

2008年2月29日発行  
第4巻第1号(通巻13号)  
ISSN 1349-8622

# SEC<sup>®</sup>

13

## journal

Software Engineering Center

### Fault-proneフィルタリング: 不具合を含むモジュールのスパムフィルタを 利用した予測手法

水野 修, 菊野 亨

IPA<sup>®</sup>

独立行政法人 情報処理推進機構  
<http://www.ipa.go.jp/>



1	巻頭言 和田 成史( 社団法人コンピュータソフトウェア協会(CSAJ)会長)
2	所長対談：羽生田 栄一 アジャイルプロセス協議会 会長 / 株式会社豆蔵 代表取締役会長 プライオリティが恒常的に変化する ビジネスアプリケーション開発に効果を発揮する アジャイルプロセス
6	論文 Fault-proneフィルタリング: 不具合を含むモジュールのスパムフィルタを利用した予測手法 水野 修,菊野 亨
16	SEC設立三周年成果報告会 ～ Software Engineering Best Practice Day ～ 実施報告
18	SEC設立三周年成果報告 ～ 三年間のSEC活動を振り返って～ 鶴保 征城
22	ソフトウェアエンジニアリング ベストプラクティス賞 審査報告
23	総 評 本位田 真一
24	ESPR・ESMRの社内展開実施事例 グローリー株式会社
28	専門職大学院組込み技術研究科の設置と建学の精神 東海大学
32	トヨタにおけるソフトウェア開発プロセスの取り組み トヨタ自動車株式会社
36	組込み技術者育成研修におけるETSS導入の取り組み 財団法人にいがた産業創造機構
40	テストアセスメントによる品質向上への取り組み 日本ユニシス株式会社
44	CoBRA法に基づく工数見積りモデル構築への取り組み 株式会社日立製作所
48	組込みシステムにおけるモデルベース開発( MBD )技術者のスキル標準 JMAAB MBDエンジニア育成ワーキンググループ
52	貢献賞 定量データ提供企業グループ
53	海外レポート ISO/PC 236 第1回会議参加報告 新谷 勝利
56	事例解説 プロセス改善を成功させるためのESxRの使い方 岩橋 正実
60	車載OS開発に関するESMRの活用事例 長江 宣宗
62	BOOK REVIEW
63	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
64	あとがき
65	お知らせ( 論文募集 / SEC journal バックナンバー )

## ソフトウェア業界変革の波、変化に対応し、変化を活かすために



社団法人コンピュータソフトウェア協会  
(CSAJ) 会長

和田 成史

(株式会社オービックビジネスコンサルタント  
代表取締役社長)

### IT業界の変革の歴史

IT業界は、これまでに何度も大きな変革を経験してきた。80年代にパソコンがビジネスに利用され始め、90年代にはダウンサイジング、オープンシステム化が本格化した。これによってメインフレームのレンタルや販売を中心としたビジネスは変革を余儀なくされた。また90年代半ばからインターネットの利用が拡大し、2000年に入るとインターネット技術は企業の情報システムに不可欠のものとなった。

### ソフトウェアを取り巻くテクノロジーの変化

そして今、ソフトウェア業界の足下に新しい変化の波がいくつも押し寄せ始めているように感じている。

たとえば、その1つがSaaS (Software as a Service) である。SaaSは、ベンダ側が保守運用を行うため、利用企業にITに関する専門知識を必要としない。また、利用料金がサブスクリプション型であるため、情報システムを新規に構築するような初期投資が不要である。

もちろん短所もある。たとえば、SaaSベンダのシステムに不具合が発生したり、ネットワークにトラブルがあったりすれば、SaaSの利用企業は業務に支障をきたすことになる。また、SaaSベンダからの情報漏えいを心配する声も少なくない。ただ、こうした問題は、第三者による情報セキュリティ監査や信頼性の高い次世代ネットワーク (NGN) の利用などによってカバーされていくだろう。

周知のとおり、経済産業省は、2008年度からSaaSを中小企業の生産性向上の鍵と位置づけて大規模な実証実験を始める。これを契機にして、SaaSの利用拡大が本格化するのではないだろうか。

OSS (Open Source Software) の本格的な利用拡大も大変革を引き起こす波の1つである。それほどスピードは速くはないが、OSSの利用は基本ソフトウェアから言語ソフトウェアやミドルウェアに拡大しており、一部であるものの、業務アプリケーション分野のOSSも登場している。かつて、OSSを巡っては、推進派と反対派による極端な議論が多く、マスコミの報道もどちらかに偏っていたが、現在では利用の現状を踏まえた冷静なものになってきているように思える。

さらに、もう1つの変化として、利用者がソフトウェアの新バージョンに昔ほど興味を示さなくなっているという事実がある。それは、ソフトウェアの持つ機能や性能が利用者のニーズを十分に満たすほどに向上したという証であり、そのソフトウェアの完成度の高さやソフトウェアベンダの技術力を証明するものである。また、情報システムの耐用年数が伸びることは社会全体として見れば悪いことではない。今後ソフトウェア開発企業は、利用者の潜在ニーズを掘り起こし、真に魅力的な新バージョンを世に送り出していくと同時に、未開拓の領域を開拓する努力が必要だろう。

### 変化をとらえ、変化を活かしていくために

こうした変化は避けられるものではない。であるならば、変化に呑み込まれてしまうのではなく、変化に対応し、変化を活かすことを考える必要がある。それには、現状に満足することなく、アンテナを高く張り、常に自らを変革していく姿勢が重要だと考えている。

SECの産学官の連携による、研究や、標準化は、このような変革の時代において、大変意義のある重要な取り組みである。今後は、当協会としても、情報連携をより一層高めてゆきたい。

# プライオリティが恒常的に変化する ビジネスアプリケーション開発に 効果を発揮するアジャイルプロセス

企業環境の変化スピードが速くなるなか、情報システムも変化を取り込んで開発することが求められている。それを実現する手法として注目されているのがアジャイルプロセスである。アジャイルプロセス協議会の会長として日本におけるアジャイルプロセスの先導役を担っている羽生田栄一氏（株式会社豆蔵 代表取締役会長）とIPA/SECの鶴保征城所長が、ビジネスアプリケーション開発をめぐる課題やアジャイルプロセスを用いた開発の考え方、そして実際のアジャイル適用手法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、日本電信電話公社  
（現NTT）入社。NTTソフトウェア研究所  
所長、NTTデータ通信株式会社取締役開発本  
部長、同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。

- ・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～）
- ・高知工科大学工学部情報システム工学科  
教授（2003年～）
- ・日本BPM協会 副会長（2006年～）
- ・実践的ソフトウェア教育コンソーシアム  
会長（2006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** 羽生田さんはアジャイルプロセス協議会の会長を務められていますね。まず、アジャイルプロセスとはどのようなものかについてお話ししたいとお思います。

**羽生田** 人によって様々な捉え方がありますが、ひとことで言うと、ソフトウェアの設計に対して意識の転換を迫る一種のパラダイム変化と見るのが一番よいと思います。従来のソフトウェア開発は、ユーザからのフィックスされた要求に対して正しく設計し、コーディングし、テストしてリリースすればよい、つまり上流工程から下流工程へという流れを持つ考えですが、アジャイルは、そのような考え方をビジネス・ドリブンの設計へと変えるものなのです。

**鶴保** 従来型のソフトウェア開発では、要求定義があいまいなままに発注が行われることが少なくありません。それが開発工程を混乱させ、リリースを遅らせる要因となっています。そうした問題を防ぐために、現在のソフトウェア開発では上流工程を明確にすることを重視しています。その点に関してはどのように考えられますか。

**羽生田** 業務系のシステム開発の8～9割は、短期間での完成が求められています。また、システムが生み出すビジネス価値やユーザが求める使い勝手にフォーカスすることが必要です。そうしたことを実現しようとする、最初の段階で100%要求を固めるというスタンスは合わないでしょう。そこでは、ユーザの業務部門やマーケット部門を巻き込み、システムの姿を揉むというプロセスのなかで仕様を少しずつ形にしていく方法が有効です。仕様が100%提示され、それを受け止めて開発するというのではなく、開発サイドもユーザサイドも一緒になって方向を練り上げるプロセスに参加することを通して、こういう設計やアーキテクチャが望ましいというところへ持っていく、ということです。

**鶴保** これまでのエンタープライズシステムの開発にはそうした考えが希薄でした。システム開発を委託する側と開発する側が分かれていたのですが、SECでは、経営者が参画しないと、要求品質が確保できない、という考え方のもとに、超上流という概念を提案してきました。アジャイルといった場合、開発プロセスの最終工程であるプログラミングのテストまでを含めて総合的に受託側と開発側が一緒に行うことが望ましい姿なのでしょうか。

**羽生田** そうです。しかし、最初から全体に適用するのが難しくければ、まずは要求定義プロセス、いわゆる仕様書を作成するプロセスをコラボレーションに行うことから始めるのもよいでしょう。関連部門のステークホルダーに加わってもらってワークショップ形式で徐々に要求モデルを揉んでいくということもできます。

## ユーザが要求定義にコミットすることが必要

**鶴保** 今、ソフトウェア業界では上流工程を含めて従来型の請

負契約をすることは厳しいといわれています。そこで、フェーシングという考え方が浮上しています。仕様を固めるところまでを第1フェーズとして契約するというように、契約を開発のフェーズ毎に分けるという考え方です。それは、アジャイルプロセスに近い考え方といえるでしょうか。

**羽生田** フェーシングだけではアジャイルプロセスといえないと思いますが、タイムボックスを設定し、そこで結果を出し評価し、次のフェーズにフィードバックすることはアジャイルプロセスの柱の1つです。アジャイルの基本的な発想は、重要なビジネス的な判断はユーザにしてもらうことです。そして、ユーザサイドと開発サイドが歩み寄って協力関係で開発を進めていきます。その意味で、フェーシングという考え方をしてもよいでしょう。しかし、アジャイルは、開発の途中で間違いが発見されたり、この機能の方が今のマーケットにとって重要だとわかったときにはプライオリティをシフトして開発に折り込むという手法ですから、たとえばフェーズの後半で「絶対にこの期間でこのコストで開発しろ」と投げるのであれば、あまりアジャイルとはいえません。

**鶴保** アジャイルでは、実際の契約はどのようになっているのですか。

**羽生田** 契約は、期間単位です。ユーザサイドと開発サイドで仮に設定した50フィーチャをこの期間で開発しましょう、ということを決めます。50個のフィーチャにはそれぞれプライオリティがあります。開発する中で、ユーザサイドと話をしていくとプライオリティやフィーチャ自体もどんどん変わっていきます。プライオリティが変更され、開発の順番や一部の内容が変わっても、ユーザサイドにとって、その時点でもっとも望ましいフィーチャが実現しているはずだ、という発想で期間契約するのです。

**鶴保** フィーチャは機能と解釈していいのですか。

**羽生田** そうです。ストーリーという言い方もします。それは内部の機能というより、ユーザサイドから見たときに意味のある業務ストーリーですね。さらにアーキテクチャ上も必要な仕組みが含まれるように設定します。

**鶴保** 開発契約を結ぶ際、技術者のスキルやパワーを考慮に入れるのですか。

**羽生田** 多少はあります。スキルレベルで開発スピードや生産性は多少違ってきます。何ヶ月か開発を続けると、その開発組織の巡回速度が見えてくるので、「生産性はこうです」とユーザサイドに示すことができます。しかし、開発対象業務の内容が大きく変わるとその数値も変更しなければならなくなります。そこは、難しいところですね。

**鶴保** そのような契約で開発を進めていくためには、ユーザサ



羽生田 栄一（はにゅうだ えいいち）1980年代中頃富士ゼロックス情報システム時代にSmalltalk-80システムに触れ、オブジェクト指向やモデリング技術に開眼。2000年にソフトウェアエンジニアリング技術専門のコンサルティング会社である株式会社豆蔵を立ち上げ、代表取締役を経て現在、取締役会長。技術士（情報工学部門）、オブジェクト指向やソフトウェア工学の実践適用に関するコンサルティングや教育を通じて後進の育成にあたる。現在、アジャイルプロセス協議会会長、SEC設計技術部会委員、パターンワーキンググループ主査、情報処理学会ソフトウェア工学研究会主査、IPA ITアーキテクト・コミュニティ委員などを務める。路地・トマソン物件・神社・富士塚・古書店等を巡る街歩きが趣味。オブジェクト指向技術関連の著書・訳書多数。

イドも開発サイドも意識を変える必要がありますね。

**羽生田** アジャイルプロセスがうまくいくかどうかは、ユーザサイドが要求定義という開発サイドとの接点のところで積極的参加にコミットしてくれるかどうかということにかかってきます。アジャイルプロセスでは、ユーザサイドがプロジェクトに入り込んで開発メンバーの一人として設計上の難しさやプライオリティを秤にかけて進めていくことが必要です。例えば、「この1ヶ月でこのフィーチャが設計的に難しいとわかったものの、ビジネス的に最も重視しているので優先してほしい」という判断をユーザは毎回行います。そして、その1ヶ月後にはその要求に対するワークプロダクトがリリースされ、デモを通じて要求と合っているか、使い勝手はどうかを評価し、次の1ヶ月のためのプランニングを行います。そういうことを毎月繰り返していくことがアジャイルプロセスなのです。

