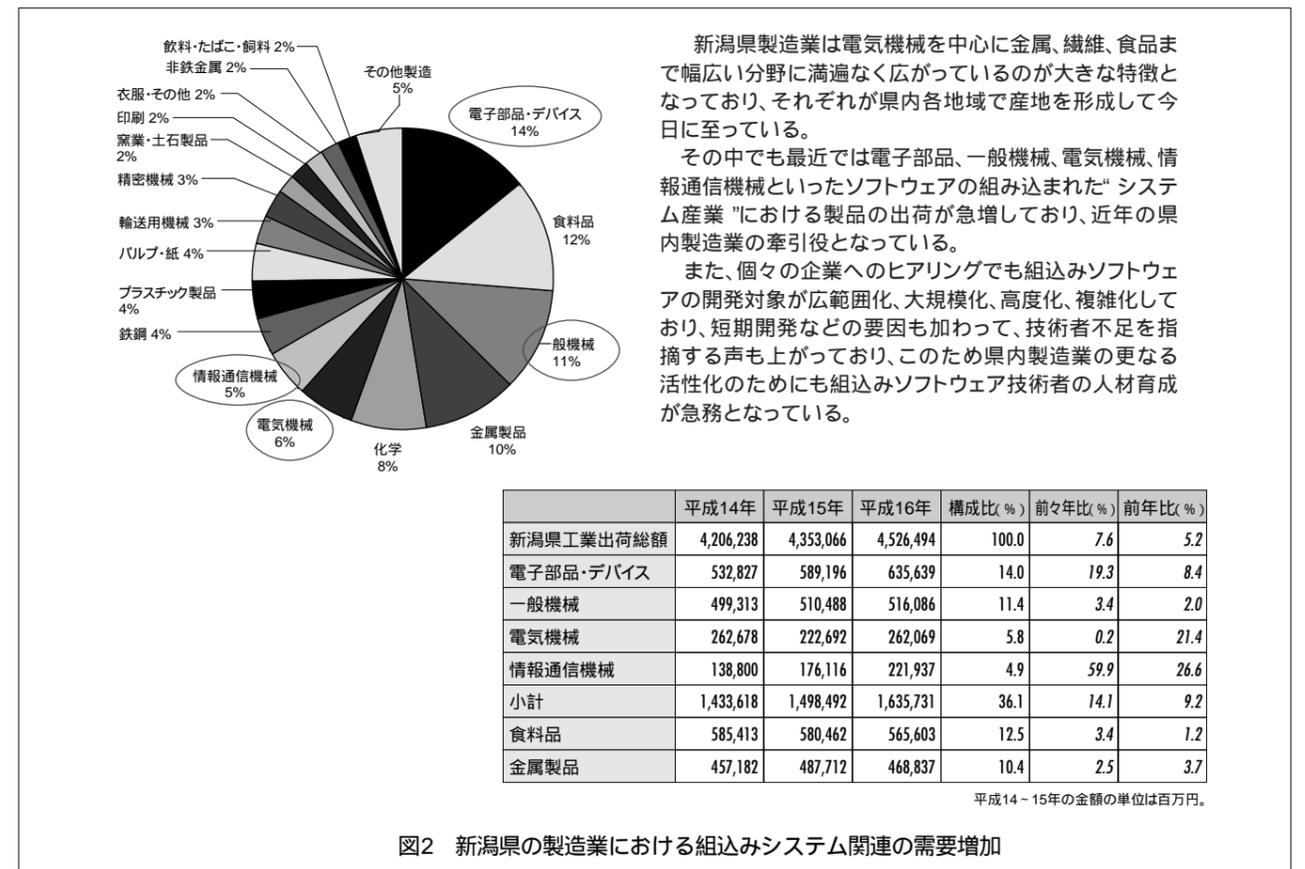


図2 新潟県の製造業における組込みシステム関連の需要増加

社団法人 組込みシステム技術協会 JASA

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

社団法人 組込みシステム技術協会 理事・事務局長

橋 重眞

2006年4月、協会設立20年目で名称を「組込みシステム技術協会」(JASA)に変更した。インテル社が世に問うた「マイクロコンピュータ」は、20世紀最大の技術革新ともいわれるが、周知のとおり、組込み技術の源泉はここにある。当時の若手ベンチャー経営者の種村良平氏、三田輝氏、真田隆幸氏らは、マイコン応用システム開発系団体設立の動きかけを通産省(当時)に対して行った。また、東海大学の原茂之教授は、今日の組込みシステムの繁栄を予測し、「マイコン応用システムに組み込まれるソフトウェアの重要性、人材育成の大切さ」を力説していた。「組込みシステム」は、車や情報家電、携帯電話等で一躍脚光を浴び出した。課題が山積する中で、JASA20年の蓄積とIPA/SECとの新たな連携による創造的事業で新生JASAを目指したい。

1 JASA 20年目の革新

1986年8月7日に通産大臣の社団法人設立許可を得て、日本システムハウス協会として勇躍スタートを切った。全国に点在していたシステムハウスの6つの任意団体が一本化されて全国団体になり、設立20年後の2006年現在、正会員130社、賛助会員28社、総計158社余りとなった。当協会では、当面、倍増の300社を目標に会員獲得に注力しているところである。「JASA改革」の大きな目標として、「組込み分野の中心的存在になろう」があり、中堅大手企業の組込みシステム部門の「部門入会」を推し進めることにポイントを置き、会費規定等の見直しを図り、門戸を開くべく努力している。

もう1つのJASA改革は、会員参加を事業活動の基盤にすることである。団体としては当然のことではあるが、「言うは易し、行うは難し」の習いで、新しい、自発的な研究会、WGの立ち上げを全面的に支援することで、会員参加を活発化させようと考えている。

当協会には、管理運営本部、企画本部、事業本部の3本部があり、主な活動は以下のとおりである。