## 1ヶ月で結果を見せる、反復によってアーキテクチャの質が向上

**鶴保** 海外では、アジャイルはどの程度普及しているのですか。

**羽生田** たとえば、アメリカでは金融業や大手の流通業でかなり使われています。アジャイルプロセス専門のコンサルティング会社により、数千人の規模でアジャイルによる開発を行っています。

**鶴保** 日本では、アジャイルで進められているプロジェクトはどのくらいあるのですか。

**羽生田** アジャイルだけで開発契約をしている小規模のソフトウェア会社は協議会に何十社が参加しています。プロジェクトの規模は大きくても数千万円で、億を超えるようなものはまだ多くはありません。また、製造業の内部で、設計だけではなく仕様を決める段階においてもアジャイルで進めている例もあり

ます。アジャイルは、ツールやパッケージなど製品開発に非常に向いているのです。ただ、いわゆるビジネスアプリケーションを受託開発するということになると、契約という問題が壁となってやりたくてもできないというのが現状です。

**鶴保** それは現在の受託構造に問題があるからでしょうか。

**羽生田** 大規模なプロジェクトの場合は、大きなベンダに保険を掛けたいという傾向と最初に仕様を固めないで開発にいくらかかるのか判断できないからということがありますね。

**鶴保** アジャイルで開発するときも予算の枠という概念はありますね。

**羽生田** ユーザサイドが「予算はこれです」と提示し、開発サイドは開発チームの人数から「では、何ヶ月のプロジェクトでやりましょう」といった感じです。あるいは、1つの反復でいくつのフィーチャを作り、それを何反復するという形ですね。ですから、見積りに関してそれほど従来型の概念と外れているわけではないのです。

**鶴保** 従来型の考えで仕様書を作ったとしても、現場的には開発対象の優先順位は変わっていきますね。

**羽生田** そうです。ですから、現場の実態を正しく仕様に取り込むとアジャイル的になるはずですが。従来の「最初に100%仕様を決めてから発注します」という考え方では現実的には回せません。製造業の場合は、実態がウォータフォールで回っていないので、アジャイル的な開発は自然に受け入れられるという土壌があります。

**鶴保** そうすると、アジャイルプロセスの活用の妨げとなっているのは契約の問題ではなく、仕事に対する取り組み方の問題ですね。

**羽生田** ご理解のとおりです。状況の変化にどんどん対応するマインドセットができあがっている製造業などの組織はアジャイルプロセスを受け入れやすいといえますね。

**鶴保** アメリカの自動車メーカーは部品メーカーと厳密な契約をしています。それに対して日本の場合は、自動車メーカーの設計を

先取りしながら部品の開発を並行して進めるコンカレントエンジニアリングを行っています。日本のソフトウェアの開発においても、これからはコンカレントエンジニアリングの考え方を取り入れるべきだと思っています。

**羽生田** そうですね。アジャイルプロセスは、実はコンカレントエンジニアリングの考え方に近いのです。そこからソフトウェアプロセスへの逆輸入的な面があります。

**鶴保** ところで、現状のシステム開発では、予算が1億円と決まると、その予算はよほどのことがないと変わりません。一方で、ユーザサイドは新しい要求を次々に出すことが珍しくありません。そして、開発サイドはそれを受け入れて無理に開発現場に流すこととなります。すると、開発作業が膨れあがって混乱し、收拾がつかなくなるプロジェクトとなります。それを繰り返してきていることが現在の問題です。

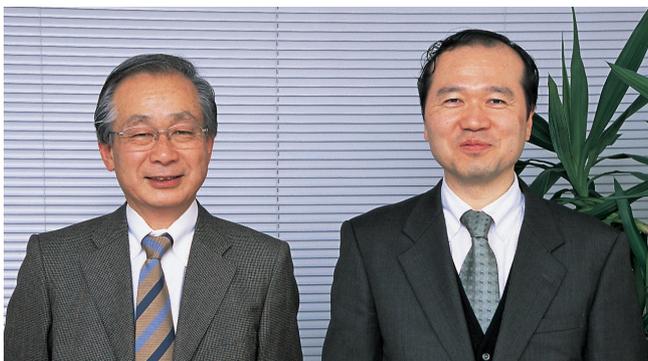
**羽生田** アジャイルの場合は、システム全体を1単位とするのではなく、たとえば、「1ヶ月で1,000万円」という期間での考え方です。1ヶ月で1,000万円に見合ったワークプロダクトを出す。その積み上げで、ムービング・ターゲットを捉えながらきちんとソフトウェアを動かしてみせるという考え方です。

**鶴保** 従来の開発では、概要設計、詳細設計、コーディング、単体テストを行い、最後になって動くものが出てくるのですが、アジャイルはそこが違うのですね。

**羽生田** そうなのです。その反復1回分の中で取り上げるフィーチャと取り上げないフィーチャをきちんと決めます。そして、1ヶ月で結果を見せます。ですから、その結果に対する評価判断が1ヶ月でできるのです。間違っている部分はその段階でわかるので、残りの期間で無駄な作業をすることがなくなります。そして、リスクの高い難しいフィーチャから取りかかることがアジャイルの原則なのです。ですから、リスクが高くて業務上重要なもののなかで、さらに重要なものを選び分けていき、1ヶ月で開発が納まるようにするのがアジャイルの手法です。

**鶴保** そのようにしてもビジネス上の都合で重要なものの開発が最後に追加になるということは起きませんか。

**羽生田** アジャイルは、そうなっても対応できるという考え方なのです。修正がすぐにできます。従来の開発手法では、下流工程の技術者は詳細設計のようにコーディングすればいいと指示されますが、アジャイルは、1ヶ月の中で要件も理解するし、設計もコーディングもテストも総合的な能力を持っているアーキテクトが全体を見て仕事を進めるのでアーキテクトの質も非常によくなります。しかも、重要なものほど前に開発するので反復を繰り返すなかで中核になるものほどテストも繰り返されることとなります。



## エンジニアのレベルアップにも大きな効果

**鶴保** アジャイルによる実際の開発プロセスはどのようなものですか。

**羽生田** 仕事の進め方でいうと、業務システムを10回反復して開発すると仮定して、最初は業務を理解しながら中心となる業務の構造について一通り動くような骨格、スケルトンの機能を作ります。そこで、業務に関する解釈が合っているか、確認します。誤解があったり、重要な機能追加の要求が出されたら、2回目の反復を行います。骨格に誤解がなければ、第3反復、第4反復で肉付けをしていくこととなります。そこでベーシックなものができる、使い勝手を向上させていくということで第5反復、第6反復、第7反復を行います。その繰り返しの中で、「マーケットの動きが変わったので最初の頃に作った機能を変更したい」という要求が出されたら、後半でも即対応します。それがアジャイルの進め方です。

**鶴保** それに伴って、ソースコードも変わるのですね。

**羽生田** 変わります。そのため、リグレッションテストが重要となります。ですから、アジャイルでは、テストファーストということで、テストコードは最初から実行可能なコードとして書かねばなりません。XP（エクストリームプログラミング）という手法は特にそうなのですが、テストドリブンといって、プログラムができていない段階で、そのプログラムを呼び出して実行可能と思われるテストコードをまず書いて実行します。すると、プログラムが本体にないわけですから当然エラーになります。本体を完成させたら、もう1回テストコードを実行して、最初に書いたテストコードが実行可能だということをもってそのプログラムのユニットテストが通ったと判断します。

要求とテストがきっかけとなって1個1個のプログラムが作られ、反復の最後の段階で各モジュールをビルドしてテストしてリリースします。反復の度に作ったプログラムをビルドしてテストするので、1反復毎にミニプロジェクトを毎回行う感じですね。そのため、全員がシステム全体のアーキテクチャに精通します。ですから、品質が悪くなりようがありません。プロジェクトに携わったメンバはずごいスピードで成長していきます。それが大きな効果です。

**鶴保** 私の経験でも、受託開発をしているソフトウェア会社から意見を聞いたところ、「どのような背景でこのプロジェクトや機能が決まったのか、わからない。それがわかれば作業に対する工夫ができる」という声がありました。システム全体を見たいというのです。技術者として当然の意見だと思うのですが、企業活動の中では意外と充足されていないのですよね。

**羽生田** アジャイルは製造業のセル生産に相当します。しかも、

XPという手法では、必ず二人で仕事をします。メンバも入れ替えるので、いろいろな仕事ができます。それを何週間も続けると、システムの全体像がわかるし、お客様のニーズもシステムの背景もわかるのです。結果的にそれが生産性を上げることにつながっています。

**鶴保** 大規模なプロジェクトになると、チームが幾つもできます。コミュニケーションが大変になりますが、その点はどうですか。

**羽生田** 従来、50人、100人規模で行っていたプロジェクトが半分くらいの人数でできることがアジャイルのいいところです。しかし、大規模なプロジェクトはやはり残るでしょう。その場合は、全体のアーキテクチャを考えるフェーズと、そのアーキテクトのもとでモジュール化されたサブシステムをアジャイルプロセスするというように分けるといいと思います。ただ、アーキテクチャの妥当性は検証したいでしょう。そのためには、全体の設計と個別のモジュール開発をコンカレントにしておいて、3ヶ月ほどの少し大きな単位でその同期をとる方法が有効でしょう。つまり、各サブシステムの担当分をビルドして全体を通して動かすのです。それを1年の間に何回か実行することが考えられます。

**鶴保** アジャイルで開発する際、エンジニアにはどの程度のスキルが必要となるのでしょうか。

**羽生田** 新人でも、言語が使える仕様書のリテラシーがあれば、2反復もするとプロジェクトの後半には普通のエンジニアに育ちます。

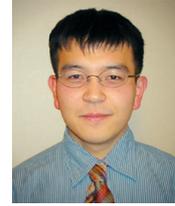
**鶴保** アジャイル普及のカギは何だと考えますか。

**羽生田** アジャイルで開発そのものに踏み込むことが大変であれば、まず、要求定義を仕様書に落とすところにアジャイル的な発想を導入したらいいと思います。今、ユーザは要求を整理することが自分の問題だと理解し始めています。ただ、どのようにしたら整理できるのかわからないのが現状です。ユーザサイドが開発サイドと一緒に要求定義のプロセスに関われれば、アジャイル的な整理の仕方とフィーチャのプライオリティ付けを連動させて開発に投げしてみようかという気持ちになるのではないのでしょうか。

**鶴保** アジャイルの啓蒙活動は、これまでソフトウェア開発のマインドを変えていこうということを中心とされてきたと思います。今後、アジャイル開発を進めるにあたっては、日本の開発組織にフィットさせる手法を探らねばなりません。そうしたテーマを取り上げることができれば、日本の製造業のよさと欧米流のソフトウェア工学のよさが融合した体系を日本から発信できることになるでしょう。今後も協力関係を築き上げていきましょう。

文：小林 秀雄 写真：越 昭三朗

# Fault-prone フィルタリング： 不具合を含むモジュールのスパムフィルタを 利用した予測手法



水野 修十



菊野 亨十

不具合を含みそうなソフトウェアモジュール (Fault-prone モジュール) の検出はソフトウェア工学における重要な問題の1つであり、これまでも多くの研究が行われてきた。それらの研究の多くはソフトウェアの複雑度メトリクスなどに基づいたモデルによる予測であった。しかし、モデルの構築にはメトリクスの収集環境が必要となるため、そのことも適用を難しくしている。そこで我々は、ソースコードに対して簡単に適用できる Fault-prone モジュールの検出手法として、スパムフィルタに基づいた Fault-prone モジュール検出法「Fault-prone フィルタリング」を提案している。この手法はソースコードのみを入力とすることができ、また、全く事前の知識がない状態からでも開発プロジェクトに適用できるという特徴を持つ。本論文では適用実験としてオープンソースソフトウェア eclipse とその関連プロジェクトに対して予測を行い、予測精度についての評価を行った。

## Fault-Prone Filtering: A Simple Approach to Predict Fault-Prone Modules Using Spam Filter

OSAMU MIZUNO, TOHRU KIKUNO

Prediction of fault-prone software modules has been one of the most classical and important area of software engineering so far. Many approaches has been carried out using software complexity metrics and mathematical models. Such approaches, however, have difficulties in collecting the metrics and constructing mathematical models based on the metrics.

We proposed a novel approach for predicting fault-prone modules using a spam filtering technique, named Fault-prone Filtering. In our approach, fault-prone modules are detected in a way that the source code modules are considered as text files and are applied to the spam filter directly. Source code modules are applied to fault-prone filter with Training Only Errors (TOE) procedure. By experiments using open source project, we confirmed that our approach has high accuracy.

### 1 はじめに

高品質なソースコードの作成はプロダクトの品質向上だけでなくコストの削減にもつながる。コードを作成した時点で不具合を含むかもしれない (Fault-prone) モジュールを特定できれば、早期にバグを除去できるだけでなく、レビュー、デバッグに費やす工数の削減も可能と

なる。そのため、これまでも Fault-prone モジュールを予測すべく、多くの研究が行われてきた [BELLINI200] [BRAIAND2002] [DENARO2002] [GUO2003] [KHOSHGOFTAAR2004] [MENZIES2007] [SELIYA2005]。従来の手法では、主にモジュールの複雑さや変更頻度等のソフトウェアメトリクスを用いて予測モデルを構築している。しかし、こうしたソフトウェアメトリクスを測定するためには、メトリクスの測定環境が必要となる。

† 大阪大学大学院情報科学研究科 Graduate School of Information Science and Technology, Osaka University

そこで、我々はソースコードのみを入力として与え、メトリクス等の測定なしにFault-proneなモジュールの予測が可能となる手法を提案する。この手法では、迷惑メール検出に利用されるスパムフィルタで利用される技術をソフトウェアのソースコードに対して適用し、純粹にコードのテキスト情報のみからFPモジュールを予測する。この手法を「Fault-prone フィルタリング」と呼ぶ[MIZUNO ESEM2007] [MIZUNO ESEC/FSE2007]。

Postini 社の調査によると、2006年11月の時点で世界中を流れる電子メールの94%はスパムメールであるとされている[POSTINI]。そのため、スパムメールをブロックする技術の開発が進められてきている。初期のスパムフィルタはあらかじめ登録した単語のパターンマッチによるものが主流であったが、この方法ではスパム送信者とのいたちごっこが続くため、根本的な解決にはならなかった。そうした中、Graham はベイズ識別器によりスパムメールの分別が可能であることを示した[GRAHAM2004]。このアイデアに触発され、多くのベイズ識別器に基づくスパムフィルタが開発され[BOGOFILTER] [POPFILER]、ユーザの手にスパムメールが届く可能性は激減しつつある[GOODMAN2007]。本研究ではYerazunis らによって開発されたスパムフィルタ CRM114 を用いる[CHHABRA2004]。

Fault-prone フィルタリングを実際のプロジェクトに適用するにあたっては、スパムフィルタリングで用いられる手法である「誤判定時のみ学習 (Training Only Errors: TOE)」という方式が現実的である。この方法は、まずモジュールの分類を行い、分類結果が実際の結果と異なっていたときのみ、そのモジュールを学習するという手法である。これは一般的に電子メールのフィルタリングを行う動作と同様である。この方法を用いることで、実際の環境に近い状況、すなわち事前の知識が全くない状態からの予測モデル構築を行うことが可能である。

本論文の以降の構成について述べる。まず第2節では、我々の提案するFault-prone フィルタリングの概要について述べる。第3節ではTOEの手続きとその効果を確認する小規模な実験について述べる。第4節では実用的な大きさの2つのプロジェクトへの適用実験について述べる。

また、第5節では実験結果についての議論及び関連研究との比較についてまとめる。最後に、第6節で本研究のまとめと今後の課題について述べる。

## 2 Fault-prone フィルタリングの概要

### 2.1 着想の背景

Fault-prone フィルタリングはスパムメール (迷惑メール) の判別を行うスパムフィルタで用いられるテキスト分類フィルタ技術を利用する。スパムフィルタは、過去に受信した電子メール内の単語群を利用して、スパムメールと通常のメールを判別するための辞書を作成する。そして、新たに受信した電子メールについては、ベイズ識別等の技術により、スパムか否かを判定する。学習は随時行われ、辞書は常にその時点の状況を反映したものになるため、新種のスパムメール等にも柔軟に対応できるとされている。この考えは、スパムメールには特定の単語群や文章が頻繁に含まれている、という事実に基づいている。

我々は、この考え方がソースコード内の不具合についても適用できるのではないかと考えた。もちろん、元々悪意を持って作成されたスパムメールと、意図的ではないがバグが混入したソースコードを全く同じものと見なすのは無理があるかもしれない。しかし、一連のソフトウェア開発においては、同じ開発者が同じ文脈でバグを混入することや、類似の関数やAPIの呼び出し等においてバグを混入してしまうことはよくあることだと考えられる。すなわち、スパムメールの中の特定の単語のように、バグが存在するところには特定のコード片が存在するのではないかと類推した。

### 2.2 スпамフィルタ: CRM114

本研究ではテキスト分類フィルタとしてCRM114を用いた[CRM114]。主にスパムフィルタとして開発されているが、汎用的な用途、例えば計算機のログ監視やネットワークのトラフィック監視等にも活用できるとされている。また、現在開発されているメールフィルタの中でも

表1 OSB で生成されるトークン例

a	=		
a		b	
a			+
a			1
=	b		
=			+
=			1
=			return
	b		+
	b		1
	b		return
	b		a
		+	1
		+	return
			a
		1	return
		1	a
			return a

高い予測精度をあげているものの1つである。

CRM114 は基本的にはベイズ識別を利用したテキスト分類フィルタであるが、複数の単語を組み合わせたものをトークンと呼び、学習・分類の単位として利用することが大きな特徴である。従来のテキスト分類フィルタは1単語をトークンとしているのに対し、複数単語の組をトークンとすることで、より複雑な学習が可能となっている。本研究では、CRM114 のデフォルトの分類手法である“Orthogonal Sparse Bigrams Markovmodel (OSB)”を使用する。OSB は任意の連続する5単語の組合せのうち、2単語からなるものだけをトークンとする手法である。

OSB によるテキスト処理について以下に簡単に示す。表1は“a = b + 1 ; return a;”という文について、トークンを生成した様子である。OSB ではある単語を起点として5単語からなる単語列に対し、正確に2単語のみを含むもののみを学習・分類の対象として抽出する。なお、本研究ではプログラム言語中の区切り文字をあらかじめ排除するため、“;”は単語群に含まれていない。

```
1: public int fact(int x) {
2:     return(x==1?1:x*fact(++x));
3: }
```

(a) 不具合を含む (FP) モジュール  $m_{FP}$

```
1: public int fact(int x) {
2:     return(x==1?1:x*fact(--x));
3: }
```

(b) 不具合を含まない (NFP) モジュール  $m_{NFP}$

図1 FP と NFP モジュールの例

```
1: public int sigma(int y) {
2:     return(y==1?1:y+sigma(++y));
3: }
```

図2 新しく作成したモジュール  $m_{new}$

### 2.3 CRM114 による分類の例

この節ではFault-proneフィルタリングがどのように不具合のあるソフトウェアモジュールを検出するのかを、単純な例を用いて説明する。図1 (a) と (b) はそれぞれ、不具合を含む (FP) モジュールと、含まない (NFP) モジュールの例である。以降ではそれぞれを  $m_{FP}$ ,  $m_{NFP}$  と表記する。fact ( ) は与えられた自然数  $x$  に対してその階乗を返すことを意図しているが、図1 (a) の実装では  $-x$  とすべきところを  $++x$  と誤記しているため、正常に動作しない。図1 (b) はその不具合を取り除いた状態である。この2つのモジュールのみが辞書に学習された時点で、図2 に示すモジュール  $m_{new}$  が新たに作成されたとし、このモジュールが不具合を含む確率を計算することを考える。なお、モジュール  $m_{new}$  は与えられた正整数  $y$  に対してその総和を求めるつもりであるが、 $m_{FP}$  と同様に本来  $--$  とすべきところを  $++$  としている。

まず、 $m_{FP}$  と  $m_{NFP}$  をそれぞれFP, NFP として学習する。この時、CRM114 によってそれぞれのモジュールについて図3 (a), (b) に示すようなトークンの集合  $T^{FP}$ ,  $T^{NFP}$  が生成される。 $m_{FP}$  から生成されたトークンの集合  $T^{FP}$  はFault-prone モジュールの特徴としてFP 辞書に格納される。同様に、 $m_{NFP}$  のトークンの集合  $T^{NFP}$  は、NFP 辞書に格納される。

public	int	public	int	public	int
public	fact	public	fact	public	sigma
public	x	public	x	public	y
int	fact	int	fact	int	sigma
int	int	int	int	int	int
int	x	int	x	int	y
int	return	int	return	int	return
fact	int	fact	int	sigma	int
fact	x	fact	x	sigma	y
fact	return	fact	return	sigma	return
int	==	int	==	int	==
x	return	x	return	y	return
x	x	x	x	y	y
x	==	x	==	y	==
x	1	x	1	y	1
return	x	return	x	return	y
return	==	return	==	return	==
return	1	return	1	return	1
return	?	return	?	return	?
x	?	x	?	y	?
x	:	x	:	y	:
==	1	==	1	==	1
==	?	==	?	==	?
==	:	==	:	==	:
==	x	==	x	==	y
1	?	1	?	1	?
1	:	1	:	1	:
1	x	1	x	1	y
1	*	1	*	1	+
?	:	?	:	?	:
?	x	?	x	?	y
?	*	?	*	?	+
?	fact	?	fact	?	sigma
:	x	:	x	:	y
:	*	:	*	:	+
:	fact	:	fact	:	sigma
:	++	:	--	:	++
x	*	x	*	y	+
x	fact	x	fact	y	sigma
x	++	x	--	y	++
*	fact	*	fact	+	sigma
*	++	*	--	+	++
*	x	*	x	+	y
fact	++	fact	--	sigma	++
++	x	--	x	++	y

(a)  $m_{FP}$  から生成されるトークン  $T^{FP}$

(b)  $m_{NFP}$  から生成されるトークン  $T^{NFP}$

(c)  $m_{new}$  から生成されるトークン  $T^{new}$

図3 各モジュールに対して生成されるトークン

新たなモジュールが与えられると、その時点で辞書に学習されている全てのトークンとのマッチングがとられ、確率の計算が行われる。図3は、 $m_{FP}$ ,  $m_{NFP}$ ,  $m_{new}$  について生成されるトークンの集合、 $T^{FP}$ ,  $T^{NFP}$ ,  $T^{new}$  を列挙したものである。図3(a)と(b)が現時点でそれぞれFPとNFPの辞書に格納されている全てのトークンであり、図3(c)は新たに生成されたトークンである。下線を引いた部分はモジュール $m_{new}$ と同一のトークンであることを表す。この図から、 $m_{FP}$ と $m_{new}$ の間で同一なトークンの数は14であり、 $m_{NFP}$ と $m_{new}$ の間で同一なトークンの

数は13であることが分かる。この情報から新規モジュール $m_{new}$ が不具合を含む確率 $P(T^{FP}|T^{new})$ を算出する。ベイズの定理によって、確率 $P(T^{FP}|T^{new})$ は次のように求められる。

$$\frac{P(T^{new}|T^{FP})P(T^{FP})}{P(T^{new}|T^{FP})P(T^{FP}) + P(T^{new}|T^{NFP})P(T^{NFP})}$$

まず、学習されているのは $T^{FP}$ と $T^{NFP}$ だけなので、それぞれのトークンの存在する事前確率は $P(T^{FP})=P(T^{NFP})=1/2$ となる。次に、 $m_{FP}$ のトークン $T^{FP}$ 内に $m_{new}$ のトークン $T^{new}$ が存在する確率 $P(T^{new}|T^{FP})=14/45$ である。また、 $m_{NFP}$ のトークン $T^{NFP}$ 内に $T^{new}$ が存在する確率 $P(T^{new}|T^{NFP})=13/45$ である。よって、確率は次式で求められる:

$$P(T^{FP}|T^{new}) = \frac{\frac{14}{45} \times \frac{1}{2}}{\frac{14}{45} \times \frac{1}{2} + \frac{13}{45} \times \frac{1}{2}} = 0.519$$

この結果、新規モジュール $m_{new}$ が不具合を含む確率は0.519となる。本研究では確率の閾値を0.50と定め、0.50以上であればFPモジュール、0.50未満であればNFPモジュールと判定する。よって、この例ではFPモジュールと判定されることになる。

### 3 Fault-Prone フィルタリングの適用

#### 3.1 誤判定時のみ学習 (TOE)

Fault-prone フィルタリングを適用するにあたって、我々は「誤判定時のみ学習 (Training Only Errors: TOE)」という方式を採用する。これは、スパムフィルタにおいても利用される方式であり、メールを到着順に分類し、その分類が正しかったかどうかを判断できる時点で学習を行う手法である。Fault-prone フィルタリングの適用実験では、次のような手順になる。

- (1) 当該プロジェクトのモジュール群を古いものから順番にソートする。

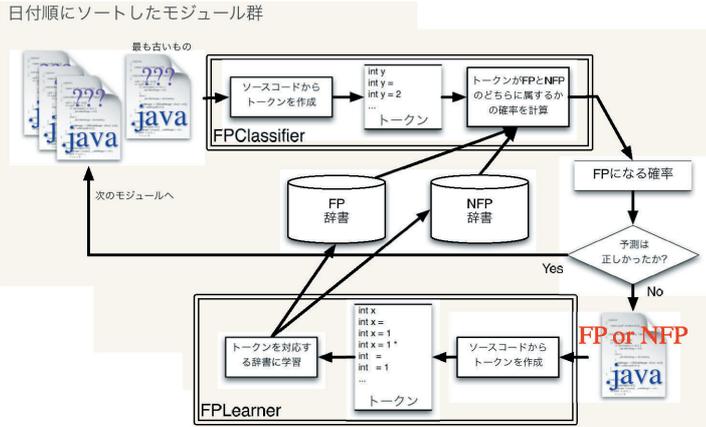


図4 Fault-prone フィルタリングでのTOEの流れ

- (2) 古いモジュールから順に1 つずつ取り出し、そのモジュールがFP かNFP であるかをFault-proneフィルタにより判定する。
- (3) モジュールに不具合が含まれていることが判明した時点で、予測が正しければ何もせずに(2)へ戻る。
- (4) 予測が正しくなければ、正しい結果を学習させる。例えば、NFP と予測したのに実際の結果がFP であった場合、FP 辞書に該当するモジュールの内容を学習させる。その後、(2)へ戻る。