管理運営本部：本部事務局として経理、財務、会員管理等、すべての機能を掌握する。また、全国6支部の活動支援の役割を持っている。

企画本部：2006年度から技術本部に名称を変えてスタートの予定で、セミナー、研修、教材開発、JASA試験等の組込み技術者育成の全般にわたって事業を推進する。

事業本部：ET(組込み総合技術展)併設のETロボコン、本年度から関西地区で実施する「ET West」を総括する。

2 成果生む、組込み関連出版

組込み技術者育成は、JASA企画本部の重要な事業で、SECと連携して開催している「組込みソフトウェアフォーラム」を実施している。札幌、名古屋、大阪、福岡でのべ500名余りが参加している。参加者に行ったアンケートによると、地域での組込み情報の少なさが強く指摘されており、狭い日本でも大きな情報格差のあることがわかった。

初級組込み技術者研修は、実機付き研修と座学研修の2種類を展開しており好評である。2005度に福岡、東京で開催した実機付き研修は、いずれも定員オーバーになる等好評であった。教材には、ARMの開発ツールを使用し、一般に実施されているものより簡易な内容となっている。座学研修の教材は、当協会の技術者研修WGが作成したもので、「はこだて未来大学」で開催した研修会では、学生に高い評価を得ている。

少し専門性のあるものは、ETカンファレンスで好評を博したの中からテーマを選び「ETセミナー」として実施している。2005年度は、4回実施しており、これも好評であった。



(ET2005)

3 出版物等、教材開発が順調

当協会は、教材開発にも注力している。調査研究報告書をベースに発行された『エンベデッド技術』(JASA組込み技術者育成委員会編著、電波新聞社発行)は、専門書としては異例の1万5千部を突破し、今も売れ続けている。第2弾の『組込みソフト技術』初版6,000部を突破し、順調に売れている。大学、専門学校を中心に、全国隈なく購読者があるようで、組込み技術の普及啓発に一役買っていると思われる。協会が編著、発行は出版社という役割分担とリスク分散が効果を上げているようである。これらの書籍は、まさに「雪中送炭」で、組込み技術者が待ち望んでいた書籍となっている。当協会としては、底辺の組込み技術者拡大を狙い、今後も書籍に限らず、教材開発に力を入れたいところである。ちなみに、書籍第3弾は、イラスト等をふんだんに使った、入門者層を狙ったやさしい組込み技術本の出版を予定している。

4 ET2006(組込み総合技術展)等

2006年度は、5月に関西では初めてとなるETWestを開催する。規模等、将来に向けての期待が大きいが、申込者数が当初の目標をクリアし、順調なスタートを切った。

メインのET2006(組込み総合技術展)は、今年が20回目で、11月15日~17日パシフィコ横浜で開催する。名実ともに認められる世界規模でのイベントに成長したと自負している。

これにより、東西での組込み技術に関するイベントが開催されることになった。地盤沈下が叫ばれる関西において、ETWestが「ものづくり再生」の起爆剤になればと期待がふくらんでいる。また、併設のETロボコンについても、チャンピオンシップをET会場で行う予定である。

5 20周年の目玉事業

JASA組込み技術者試験の実現に向けて試験全体の名称は、「組込みシステム技術者試験」(Embedded Technology Engineer Certification・仮称)であり、IPA/SECが進めるETSSに準じた、組込み技術のスキル認定を行うことによって人材の育成を図ろうというのが目的である。

試験方法は、今のところCBT(Computer Based Test)を採用する予定である。国家試験との関係、各関連団体との連携等、課題も残されているが、新生JASAの事業として、大きく一歩を踏み出したといえるだろう。

6 SECとの連携

昨年度から全国で「組込みソフトウェアフォーラム」を開催している。ETSSの普及啓発事業の展開(試験)地域への組込みシステム普及啓発等、現場への適用作業が望まれている。

独立行政法人 情報処理推進機構 人材育成推進部

<http://www.ipa.go.jp/jinzai/risa/index.html>

独立行政法人 情報処理推進機構 人材育成推進部長

萬井 正俊

独立行政法人 情報処理推進機構 (IPA) は、「創造」、「安心」、「競争力」の3つの理念の下で事業を推進している。この3つのテーマは、要して言えば、「新技術の開発・普及」と「人材の発掘・育成」ともいえるだろう。本稿では、人材育成推進部の地域における人材育成の取り組み及びSECとの連携について、その概要を紹介する。

1 人材育成推進部の事業

地域で高度なIT人材の育成を推進

我が国における高度なSE、プログラマー等の人材不足を懸念する提言を受け、平成元年、「地域ソフトウェア供給力開発事業推進臨時措置法」(以下、「地域ソフト法」と略す。)が制定された。これに基づき、IPAに人材育成推進部が設立された。また、全国に20か所(現在は18か

所)の第三セクター方式による株式会社地域ソフトウェアセンターが設立され、各地で高度なIT人材の育成を開始した。その後、研修事業は「新事業創出促進法(平成11年2月)」に基づく情報関連人材育成事業として引き継がれ、これまでのシステム開発技術の研修に加えて情報の利活用も研修事業の対象となった。そして、「中小企業新事業活動創出促進法(平成17年4月)」¹に基づき、

IPAは情報関連人材育成事業を行う新事業支援機関に対して、教材の開発・提供、指導及び助言を行うものと規定された。このような法律改正を受けて、地域センターにおける今日の事業は、大きな広がりを持って運営されている。

ユーザ企業の人材育成 (ITの利活用の推進)

経済産業省は、ITの活用によって中小企業の経営課題を解決して産業競争力を向上することを目指し、平成11年より、「ITSSP事業」²を実施した。平成16年からは、「IT経営応援隊事業」³として人材育成推進部が経済産業省からの委託を受けて支援活動を開始し、現在は、経営者研修会、CIO研修会、IT経営百選、地域IT経営応援隊事業の4つのプログラムで全国の中小企業を支援している。



策の両面から組み産業の支援や組みエンジニアの育成に関する事業が展開されている。セミナーの開催から着手される事例が多く、セミナーの開催に際して講師の派遣依頼を受けている。

組みエンジニアの育成支援

情報関連人材育成支援機関においては、組みエンジニア育成に関するノウハウを十分に蓄積していないこともあるため、カリキュラムの開発指導や教材に関する情報提供等、組みエンジニア育成研修に関する支援要請にも応えている。

地域の高等教育機関と連携

産業界から大学等高等教育機関におけるITエンジニアの教育について実践スキルの習得を積極的に推進して欲しいという要望がある。組みの分野でも同様の事情であることから、地方自治体や地域の情報関連人材育成支援機関を通じて、地域の高等教育機関と連携を図っている。