図4 にこの手順を図示する。この実験はモジュールが作られた順を追ってFault-prone フィルタを適用するため、実際の開発環境に近い状態を再現することができる。

### 3.2 小規模なプロジェクトへの適用

ここでは図5 に示す小規模なソースコードの履歴についてTOE 実験を行った例を示す。図5 は階乗と総和の計算の結果を表示するプログラムを改訂した履歴である。ここでは5 つのリビジョン (Rev.) があったとしている。なお、本研究ではJava を開発言語としたプロジェクトを対象としたため、「モジュール」の単位をJava クラス内の「メソッド」と定義した。

表2 に、この実験で各モジュールがどのように予測されたのかと、その予測の正誤を示す。表2 の「実際」はそのモジュールが実際に不具合を含んでいた (FP) か、いなかったか (NFP) を示す。なお、本研究ではモジュールが変更された時にそのモジュールの予測を行うため、

```

Hello.java.1.1
public class Hello {
    public static int fact(int x) {
        return(x<=1?1:fact(x++));
    }
    public static void main(String[] args) {
        int n = 10;
        System.out.printf("%d\n",n,Hello.fact(n));
    }
}

Hello.java.1.2
public class Hello {
    public static int fact(int x) {
        return(x<=1?1:fact(x--));
    }
    public static void main(String[] args) {
        int n = 10;
        System.out.printf("%d\n",Hello.fact(n));
    }
}

Hello.java.1.3
public class Hello {
    public static int fact(int x) {
        return(x<=1?1:x*fact(x--));
    }
    public static int sigma(int y) {
        return(y<=1?1:fact(y++));
    }
    public static void main(String[] args) {
        int n = 100;
        System.out.printf("%d\n",Hello.fact(n));
        System.out.printf("%d\n",Hello.sigma(n));
    }
}

Hello.java.1.4
public class Hello {
    public static int fact(int x) {
        if (x < 0) {
            return -1;
        } else {
            return(x<=1?1:x*fact(x--));
        }
    }
    public static int sigma(int y) {
        return(y<=0?0:y+sigma(y--));
    }
    public static void main(String[] args) {
        int n = 100;
        System.out.printf("%d\n",Hello.fact(n));
        System.out.printf("%d\n",Hello.sigma(n));
    }
}

Hello.java.1.5
public class Hello {
    public static int fact(int x) {
        return(x<0?-1:(x<=1?1:x*fact(x--)));
    }
    public static int sigma(int x) {
        return(x<=0?0:x+sigma(x--));
    }
    public static void main(String[] args) {
        int n = 128;
        System.out.printf("%d\n",Hello.fact(n));
        System.out.printf("%d\n",Hello.sigma(n));
    }
}

```

図5 対象とするソースコード履歴

その直前の版から変更がないモジュールは予測の対象とならない。そのため、変更のないモジュールはこの表には含まれていない。Rev.1.1 から 1.5 までの間に変更されたモジュール数は11 である。

表2 TOE による予測結果

Rev.	モジュール名	実際	予測	正誤
1.1	main ( )	NFP	NFP	正
1.1	fact ( )	FP fact( x++ )が間違い 本来はx*fact( x-- )	NFP	誤
1.2	fact ( )	FP fact( x-- )が間違い 本来はx*fact( x-- )	FP	正
1.3	main ( )	NFP	FP	誤
1.3	fact ( )	FP 条件判定にx<0がない	FP	正
1.3	sigma ( )	FP fact( y++ )が間違い 本来はy+sigma( y-- )	FP	正
1.4	sigma ( )	NFP	FP	誤
1.4	fact ( )	NFP	FP	誤
1.5	main ( )	NFP	NFP	正
1.5	fact ( )	NFP	NFP	正
1.5	sigma ( )	NFP	NFP	正

より詳しく表2を見ると、次のようなことがわかる。

- fact ( ) については、Rev. 1.1 で本来FP であるところを NFP と誤判定している。続く Rev. 1.2 と Rev. 1.3 では正しく FP を FP と予測しているが、Rev. 1.4 で NFP を FP と誤判定する。最後に 1.5 では NFP を NFP と正しく予測している。
- sigma ( ) については、Rev. 1.3 で FP を FP と正しく予測しているが、Rev. 1.4 で NFP を FP と誤判定している。

以下では誤判定が発生した部分について詳しく説明する。

- Rev. 1.1 fact ( ) : 直前の Rev. 1.1 main ( ) において学習が行われないので FP, NFP とともに辞書は空である。そのため、事前に決められた値である NFP へと予測をしている。結果としてこの予測は間違いなので、Rev. 1.1 fact ( ) の内容が FP 辞書に学習される。
- Rev. 1.3 main ( ) : (1) で FP 辞書が学習されたことにより、main ( ) と FP の辞書内に一致点が発生する。そのため、main ( ) が FP と誤判定される。その結果 NFP 辞書に Rev. 1.3 main ( ) の内容が学習される。
- Rev. 1.4 sigma ( ) : バグは取り除いたが FP 辞書のほうに一致点が多いため FP と誤判定され、内容が FP 辞書に学習される。

表3 対象プロジェクトの概要

名称	EMF	EP
開発言語	Java	
リポジトリのサイズ	962MB	15.6GB
収集した不具合の状態	Resolved, Verified, Closed	
不具合の解決状況	Fixed	
不具合の重大度	blocker, critical, major, normal	
上記条件での不具合の数	4,042	44,600
CVS のログから発見した不具合数	2,832( 70.1% )	24,344( 54.6% )
FP モジュールの数	10,636	73,902
NFP モジュールの数	152,821	1,289,463

表4 実験結果の凡例

		予測	
		NFP	FP
実測	NFP	$N_1$	$N_2$
	FP	$N_3$	$N_4$

- Rev. 1.4 fact ( ) : こちらも (3) と同様バグは取り除いたが、FP の辞書に一致点が多いため FP と誤判定され、内容が FP 辞書に学習される。

## 4 大規模プロジェクトへの適用実験

### 4.1 対象プロジェクト

ここでは、オープンソースソフトウェアである Eclipse Modeling Framework ( EMF ), Eclipse Project ( EP ) [ECLIPSE] の 2 種類の開発データについて TOE を用いた実験を行った結果を示す。

TOE においては、本来事前にソースコードを準備する必要はない。しかし、本実験にあたっては事前にソースコードリポジトリからモジュールを取得しておき、そのモジュールにバグが含まれていたか否かの真値をあらかじめ把握しておく必要がある。そこで、本研究では文献 [SLIWERSKI2005] に示されているアルゴリズムを用いてオープンソースのソフトウェアリポジトリから FP モジュールと NFP モジュールを抽出した。

実験対象である Eclipse とその関連プロジェクトにおけ

るFP モジュール抽出の結果を表3 に示す．なお，モジュールの抽出に要する時間はプロジェクトの規模によって異なるが，Eclipse Project (EP) については約17 時間であった．

#### 4.2 予測精度の評価尺度

表4は今回の実験で得られる結果の凡例である．N1，N2，N3，N4 は横に示す予測と縦に示す実測にそれぞれ該当する例数を表す．この得られた結果の評価指標として精度 (Accuracy)，再現率 (Recall)，適合率 (Precision) を用いる．

精度 (Accuracy) は全モジュールのうち，実測がNFPのモジュールをNFP，実測がFPのモジュールをFPと正しく予測した割合を示す．よって精度は凡例の表4を用いると以下のように定義される．

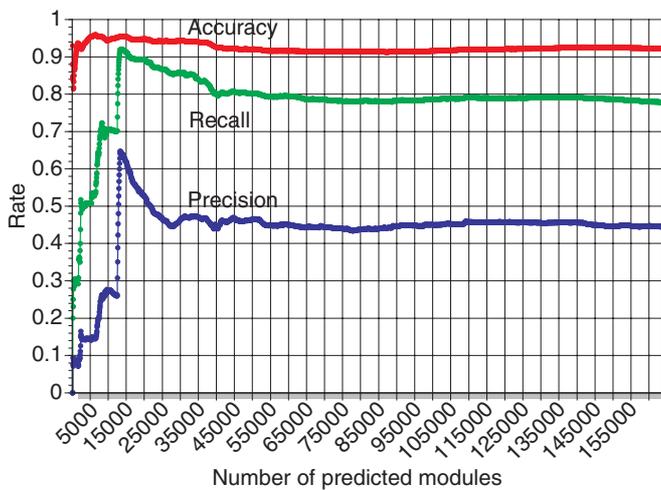
$$Accuracy = \frac{N_1 + N_4}{N_1 + N_2 + N_3 + N_4}$$

精度は予測の全体的な傾向を把握するには便利であるが，実測値の偏りなどに大きく影響を受ける指標であるため，この値のみで予測の良さを判断するのは危険である\*1．そのため，本研究では以下の2 つの指標に重点を置く．

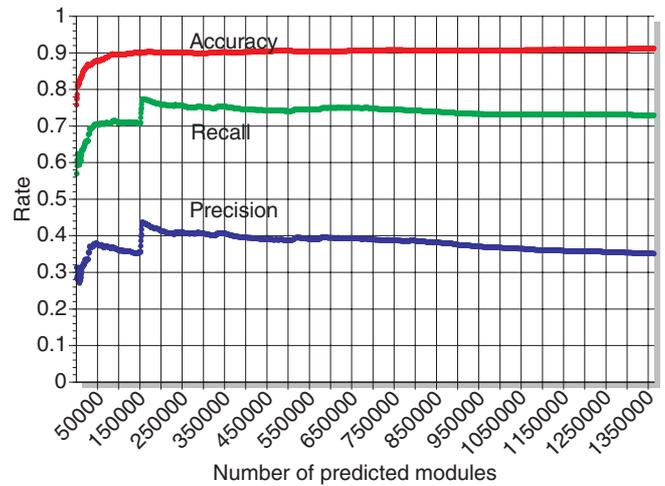
再現率 (Recall) は実測がFPである全てのモジュールのうち，正しくFPと予測できたものの割合を示す．よって再現率は以下のように定義される．

$$Recall = \frac{N_4}{N_3 + N_4}$$

直感的には，再現率は「予測によって実際の不具合をどれだけ網羅できるか」を示している．そのため，Fault-proneモジュール予測にあっては，不具合を未然に防くと



(a) EMF



(b) EP

図6 TOE 実験における各指標の推移

表5 TOE 実験での予測結果

(a) EMF の予測結果

		予測	
		NFP	FP
実測	NFP	142,482	10,339
	FP	2,358	8,278

精度 (accuracy) 0.922  
 再現率 (recall) 0.778  
 適合率 (precision) 0.445

(b) EP の予測結果

		予測	
		NFP	FP
実測	NFP	1,189,740	99,723
	FP	19,985	53,917

精度 (accuracy) 0.912  
 再現率 (recall) 0.730  
 適合率 (precision) 0.351

\*1 例えば，全ての予測をNFPとした場合でも，精度は  $\frac{N_1}{N_1 + N_2}$  となるので，実測でのFPとNFPの比を表す値を示す．

いう観点から最も重視すべき指標といえる。適合率 (Precision) は予測がFPであるモジュールのうち、実測がFPであったものの割合を示す。すなわち、適合率は以下のように定義される。

$$Precision = \frac{N_4}{N_2 + N_4}$$

適合率は、直感的には1つの不具合を見つけるのにどのくらい無駄なモジュールを調べる必要があるか、すなわちテストのためのコストを表している。

上記指標を利用すると3.2節における小規模な実験についても予測精度の評価を行うことができる。表2より、精度は0.636、再現率は0.75、適合率は0.50と計算できる。

### 4.3 適用結果

図6にTOE実験の経過を示すグラフを、表5に最終的な予測結果を示す。

図6のグラフについて説明する。このグラフの縦軸は精度、再現率、適合率の値を表す。また、横軸は作成時刻の時系列順にソートした全モジュールの通し番号を表している。TOEにおいては、古いものから順に分類・学習を行うため、一度全てのモジュールを作成時刻でソートしておき、古いものから若い番号を付けてある。例えば横軸の10,000はこの開発において10,000番目に作成されたモジュールであることを表し、その時点までのモジュールを予測した結果としての3つの指標の値をプロットしたものがグラフになっている。

表5は、各プロジェクトについて全てのモジュールを予測し終わった時点での予測と実測をクロス集計したものである。表5からは、Fault-proneフィルタリングによって精度と再現率に関してはそれぞれ高い値が得られたことを確認できる。再現率に関しては共に0.75前後の値を示しており、高い不具合予測能力が確認できる。また、適合率に関しても0.40前後と比較的高い値を示していることから、不具合の予測に必要なコストもそれほど高くないものと考えられる。このことから、これら2つのプロジェクトにおける実験ではFault-proneフィルタリングがうまく適用できていることが確認できた。

なお、TOE実験に要する時間は単純にモジュールの数

に比例する。最大の規模をもつEPにおいては約20時間を要したが、1つのモジュールの予測・学習に要する時間は0.5秒前後である。

## 5 議論

### 5.1 実験結果の考察

まず、表5に示す予測結果について考察する。2つのプロジェクトに対して適用した結果を比較すると、2つのプロジェクトについてほとんど同様の傾向を示している。その傾向とは、(1)再現率はおよそ0.75付近であること、(2)適合率はおよそ0.40付近であること、(3)精度はおよそ0.90付近であること、である。再現率についてみると、0.75前後という値は実際に発生する4つの不具合のうち3つまでは補足できるということを表しているため、率としては高いと考えられる。一方、適合率についてみると、0.40前後という値は、1つの不具合モジュールを発見するためにもう1つか2つは実際には不具合のないモジュールを調べなければならないことを表している。

実は、FPとNEPを判定する確率の閾値を動かすことで、再現率と適合率はある程度操作することが可能である。例えば、FPと判定される可能性を高く閾値を設定すれば、再現率は向上するが、適合率は下落する。再現率と適合率はトレードオフの関係になるため、品質を確保するためにどの程度までコストを費やせるかという問題になる。その解は実際に使う状況に依存すると思われる。

次に、図6について考察する。2つのグラフからは共に実験開始時は精度、再現率、適合率の値が激しく乱高下するのが確認できる。これはTOEの手順では全くの無学習から実験を行うためである。ある程度の数の学習を行った時点で指標の値が安定することも確認できる。例えば、EMFの例では実験開始から15,000モジュール程度の分類・学習を行った後に実用的な精度が得られている。一般的にスパムメールのフィルタリングでは高々数十通程度メールを学習させることで実用に耐える精度が得られることが確認されている。しかし、ソースコードに適用した場合の精度の向上は比較的遅いため、初期段階で

の精度向上が今後の課題である。

また、TOEの過程でEMFの13,000モジュール付近(図6(a))や、EPの150,000モジュール付近(図6(b))のように、再現率と適合率の値が激しく上昇することがあるのが確認できる。この現象は該当する期間に出現するモジュールの予測結果が全て正しくなることによって発生している。全ての予測結果が正しくなる理由については現在調査中であるが、リファクタリングなどにより、従来と全く同内容のソースコードがリポジトリに登録された結果このような現象が発生しているのではないかと予想している。

## 5.2 従来研究との比較

Fault-prone モジュールを予測する研究は古くから行われている。1999年以降だけでも多くの研究がなされている[BELLINI2005][BRAIAND2002][DENARO2002][GUO2003][KHOSHGOFTAAR2004][MENZIES2007][SELIYA2005]。

多くの研究では複雑度やソフトウェアの構造に関するソフトウェアメトリクスを収集し、数理的モデルを作成することで予測を行っている。例えば、Briandらはオブジェクト指向メトリクスからロジスティック回帰モデルを作成することでFault-prone モジュールの予測を行っている[BRAIAND2002]。また、KhoshgoftaarらはMcCabeの複雑度メトリクスやHalsteadのメトリクス等を用いて、さまざまな分類手法によるFault-prone モジュール予測の比較を行っている[KHOSHGOFTAAR2004]。

ここでは、これまでに提案されてきた手法をその予測精度を中心にまとめる。Denaroらはロジスティック回帰モデルに基づくFault-prone 予測を提案し、その精度は0.906、再現率は0.682であったとしている[DENARO2002]。上で述べたBriandらの手法では精度が0.840、再現率は0.483であった[BRAIAND2002]。また、GuoらはDempster-Shafer Belief Network を用いた予測手法を提案し、精度は0.690、再現率は0.915であったと報告している[GUO2003]。Belliniらの研究では判別分析を用いて予測を

行った結果、精度は0.736、再現率は0.543となっている[BELLINI2005]。Khoshgoftaarらは再現率を評価尺度とせずに第二種の過誤の率<sup>\*2</sup>を評価尺度としている[KHOSHGOFTAAR2004]。それによると、分類手法によって異なるものの、およそ10%から20%の第二種の過誤が発生するとしている。

我々の提案するFault-prone フィルタリングでは表5から精度の最大値は0.922、再現率の最大値は0.778(共に表5(a)より)と比較的高い値を示している。また、第二種の過誤の率も計算してみると、2%前後と低い値を示した。

もちろん実験する環境が違うために一概に優劣を比較はできないが、Fault-prone フィルタリングでは従来手法で示されている程度の精度、再現率を達成できていることがわかる。

## 5.3 現場への適用

本研究の成果は、開発の現場において次のように利用できると考えている。まず、モジュールの作成・変更の際にFault-prone フィルタリングを適用することにより、各モジュールにおける不具合の含有可能性を確率として得ることが可能になる。フィルタリングに利用するのはソースコードのみであるため、適用に必要な手間は少ないと考えている。得られた確率の情報から各モジュールについての不具合の出やすさがわかるため、例えば高い不具合確率が示されたモジュールに関しては、以降の開発においてそのモジュールに関連する部分を十分に精査することが可能となる。また、テストやレビューによる不具合発見時にはその不具合に関連するモジュールの情報を学習という形でフィードバックすることにより、不具合検出の精度を改善していくことが可能になる。

ただし、現時点での本手法はあくまでモジュールの中に不具合が存在する可能性を指摘するのみであるため、具体的な不具合の検出等には別の手法との組合せが有効であると考えられる。例えば、本手法は多くのモジュール中で特に不具合の含まれそうなものをおおまかに選択する

\*2 第二種の過誤の率は表4の表記では  $\frac{N_2}{N_1+N_2+N_3+N_4}$  となる。

ものとし、具体的な不具合の検出には静的解析ツール等を併用する、等の方法が考えられる。

将来的には、あるモジュールについて不具合を含む確率と共にそのモジュールに関する過去の不具合情報を提供することで、本手法単体での有用性も高められると期待している。

## 6 まとめと今後の課題

本論文では、スパムフィルタを利用したFault-prone モジュールの予測手法、Fault-prone フィルタリングとその実用的な適用手法、Training Only Errors (TOE) について述べた。また、2つのオープンソースソフトウェアプロジェクトに対して本手法を適用した実験についても述べた。実験の結果、本手法が高い予測性能を持つことが確認できた。

本研究で提案するFault-prone フィルタリングはソースコードのみを入力として実施することができるため、実プロジェクトへの適用も容易であると考えられる。今後は企業で実施された開発への適用が挑戦すべき課題の1つである。

### 参考文献

- [BELLINI2005] P. Bellini, I. Bruno, P. Nesi, and D. Rogai : Comparing fault-proneness estimation models. In Proc. of 10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05), pp. 205-214, 2005
- [BOGOFILTER] <http://bogofilter.sourceforge.net/>
- [BRAIAND2002] L. C. Briand, W. L. Melo, and J. Wust : Assessing the applicability of fault-proneness models across object-oriented software projects. IEEE Trans. on Software Engineering, 28 (7) :706-720, 2002
- [CHHABRA2004] S. Chhabra, W. S. Yezauris, and C. Siefkes : Spam filtering using a markov random field model with variable weighting schemas. In Proc. of Fourth IEEE International Conference on Data Mining (ICDM 2004), pp. 347-350, 2004
- [CRM114] CRM114 - the Controllable Regex Mutator. : <http://crm114.sourceforge.net/>
- [DENARO2002] G. Denaro and M. Pezze : An empirical evaluation of fault-proneness models. In Proc. of 24th International Conference on Software Engineering (ICSE '02), pp. 241-251, 2002
- [ECLIPSE] Eclipse Project : <http://www.eclipse.org/>
- [GOODMAN2007] J. Goodman, G. V. Cormack, and D. Heckerman : Spam and the ongoing battle for the inbox. Communications of the ACM, 50 (2) :pp.25-33, February 2007
- [GRAHAM2004] P. Graham. Hackers and Painters : Big Ideas from the Computer Age, chapter 8, pp. 121-129. O'Reilly Media, 2004
- [GUO2003] L. Guo, B. Cukic, and H. Singh : Predicting fault prone modules by the Dempster-Shafer belief networks. In Proc. of 18th IEEE International Conference on Automated Software Engineering (ASE'03), pp. 249-252, 2003
- [KHOSHGOFTAAR2004] T. M. Khoshgoftaar and N. Seliya : Comparative assessment

of software quality classification techniques: An empirical study. Empirical Software Engineering, 9, pp.229-257, 2004

[MENZIES2007] T. Menzies, J. Greenwald, and A. Frank : Data mining static code attributes to learn defect predictors. IEEE Trans. on Software Engineering, 33 (1) :pp.2-13, January 2007

[MIZUNO ESEC/FSE2007] O. Mizuno and T. Kikuno : Training on errors experiment to detect fault-prone software modules by spam filter. In The 6th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE 2007), pp. 405-414, 2007, Dubrovnik, Croatia.

[MIZUNO ESEM2007] O. Mizuno, S. Ikami, S. Nakaichi, and T. Kikuno : Fault-prone filtering: Detection of fault-prone modules using spam filtering technique. In Proc. 1st International Symposium on Empirical Software Engineering and Measurement (ESEM2007), (to appear). Madrid, Spain.

[POPFILER] POPFile : <http://popfile.sourceforge.net/>

[POSTINI] Postini Inc. Postini Announces Top Five 2007 : Messaging Security Predictions As Email Spam Becomes Front Burner Issue Again In The New Year. [http://www.postini.com/news\\_events/pr/pr120606.php](http://www.postini.com/news_events/pr/pr120606.php)

[SELIYA2005] N. Seliya, T. M. Khoshgoftaar, and S. Zhong : Analyzing software quality with limited fault-proneness defect data. In Proc. of Ninth IEEE International Symposium on High-Assurance Systems Engineering (HASE'05), pp. 89-98, 2005

[SLIWERSKI2005] J. Sliwerski, T. Zimmermann, and A. Zeller : When do changes induce fixes? (on Fridays.). In Proc. of Mining Software Repository 2005, pp. 24-28, 2005

# SEC設立三周年成果報告会 ～ Software Engineering Best Practice Day ～ 実施報告

2007年11月28日、SECは「SEC設立三周年成果報告会～ Software Engineering Best Practice Day～」を開催いたしました（於：ウェスティンホテル東京）。第一部「招待講演・表彰」、第二部「導入事例報告会（エンタプライズ系・組込み系）」、第三部「情報交換会」の三部構成で行われ、おかげさまで全てのプログラムが満席となり、盛況のうちに幕を下ろすことができました。ここに、当日の様子を簡単にご紹介いたします。なお、講演資料の一部はSEC-Webサイト（<http://sec.ipa.go.jp/>）からダウンロードが可能です。そちらもご覧ください。

第一部では、主催者挨拶、来賓挨拶につづき招待講演として、「日本の科学技術政策とIT技術への期待」と題して総合科学技術会議 議員 奥村直樹氏にご講演いただきました。次に、SEC 所長 鶴保証城が3年間の活動を総括してご報告、講演いたしました。

引き続き、「ソフトウェアエンジニアリング ベストプラクティス賞」の表彰式に移り、審査委員長 本位田真一氏よりご講評いただき、IPA理事長 藤原武平太による表彰が行われました（ソフトウェアエンジニアリング ベストプラクティス賞の詳細については本誌22頁以降をご覧ください）。

第二部では、エンタプライズ系と組込み系に分かれてパラレルセッションを行い、それぞれ2本ずつの講演とパネルディスカッションによりSEC成果の導入事例をご紹介いただきました（写真1）。

第三部の情報交換会では、なごやかな雰囲気の中、参加

者の皆様の交歓が行われました。SEC設立以来変わらずにご支援くださっている方々にも多数ご参加いただき、SECメンバー同、心よりお礼申し上げます。

ホワイエに設置した書籍コーナーでは、3年間に成果物として発行したSEC BOOKSシリーズなどの書籍と『SEC journal』を配布いたしました（写真2）。26種類全てを抱えると相当な重量となりましたが、ご用意した書籍のほぼ全てをお持ち帰りいただきました。ぜひ積極的に導入、活用していただくことを期待しております。

講演後に回収したアンケートには、SECに対する高い評価、期待、激励の言葉と、今後の活動への要望が寄せられました。

次期3年も、SECはますます精力的にソフトウェアエンジニアリング活動を推進してまいります。今後ともSECの活動にご理解、ご協力賜りますよう、よろしくお願いいたします。  
「SEC journal」編集部