2 情報関連人材育成支援機関全国会議

中小企業新事業活動創出促進法の施行を受け、平成17年11月に情報関連人材育成支援機関を対象とした全国会議を開催した。ここでは、地域での活用が期待されるIPAの施策を、情報セキュリティ対策、ソフトウェア産業支援、IT人材育成の3つに大括りして紹介した。

3 SEC組み系プロジェクトとの連携

人材育成推進部は地域の情報関連人材育成支援機関との窓口として機能してきたことから、IPAに対する様々な依頼を受ける。最近は組み分野に関するニーズが多く、SEC組み系プロジェクトメンバと連携を強化しながら、地域の各機関を支援している。

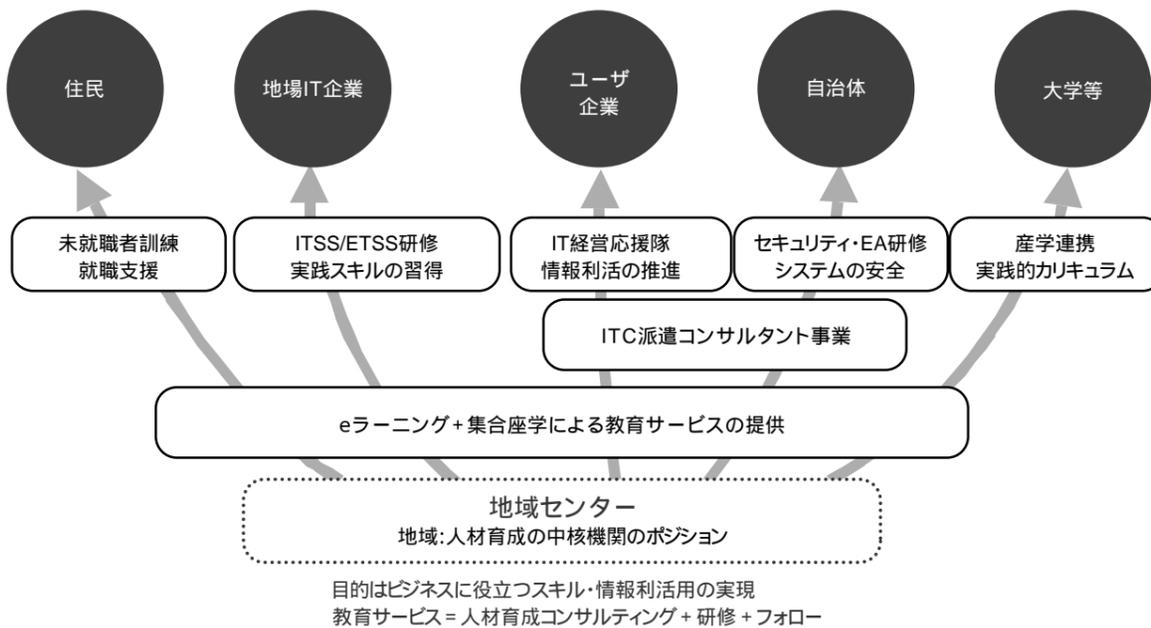
組み分野のセミナー講師の派遣依頼

一部の地方自治体においては、県内産業支援と雇用政

4 地域でのIT人材の育成等今後の事業展開

地域センターを含む情報関連人材育成支援機関との連携強化については、次のようなコンセプトにより事業展開することとしている。

- ・地域ニーズの変化への対応
- ・中堅・中小企業の社外CLO(Chief Learning Officer)の役割を負う
- ・実践スキルの向上を目的としてブレンDED型研修(知識(eラーニング)+実践スキル(ケース研修))へシフト
- ・研修コースはITスキル標準の考え方による等なかでも、研修事業についてはインフラ整備等大きな課題もあるが、有効な研修コースを開発し、実践スキルの習得等、地域貢献を果たしていきたいと考えている。



ITSS:Information Technology Skill Standards ITC:IT Coordinator
ETSS:Embedded Technology Skill Standards