写真1 皆様メモをとりつつ熱心に聴講されました



写真2 書籍コーナー



# SEC設立三周年成果報告会 プログラム

～Software Engineering Best Practice Day～

【第一部 招待講演・表彰】 会場：ギャラクシールーム		
10:00～10:10	<b>主催者挨拶</b> 独立行政法人 情報処理推進機構 理事長 藤原 武平太	
10:10～10:20	<b>来賓挨拶</b> 経済産業省 大臣官房 審議官(IT戦略担当) 吉崎 正弘氏	
10:20～10:50	<b>招待講演「日本の科学技術政策とIT技術への期待」</b> 総合科学技術会議 議員 奥村 直樹氏	
10:50～11:35	<b>SEC成果報告</b> 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	
11:35～11:45	<b>ソフトウェアエンジニアリングベストプラクティス賞講評</b> 審査委員長 本位田 真一氏 (国立情報学研究所アーキテクチャ科学研究系研究主幹・教授／ 東京大学大学院情報理工学系研究科コンピュータ科学専攻 教授)	
11:45～12:20	<b>表彰</b> 独立行政法人 情報処理推進機構 理事長 藤原 武平太	
12:20～13:30	休憩	
【第二部 導入事例報告会(エンタプライズ系)】 会場：ギャラクシールーム(A+B)	【第二部 導入事例報告会(組込み系)】 会場：ギャラクシールーム(C+D)	
13:30～14:00	<b>講演「定量データに基づくソフトウェアエンジニアリングの実践」</b> 日本ユニシス 株式会社 品質保証部長 服部 克己氏	<b>講演「車載基盤ソフトウェアにおけるSEC成果の適用評価について」</b> 有限責任中間法人 JASPAR運営委員／ トヨタ自動車株式会社 BR制御ソフトウェア開発室 主幹 佐藤 浩司氏
14:00～15:20	<b>パネルディスカッション「工数/コスト変動要因の定量化とコントロール～定量的見積り方式を導入して～」</b> <b>コーディネータ</b> 株式会社ジャステック 常務取締役 兼 常務執行役員 営業本部長 太田 忠雄氏 <b>パネリスト</b> 株式会社日立製作所 情報・通信グループ プロジェクトマネジメント統括推進本部 担当部長 幕田 行雄氏 沖電気工業株式会社 情報通信グループビジネスサポート本部 担当部長 羽田野 尚登氏 アイエックス・ナレッジ株式会社 第5事業部グループマネージャー 桑原 高志氏	<b>パネルディスカッション「ESxRの導入と実践～企業の現場が語るESxRへの期待～」</b> <b>コーディネータ</b> IPA SEC 組込み系プロジェクトリーダー 門田 浩 <b>パネリスト</b> アルハイム株式会社 ソフト開発管理推進室 室長 三原 幸博氏 有限責任中間法人 JASPAR運営委員／ トヨタ自動車株式会社 BR制御ソフトウェア開発室 主幹 佐藤 浩司氏 日本電気株式会社 ソフトウェアエンジニアリング本部 グループマネージャー 三橋 二彩子氏
15:20～15:30	休憩	休憩
15:30～16:00	<b>講演「ユーザ企業における共通フレーム2007の実践的活用」</b> 清水建設株式会社 情報システム部 情報化企画・推進グループ 課長 寺田 尚弘氏	<b>講演「車載ECUソフトとリソースの現状について～人材育成へのETSS利用～」</b> トヨタテクニカルディベロップメント株式会社 第4電子開発部 部長 第2ソフトウェア開発部 シニアプロフェッショナルエキスパート 福森 英夫氏
16:00～17:20	<b>パネルディスカッション「プロジェクト見える化の適用実践と効果及び課題」</b> <b>コーディネータ</b> 新日鉄ソリューションズ株式会社 金融ソリューション事業部 生産技術グループ シニアマネージャ 長岡 良哉氏 <b>パネリスト</b> 株式会社 エヌアイディ 事業本部 品質管理課 課長 新名 貴久氏 T&D情報システム株式会社 事業三部部長 川崎 充範氏 エヌ・ティ・ティ・ソフトウェア株式会社 代表取締役社長 伊土 誠一氏	<b>パネルディスカッション「ETSSの実装と活用～企業の現場が語るETSSへの期待～」</b> <b>コーディネータ</b> 東海大学専門職大学院 組込み技術研究科 研究科長 教授／ IPA SEC リサーチフェロー 組込みスキル標準(ETSS)領域責任者／ 組込みシステム技術協会理事 大原 茂之氏 <b>パネリスト</b> 富士ゼロックス株式会社 開発生産統括本部 開発管理本部 技術教育センター マネージャー 今田 正卓氏 JMAAB/株式会社デンソー システム開発部開発室 主幹 森 裕司氏 財団法人 いいがた産業創造機構 情報戦略プロジェクト 新事業育成メンター プロジェクトマネージャー 星野 雅博氏 トヨタテクニカルディベロップメント株式会社 第4電子開発部 部長 第2ソフトウェア開発部 シニアプロフェッショナルエキスパート 福森 英夫氏
【第三部 情報交換会】 会場：スタールーム		
18:00～19:30	情報交換会	

# SEC設立三周年成果報告

## ～三年間のSEC活動を振り返って～

SEC所長  
鶴保 征城

業務の効率化から企業の付加価値を向上させる目的へと変貌しつつある情報サービス産業は、今や2006年度売上高17兆円、従業員数は84.1万人となり、日本の産業の重要な一角を占めるようになった。また、自動車、携帯電話、情報家電といったハードウェアにソフトウェアを組み込む、組込みソフトウェア産業は2007年度には開発費が3.3兆円、技術者数は23.5万人となり、年々増加の一途を辿っている。

それに伴い、「ソフトウェアの複雑化、曖昧な要件定義に起因するシステムトラブル」、「多重下請や要員派遣に依存したビジネスモデル」、「人材育成」等、様々な問題が顕在化している。例えば、つい最近起こった公共交通機関トラブルは、プログラム設計のミス、変更管理の不備、不適切なコーディング、更にテストの不徹底が複合的に生じた結果である。これはソフトウェアエンジニアリングの欠如を端的に示すものである。また、ビジネスモデルの問題も非常に根が深い。というのは、要員派遣や下請けの受注形態のベースとなっている基本的な考え方は技術者の時間売りであるため、受注側の経営者にとっては、プロジェクトが混乱し、技術者の対応する時間が増えれば増えるほど売上げが増えるという奇妙な構図となっているからである。

### 研究・活動及び成果の概要

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（以下SEC）は、このようなソフトウェア開発にかかわる諸問題を解決するため、経済産業省の支援の下、2004年10月に設立された。エンタプライズ系ソフトウェアと組込みソフトウェアにおける開発力強

化のための活動を展開すると同時に、ソフトウェアエンジニアリングの現場で、SECの成果適用を推進してきた。

現在、エンタプライズ系ソフトウェアと組込みソフトウェアの開発力強化のためのタスクフォースには、129の企業及び24の大学が参加しており、産学官からの合計361名のメンバーとSEC研究員44名が集い、継続して議論・研

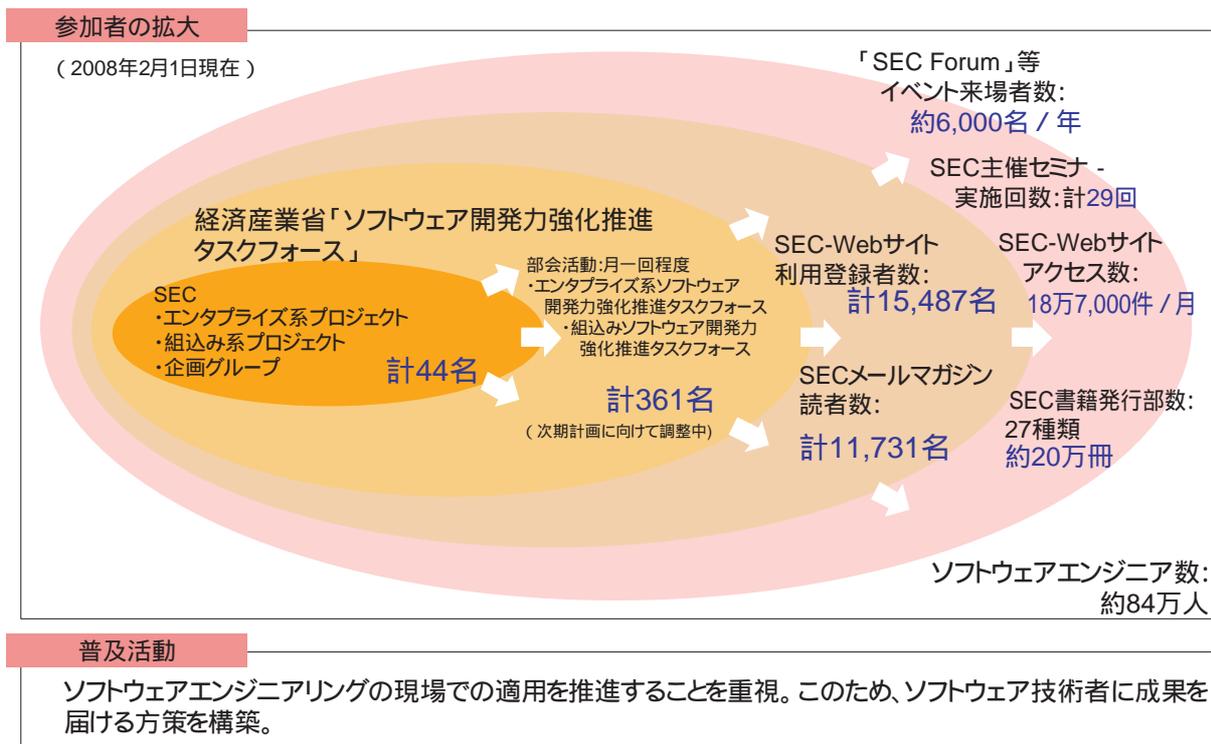


図1 SECの活動状況

究を重ねている（図1）

このように研究を進めてきたSECも2007年10月で丸3年が経過し、これまでの部会活動や普及活動を通じた成果がようやく実を結び始めている。これらは、合計27種類の出版物（発行部数総計20万冊）としてまとめたほか、3種類のツール（「ソフトウェア開発プロジェクト可視化ツール（EPM<sup>\*1</sup>ツール）」、「定量データに基づくプロジェクト診断支援ツール」、「組込みコーディング作法ガイド準拠テストスイツ」）を開発・公開している。また、SECの成果を対象とした実証実験は合計98件行われており、企業・団体における先進プロジェクトでも実際に適用されている。

### エンタプライズ系プロジェクト

エンタプライズ系プロジェクトは、現在、9つのプロジェクトが進行している（図2）。最近の主要な成果を以下に紹介する。

定量データ分析・手法では、複数の企業に協力いただき、2005年から着々と開発事例のデータを収集・分析し、毎年その結果をデータ白書としてまとめてきた。2007年度は合計1,774件に及ぶ開発事例のデータを収集・分析し、「ソフトウェア開発データ白書2007」として刊行した。さらに、この内容に基づいて自社のプロジェクトが置かれている相対的な状況やレベルを把握することのできる診断ツールを開発し、2008年1月に公開した。

プロジェクト見える化では、プロジェクトが抱える問題

を顕在化させる手法のガイドラインとして、プロジェクトにおける典型的なトラブルと対策を類型化し、「ITプロジェクトの『見える化』～上流工程編/下流工程編～」としてとりまとめた。また、ソフトウェア開発プロジェクトの可視化ツール「EPMツール」の検証に、現在合計43の組織が参加しているほか、全社で展開する事例も出てきた。

実践的な見積り手法では、ITソフトウェア開発時の見積り精度を向上させるための見積り手法として「ソフトウェア開発見積りガイドブック」を、また、ベンダ企業及びユーザ企業の保守フェーズの見積りにおける成功例・失敗例に基づく実践的な見積り手法として「ソフトウェア改良開発見積りガイドブック～既存システムがある場合の開発～」を、それぞれ刊行した。

ソフトウェアライフサイクルプロセスの国際規格であるISO/IEC 12207は、ソフトウェアの構想から開発、運用、保守、廃棄までの必要な作業項目を包括的に規定したものである。開発プロセス共有化では、このISO/IEC 12207を拡張して「共通フレーム2007」としてとりまとめた。「共通フレーム2007」は、経営者を含め、発注側と受注側の両者間で必要な、役割と責任の分担や業務の範囲・内容の例示、更に契約に関するの共通の認識をリファレンスとして広く利用できるようにしたものである。

プロセス改善では、プロセス改善に関するガイドとして「プロセス改善ナビゲーションガイド～なぜなに編/プロセス診断活用編～」をそれぞれ刊行、ISO/IEC 15504準拠の「SPEAK<sup>\*2</sup> IPA版」をSEC Webサイトからダウンロード可