図1 情報関連人材育成事業のイメージ

1 <http://www.chusho.meti.go.jp/keiei/shinpou/index.html>

2 ITSSP(ITソリューション・スクエア・プロジェクトの略称)。経営課題の解決事例がスクエア(街角の広場)のように集まり円滑に流通しあうといった意味と、個々の解決策がスクエア(二乗)され、さらに高度化されるとの意味を込めて名付けられた。

3 <http://www.itouentai.jp/>

BOOK REVIEW

Javaの10年

岩山知三郎 著

ISBN : 4-87566-322-6 コンピュータ・エージ社刊
A5判・236頁・定価2,415円(税込) 2005年11月刊



莫大な知的活動10年の集積を描く

携帯電話で電子マネーの残金を調べる度に、小さなコーヒーカップのマークを見るようになった。日常生活で見かけるこの小さなマークの背後にどんなドラマ、どんな努力の集積があったのか、文字通りJavaの10年を鳥瞰したユニークな歴史書である。わずか10年のめまぐるしい歴史であるが、その中に人間の知的努力の莫大な集積があった。多くのプレーヤーと著者自らが、空前絶後ともいえる、その歴史を共有した臨場感あるドキュメンタリーとして安心できる技術解説と合わせて描ききっている。

時間軸に沿って記述され、著者とともに10年間の時間飛行が出来るが、それは決して平穏な予知可能なものではなかった。その中には、無私の創造活動、そして野心

やビジネス的挑戦の挫折と成功、企業内政治、人間模様が網の目のように織り込まれている。世の中にインパクトのある真の「成果」とはこういうものだ、ということ語っているように思える。

このような一大ムーブメントの発祥が、米国シリコンバレーではなく日本だったら、どんなに素晴らしいだろうと思う。著者の前著『ビル・ジョイの冒険』と併せて多くの方にお勧めしたい。

(神谷芳樹)

ピープルウェア 第2版 ヤル気こそプロジェクト成功の鍵

Tom DeMarco・Timothy Lister 共著 松原友夫・山浦恒央 訳

ISBN : 4-8222-8110-8 日経BP社刊
A5判・310頁・定価2,310円(税込) 2001年11月刊



ソフトウェア開発は、人が命

数あるコンピュータに関する書籍の中でも、名著として知られるのが本書である。技術以外のスキルについて考える機会があり、約10年ぶりに手にとって見た。地元の図書館では、ビジネス書(人材管理)のコーナーに置かれていた。

本書(翻訳)のサブタイトルは「ヤル気こそプロジェクト成功の鍵」である。これまでの開発現場を振り返っても実感できる言葉である。しかし、ソフトウェアエンジニアリングの利活用が前提かもしれないが...

私のミッションである組込みスキル標準では、技術スキルのみを対象にスキル基準を定めた。しかし、実際に開発を成功させるには、ヤル気がベースになってくると思っている。キャリア基準では、技術スキルや知識を使って成果を出すには、パーソナルスキルとビジネススキルが必要で

当時は、第1版であった。

あると定義した。さらには、その根幹にコンピテンシやコアスキルが存在することを明らかにした。エンジニアリングだけに限定しても、技術スキルだけでは、成果が出ないと述べている。

本書の内容は盛りだくさんであり、中途半端な紹介は誤解を招きかねないので避ける。立ち読みでもいから、本書を一度手にとって欲しい。目から鱗というか、「そうそう」もしくは「あるある」、「よくぞ言ってくれた」という気持ちになるだろう。

ソフトウェア開発は、人が命である。よい土壌からよい野菜が収穫できるように、人に対する配慮や環境づくりがソフトウェア開発を成功に導く鍵である。(渡辺 登)

ソフトウェア・エンジニアリング関連イベントカレンダー

作成: SEC Journal編集委員会

開催時期	開催日	イベント名	主催	開催場所	URL
2006年5月	10日(水)~11日(木)	Embedded Technology West 2006(組込み総合技術展 関西)	社団法人 組込みシステム技術協会(JASA)	大阪府大阪市・マイドームおおさか	http://www.jasa.or.jp/etwest/
	17日(水)~19日(金)	IPAX2006 (ビジネスショウと共同開催)	IPA	東京都江東区・東京ビッグサイト	http://www.ipa.go.jp/
6月	9日(金)	連続セミナー 第1回 「CIOを取り巻く環境」	社団法人 情報処理学会	東京都千代田区・東京電機大学神田キャンパス 7号館1F丹羽ホール	http://www.ipsj.or.jp/
	12日(月)~13日(火)	SEC Forum 2006 12日:エンタプライズ系セッション 13日:組込み系セッション	IPA/SEC	東京都千代田区・大手町サンケイプラザ	http://sec.ipa.go.jp/
	28日(水)~30日(金)	SODEC / ESEC	リード・エグジジション・ジャパン	東京都江東区・東京ビッグサイト	http://www.sodec.jp/ http://www.esec.jp/
7月	7日(金)	ITスキル標準プロフェッショナルコミュニティ(IPCF2006)	IPA/ITスキル標準センター	東京都港区・明治記念館	http://www.ipa.go.jp/
	12日(水)~13日(木)	ソフトウェア・プロセス・エンジニアリング・シンポジウム2006 (SPES2006)	社団法人 情報サービス産業協会(JISA)	東京都江東区・日本科学未来館	http://www.jisa.or.jp/seminar/
	18日(火)~21日(金)	ソフトウェア・シンポジウム2006	ソフトウェア技術者協会	熊本県熊本市・ウエルシティ熊本	http://ss2006.kmt-iri.go.jp/
10月	2日(月)	情報化月間記念特別行事	経済産業省	東京都港区・全日空ホテル	http://www.ipa.go.jp/
	5日(木)~7日(土)	ネットワーク・セキュリティ・ワークショップ in 越後湯沢 2006	NPO新潟情報セキュリティ協会(ANISec)	新潟県湯沢町・湯沢町公民館 / イナモト旅館	http://www.yuzawaonsen.gr.jp/conf/
	11日(水)~13日(金)	SEPG Japan 2006 (仮称)	日本SPIコンソーシアム(JASPIC)	茨城県つくば市・つくば国際会議場	http://www.jaspic.jp/
	18日(水)~20日(金)	Security Solution 2006	日経BP社	東京都江東区・東京ビッグサイト	http://expo.nikkeibp.co.jp/secu-ex/
	24日(火)	IPA Forum 2006	IPA	東京都港区・明治記念館	http://www.ipa.go.jp/
11月	15日(水)~17日(金)	Embedded Technology 2006 (組込み総合技術展)	社団法人 組込みシステム技術協会(JASA)	神奈川県横浜市・パシフィコ横浜	http://www.jasa.or.jp/et/

上記は変更される場合があります。参加の際に必要な詳細事項は主催者にお問合せをお願いします。

「SEC Forum 2006」のご案内

6月12日(月): 午前は、経済産業省商務情報政策局情報処理振興課長 鍛冶克彦氏による基調講演、及びSEC鶴保証城所長の講演。午後の講演は下記の内容。

「エンタプライズ系プロジェクト報告」「要求品質の確保に向けた具体的活動」「説明力のあるソフトウェア開発見積り」「定量データの定義・収集・分析」「ITプロジェクト見える化~下流工程編~」「先進ソフトウェア開発プロジェクトからの報告」

6月13日(火): 午前は、組込み系プロジェクトの総括と、経済産業省「組込みソフトウェア産業実態調査」の報告。午後の講演は下記の内容。

「ETSS: 人材育成・活用のための組込みスキル標準」「スキルマネジメント標準としてのETSS(講演: Dr.Richard Mark Soley (OMG会長))」「組込み開発プロセス標準」「品質向上のための設計モデリングの勧め」「組込みシステムのユーザビリティ向上」

入場無料 定員400名 事前にWebでの聴講申込みが必要です。詳しくはSECのWebサイト(<http://sec.ipa.go.jp/>)をご覧ください。

上記の内容は、変更となる場合もあります。

先日、経済産業省(METI)へSECの2005年度の成果物提供を行い、現在、成果物確認が行われているところです。確認終了後にはSECのWebサイトにて公開予定ですが、先行して、本SEC journalにて、「SEC2005年度活動概要」を特集としてまとめました。