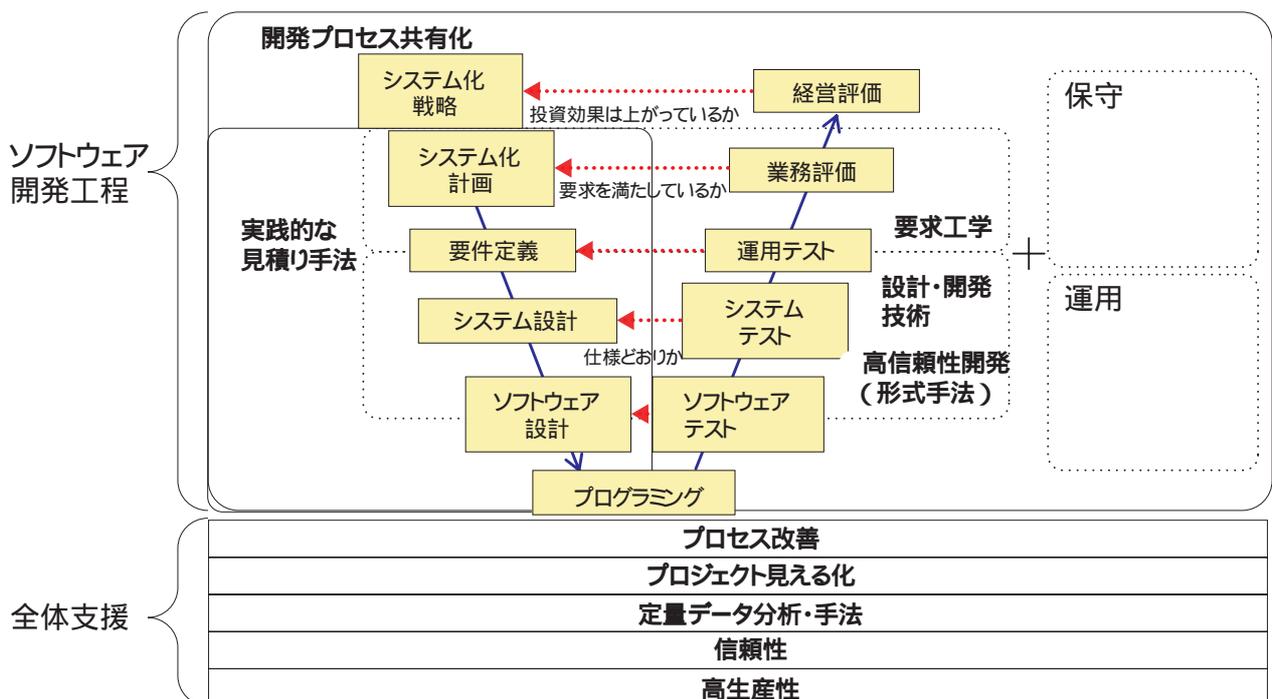


図2 エンタプライズ系プロジェクト

能とした。さらに、実証実験の開始、プロセス改善セミナーの開催等を通して、プロセス改善に関する啓発活動、プロセス改善活動の全国的普及に努めている。

### 組込み系プロジェクト

組込み系プロジェクトは、現在、エンジニアリング領域とスキル領域のプロジェクトに集約し活動している(図3)。最近の主要な成果を以下に紹介する。

高品質で生産性の高い組込みソフトウェアを効率的に開発するための開発手法・技術を整備・普及することを目的として、組込みソフトウェア開発ソリューション「ESxR<sup>\*3</sup>」を提供している。

- ・実装品質：「ESCR<sup>\*4</sup>」(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])

C言語を用いたコーディングに関する品質を意識した“良いソースコードの書き方”を、信頼性、保守性、移植性、効率性の4つの視点からまとめたガイドである。本ガイドに基づいて「組込みコーディング作法ガイド準拠テストスイーツ」も開発している。

- ・開発プロセス技術：「ESPR<sup>\*5</sup>」(組込みソフトウェア向け開発プロセスガイド)

組込みソフトウェア開発に必要な最小限の作業とプロセスを抽出・整理したガイドである。開発作業の指針とドキュメントテンプレートを提供している。改訂版にて「システム・エンジニアリング・プロセスと、システムの安全性や信頼性を実現するためのセーフティ・エンジニアリング・

プロセス」を追加した。

- ・マネジメント技術：「ESMR<sup>\*6</sup>」(組込みソフトウェア向けプロジェクトマネジメントガイド)

システム開発計画書に盛り込むべき具体的な要素を詳細に規定している。組込みソフトウェア開発における計画立案上、注意すべき事項を整理して紹介している。

また、組込みソフトウェア開発力強化のために、有効な人材育成や人材活用を実現するための指針として「組込みスキル標準 (ETSS<sup>\*7</sup>)」を策定し、「組込みスキル標準 ETSS概説書」の2006年度版を提供している。さらに、SEC Webサイトにて「ETSSスキル基準、ETSSキャリア基準、ETSS教育研修基準」を公開している。

その他、組込みシステムにおけるユーザビリティの重要性を紹介した「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)」、組込みソフトウェアの開発における有用なモデリング手法や設計における留意点等をまとめた「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)」、また、組込みシステムの安全性や信頼性の確保・向上させるための基本的な考え方を紹介した「組込みシステムの安全性向上の勧め (機能安全編)」をそれぞれ刊行した。

### 先進プロジェクト

2005年2月にソフトウェアエンジニアリング技術研究組合 (COSE<sup>\*8</sup>) が設立され、プローブ情報プラットフォーム

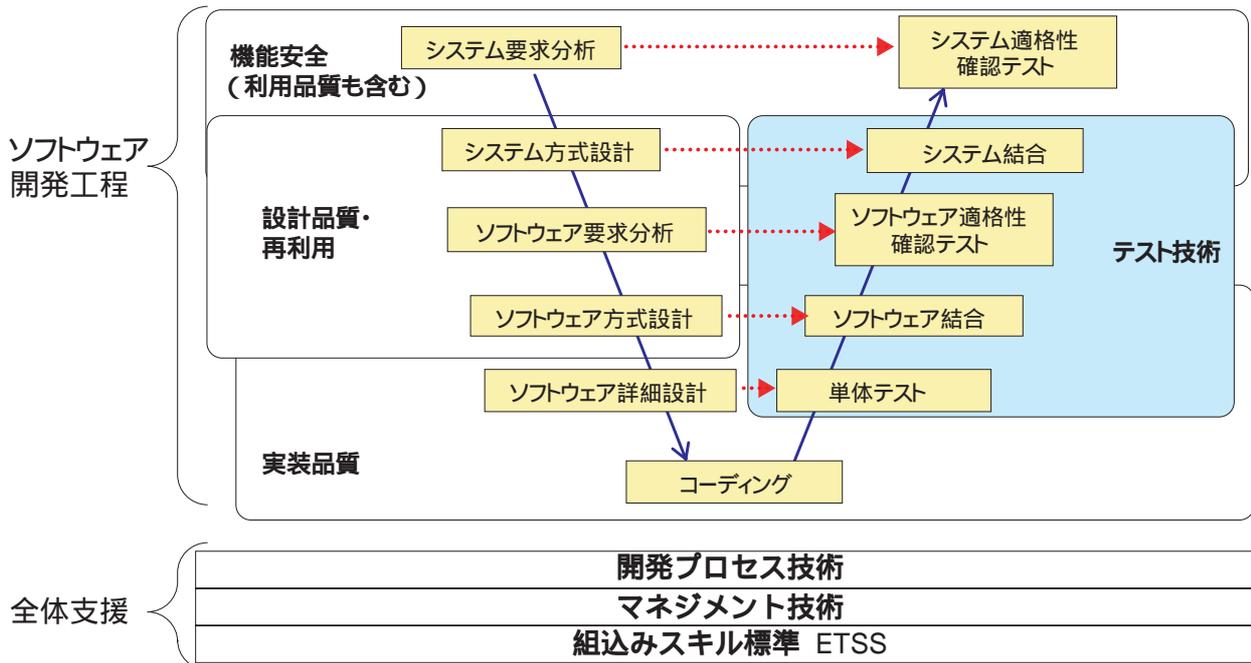


図3 組込み系プロジェクト

ムソフトウェア開発が、組合員企業の最新技術を持ち寄って行われた。このプロジェクトは既に終了しているが、この中でSECは進行中のさまざまなプロジェクトデータの計測を行い、収集したデータの分析結果をプロジェクトマネジメントに反映した。

また、次世代自動車ソフトウェア開発のための有限責任中間法人JasPar\*9が実施している開発プロジェクトについて、「EPMツールを利用したプロジェクトデータの収集」、「車載ソフトウェア分野に必要なスキルの定義」、「ESPRとESMRのテンプレートを利用したソフトウェア開発」について協力・支援している。

SECは高度なソフトウェアエンジニアリングの研究・普及を通じたソフトウェアのQCD（品質、コスト、納期）を向上させるべく、着実に成果を挙げ、いくつかのプロジェクトは成果作成期から成果普及期に移行している（図4）。このように、SECは情報サービス産業の問題解決のために努力しているが、現在の状況を鑑みると、まだまだその道のりは険しいと言わざるを得ない。例えば、開発モデルの問題である。これまで、工程をモジュール化し、前工程の生産物に不具合がないことを確認して次工程に進むやり方であるウォーターホールモデルが主流であった。ウォーターホールモデルは、上流工程でシステムの要求を明確に決定し、その後の工程での要求変更が比較的少ないシステム開発とは相性がよい。しかし、昨今のシステム開発では、上流工程で要求を詳細には決めることができない、あるいはは

上流工程で決めた要求に、その後の工程で頻繁に変更が生じるケースが増加している。このようなシステム開発にウォーターホールモデルを適用するには限界がある。スパイラル開発、イテレーション開発、アジャイル開発のような反復型開発を積極的に取り入れていくことが必要であろう。これは、システム開発における要求変更の割合という特性で分析した一例であるが、その他にも対象分野、規模（ソースコードや開発人員）、信頼性に対する要求の度合い、コストや契約形態を含めたビジネスモデル（受注型のシステムインテグレーションなのか、パッケージ開発なのか、ITサービス向けのシステム開発なのか）等、システム開発には様々な特性が存在する。一括りにシステム開発というのではなく、特性によって分類・整理した上で、適切な開発モデルを提示していきたい。

今後もSECが産学官の連携の媒体となり、このような問題を一つひとつ解決していくことが重要と考えている。

- \*1 EPM : Empirical Project Monitor
- \*2 SPEAK : Software Process Evaluation & Assessment Kit
- \*3 ESxR : Embedded System development eXperimental Reference
- \*4 ESCR : Embedded System development Coding Reference
- \*5 ESPR : Embedded System development Process Reference
- \*6 ESMR : Embedded System development Management Reference
- \*7 ETSS : Embedded Technology Skill Standards
- \*8 COSE : COnsortium for Software Engineering
- \*9 JasPar : Japan Automotive Software Platform Architecture

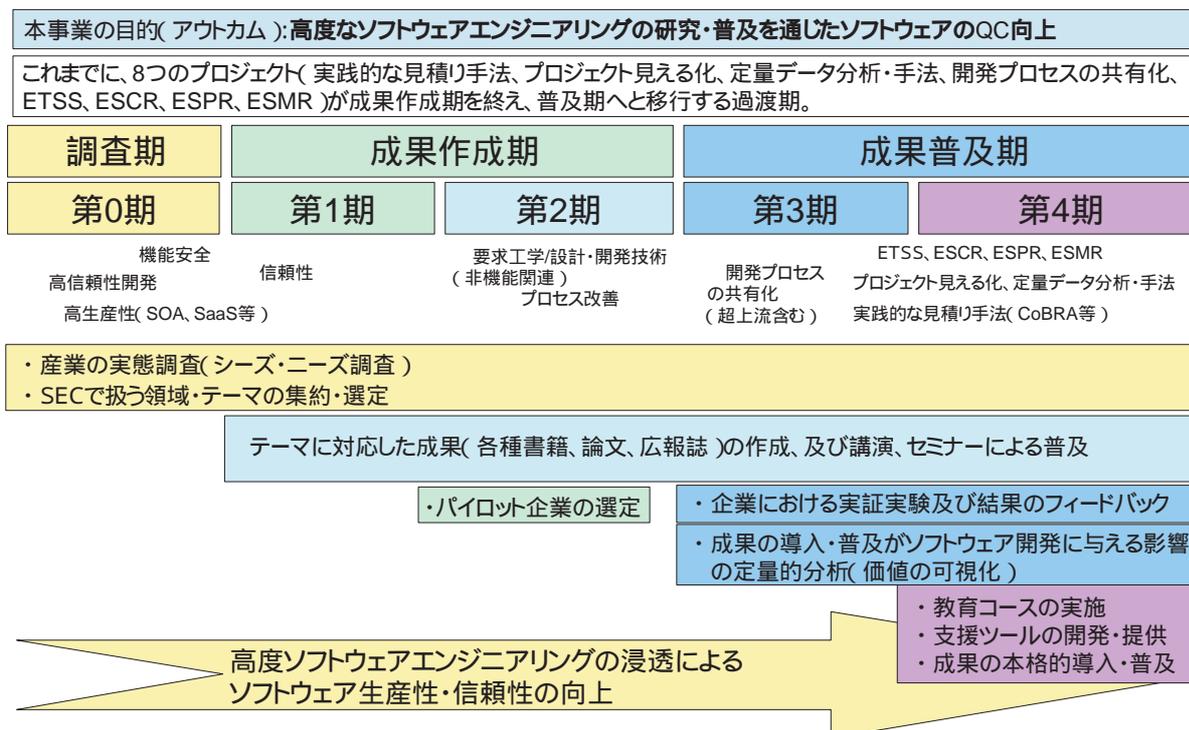


図4 SEC活動のライフサイクル



# ESPR・ESMRの社内展開実施事例



グローリー株式会社  
技術推進部

谷川 正仁



グローリー株式会社  
技術推進部  
テクニカルマネージャー

岡田 修二

グローリー株式会社には、各カンパニーの設計部長をメンバーとする全社横断的な設計標準委員会がある。そのなかにメカトロソフト担当TM（テクニカルマネージャ）及びGM（グループマネージャ）から構成されるメカトロソフト分科会があり、開発プロセスの整備・技術者教育・ソフト管理指標の検討を実施している。