さて、これまでも多くのご質問を受けましたが、ここで改めてMETIとSECとの役割分担についてご説明します。例えば「見積手法部会」は「エンタプライズ系ソフトウェア開発力強化推進タスクフォース(METI直轄の委員会)」の1つです。各部会はテーマ毎に設置され、目的が達成されると解消します。SECは、各部会の成果物(成果物はMETIに帰属しますが、METIからSECに移譲されています)を、経営層、開発技術者等の対象者に向けて、わかりやすい出版物等としてまとめ、大きなミッションである啓蒙・普及活動において提供させていただいております。本SEC journalもそのうちの1つです。今期の出版書籍としては、他に8種類を予定しています。これらの出版書籍は一般書店でも購入できますが、6月12日、13日に開催予定の「SEC Forum 2006(於:大手町サンケイプラザ)」においては無償配布しますので、ご参加頂きたくお願いいたします。

今号では共同研究を行って頂いている大学や団体の方々には少ない誌面で報告して頂きましたが、次号では詳細報告を掲載できるように準備しております。

毎度、SEC journalで論文の募集を行っておりますが、残念ながら今号の締め切り間に合う投稿がありませんでした。全て広報担当者である私の責任で深く反省しております。開発者の皆さんに情報がより多く届くように、SEC Webサイトの利用者登録者を増やすべく、今度の「Embedded Technology West 2006(5月10日、11日 於:マイドームおおさか)」では登録者の受付を行う予定です。また、SEC Webサイトでは登録を随時受け付けておりますので、周りの方々にお声を掛けて頂き、「出版物のPDF閲覧や、メルマガ」等により、SECの成果物の活用を期待いたします。(ヒゲ)

本journalに対してのご意見もお待ちしております。<ご意見用メールアドレス: sec-journal@ipa.go.jp>

SEC journal 編集委員会

編集委員長

猪狩 秀夫

編集委員(50音順)

青木 奈央

赤田 眞弓

伊東 稔

菊地奈穂美

田丸喜一郎

樋口 登

松浦 清

神谷 芳樹

門田 浩

渡辺 登



東京都文京区 六義園(2006年4月 神谷芳樹撮影)

SEC journal® 第2巻第2号(通巻6号) 2006年4月28日発行

© 独立行政法人 情報処理推進機構 2006

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 鶴保 征城

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。

お知らせ

SEC journal 論文賞

SEC journalでは、掲載論文等について、
下記のように論文賞を設定しています。
投稿をお待ちしております。

論文賞概要

対象論文

SEC journal 5号~7号掲載論文
SEC journal論文賞投稿論文

論文賞審査会

IPA Forum 2006(2006年10月24日開催予定)
SECコンファレンスにて優秀論文発表及び最優秀論文審査会開催予定

前回のIPA Forum 2005開催結果

<http://sec.ipa.go.jp/event/20051024.php>

SEC journal論文賞

最優秀賞 : 1名 副賞賞金 100万円
優秀賞 : 3名 副賞賞金 50万円

SEC所長賞 : 1名 副賞賞金 20万円
(対象:SEC journal掲載済論文)
副賞賞金は旧「ソフトウェア工学研究財団(RISE)」から継承した基金より充当します。

SEC journalに投稿された論文は、査読により本誌への掲載を決定します。
SEC journal論文賞に投稿された論文は、査読後、上記掲載論文と合わせて審査し、優秀論文を選定します。
選定された論文の著者には、SEC コンファレンスにて発表していただき、当日の審査会にて最優秀賞を決定、表彰いたします。

応募様式

応募様式は、
右記のURLをご覧ください。



<http://sec.ipa.go.jp/secjournal/oubo.php>

SEC journal バックナンバーの ご案内

<http://sec.ipa.go.jp/secjournal/>
よりご注文いただけます



SEC Journal No. 6

SEC Journal No.6
第2巻第2号(通巻6号)
2006年4月28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 鶴保 征城

独立行政法人 情報処理推進機構

Tel.03-5978-7543 Fax.03-5978-7517
URL: <http://www.ipa.go.jp/>
定価1,470円(本体1,400円)



IPA[®]
独立行政法人 情報処理推進機構