設計標準委員会に属するソフト関連の分科会としては、上記メカトロソフト分科会（組込み系、現金処理機等のメカトロニクス制御）とシステムソフト分科会（エンタープライズ系、決済関連ソリューション等）がある。

ESPR及びESMRについて、メカトロソフト分科会を中心に社内展開した。

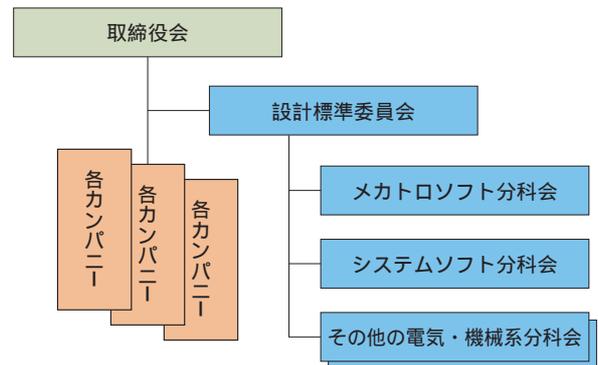


図1 設計標準委員会の組織体制図

## 1 G-SpiceE/M展開までの背景及び展開までの進め方

グローリーでは十数年前に“SWING手順”と称する“先ずメカ機構ありき”の組込み系主体のシステムソフト開発プロセスを構築していた。しかしSWING手順では、開発環境の変化、ISO 9001:2000準拠、市場要求レベルの変化、機械 電気 ソフトへの重点移行等への対応が難しく、開発者自身が独自にSWING手順を運用することで凌いでいた。しかし、特にシステムソフト系では独自運用の弊害が大きくなり、システムソフト分科会においては、数年前から企画等の上流工程でのプロセス整備に注力するとともに、Web技術を採用し、容易なメンテナンスと実例（各設計ドキュメント）へのリンクを可能とするシステムソフト開発プロセス“G-SpiceS”の整備を開始した。

一方単純なこととしては、各設計ドキュメントの名称が微妙に異なっていたため目的設計ドキュメントの検索に不都合が発生していたが、前方一致で検索可能となるよう

に設計ドキュメント名称付けのルールも見直した。

例：画面機能仕様書 機能仕様書（画面）

例：操作仕様書 機能仕様書（操作）

これらの改善の流れのなかで、組込み系においては、若手技術者の多い部門からSWING手順の根本的な見直しが必要とされた。設計者なら知っている当然と思われる設計ドキュメントのインプットとアウトプットの関係について理解の少ないまま設計作業を進めている若手も多く見られた。

その原因を調べるため、情報処理系の学部・修士卒の新入社員に研究室時代のソフト設計プロセスを聞き取った。その結果、設計ドキュメントも修正しないまま（そもそも設計ドキュメント自体が存在しない事が多い）先輩が作成したプログラムを直接修正していたことがわかった。そのためソフトウェア工学を学んでいると思うが、それを理解している新入社員は少なかった。

これらの課題に対応し現状を半歩進めるため、組込み系設計プロセスの再構築を模索していたときにSECのESPR

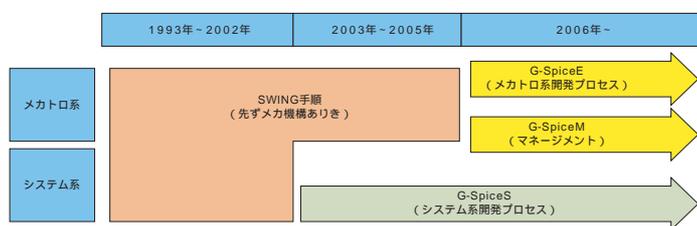


図 2 活動実績

の存在を知った。詳細は後で記載するが、ESPRをベースに当社開発プロセス（SWING手順）の考え方を加味して新開発プロセス（G-SpiceE）を策定した。

メカトロソフト分科会のメンバ（TM、GM）には、できるだけ負担がかからないように事務局2名で原案を作成し、2～3回/月程度の割合で分科会を開催して審議し、それをメンバが自部門で説明・意見集約するという方法で進めた（約8ヶ月）、同様にG-SpiceMも推進した（約6ヶ月）（図2）

## 2 G-Spice全体構成

G-Spiceは、組込み系プロセスガイドのG-SpiceE、システム系プロセスガイドのG-SpiceS、マネジメントガイドのG-SpiceM、教育体系のG-SpiceSS（計画中）から構成されている。上記仕組みは社内イントラネット環境上で構築されており、ソフト設計者がWebブラウザで参照できる。

今回受賞対象となったG-SpiceE、G-SpiceMについて以下に説明を行う。

## 3 G-SpiceEの概要・構成・特徴

### (1) 概要

G-SpiceEはESPRをベースに、自社で実施すべき作業が明確となるように上流工程に対して重点的に変更を加えた。利用される場面をまとめると表1ようになる。

表 1 G-SpiceEの利用目的

利用される場面	開発プロセスガイドの利用目的
技術構想の検討	・技術構想マップを使用し、構想に必要なノウハウの確認及び製品全体のビジュアル化を行い、上流工程での知恵の集約を図る。
プロジェクト計画	・各工程で実施すべき作業内容と目的を確認する。 ・プロジェクトに最適なフェーズ設計を検討する。
ドキュメントの作成	・ドキュメントガイドラインを参考に記載すべき必要事項に漏れがないかを確認し、ドキュメント精度の向上を図る。 ・事例ドキュメントを参考にし、ドキュメントの品質向上を図る。
設計委託	・委託先に対して委託作業及びアウトプットドキュメントの明確化と記載内容の深さを指示し、受入時の品質向上を図る。
技術者教育	・ソフトウェア工学に基づいた開発プロセスの重要性及びプロセスに基づいた仕事の進め方を理解する。

### ポイント

ESPRで定義されたプロセス名と社内では実施しているプロセス名との対比表を用意し、自社のプロセスとして実施すべき作業をタスク単位に分割し、アクティビティガイドとしてまとめている。

システム方式設計で機械設計・電気設計・ソフト設計の役割の明確化を行い、技術構想マップを用いて技術構想すべき項目の全体像を見えるようにするとともに過去の技術構想書を集約した事例が検索可能な仕組みを用意している。

設計ドキュメントの充実を図るために、ドキュメント毎にドキュメントガイドを設け、記載すべき項目一覧や内容の統一を図っている。また過去に作成した設計ドキュメントの中からベストドキュメントを選別した事例集へのリンクが貼られている。

### ESPRとの整合表

開発プロセスを次の8つに分けてESPRとの整合表を準備している。

GSP1システム企画・分析、GSP2システム方式設計、GSP3ソフトウェア外部設計、GSP4ソフトウェア内部設計、GSP5プログラム製造、GSP6設計検証、GSP7評価、GSP8運用保守

### (2) 構成

G-SpiceEはA：開発プロセスガイドをトップページとし、B：ドキュメントフロー、C：アクティビティガイド、D：ドキュメントガイド、E：ソフトウェア技術構想展開マップガイドから構成されており、B：ドキュメントフローから各資料への参照が可能となっている（図3）。

### (3) 特徴

G-SpiceEはソフトウェア技術構想展開マップの活用を特徴としている。

組込み系ソフト開発には機械設計・電気設計・ソフト設計の連携が欠かせない。システム方式設計では装置全体への要求機能からユニット単位への機能に展開していく過程において機械設計・電気設計・ソフト設計の整合性をお互いに確認しておくことが重要である。ソフトウェア技術構想展開マップガイドを用意し、活用しやすくしている。

ソフトウェア技術構想展開マップの活用ポイント

ソフト設計に必要な機械設計・電気設計からのアウトプット資料の種類・納期を合意。

製品全体から見たソフト設計に必要な技術検討項目を抽出する。

過去の類似機種情報を参考に設計を行う。

ソフトウェア技術構想展開マップガイドの運用 (図4)

ソフトウェア技術構想展開マップガイドはA：ソフトウェア技術構想展開マップ、B：技術構想書実例集（全体構成・機能・機構）から構成されている。

A：ソフトウェア技術構想展開マップは横軸に大中小の3分類された技術検討項目一覧が用意してあり（例：小項目（分岐制御）は約150項目準備）、ソフト設計者は開発案件毎に技術検討項目一覧からテーマ規模・技術要素に応じた技術構想すべき項目の抽出を行い、テーマ毎のA1：技術構想マップを作成する。

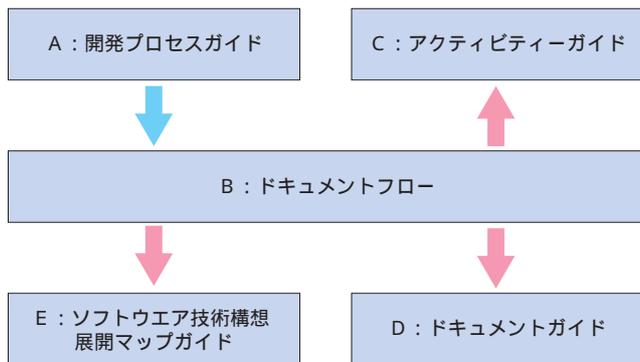


図3 G-SpiceE全体構成図

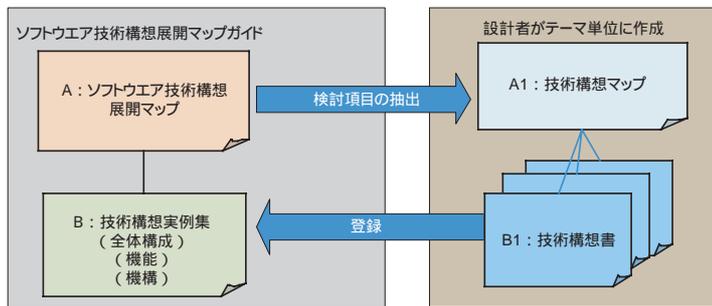


図4 ソフトウェア技術構想展開マップガイドの運用

その際、B：技術構想書実例集から技術検討項目毎に分類された技術構想書の実例を参照することもできる。

B：技術構想書実例集は検索可能となっており、テーマ完了時点でB1：技術構想書は見直しがかげられ技術資産としてB：技術構想書実例集へ登録される仕組みとなっている。

## 4 G-SpiceMの概要・構成・特徴

### (1) 概要

プロジェクトの成否は計画の良し悪しが決定するといわれている。G-SpiceMはESMRをベースにし、プロジェクト計画に重点を置いてマネージメントすべき全体像のなかからテーマ規模に応じた管理項目の選択を可能にしている。

利用される場面をまとめると表2のようになる。

#### ポイント

管理すべきマネージメント項目がソフトウェア作業項目計画（詳細は後述）で明らかにされている。

完了したテーマが整理され、マネージメント面で実施された過去のノウハウが参照可能となっている。

プロジェクトチェックシートが準備され、計画の時点で管理項目の漏れがチェックできるようになっている。

### (2) 構成

G-SpiceMはA：プロジェクトマネージメントガイドをトップページとし、B：プロジェクトマネージメントガイドマップ、C：解説ガイド、D：プロジェクトチェックシート、E：標準フォーマット、F：テーマ実例から構成されている（図5）。

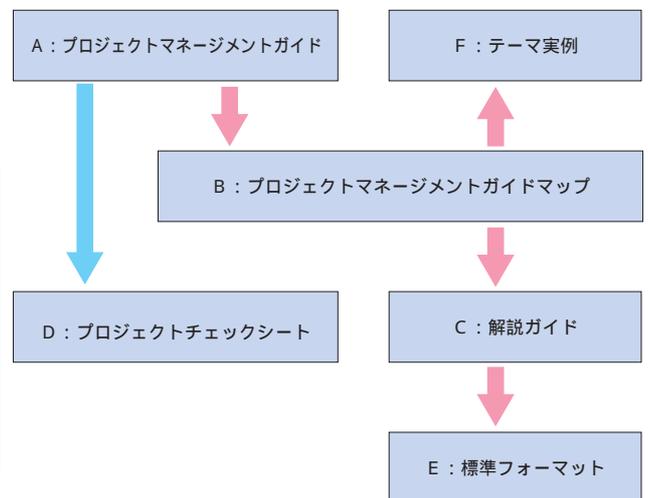


図5 G-SpiceM全体構成

たとえば、B：プロジェクトマネジメントガイドマップはプロジェクトマネジメントすべき、次の8つの知識体系から構成されている。

- GSM1 プロジェクトの概要
- GSM2体制
- GSM3リソース計画
- GSM4作業計画
- GSM5品質保証計画
- GSM6リスクマネジメント・課題管理
- GSM7設計委託管理
- GSM8プロジェクトデータの蓄積と有用活用

知識体系毎に解説ガイドが用意されており、過去プロジェクトの実例が参照できるようになっている。

### (3) 特徴

G-SpiceMはグローリーが認証取得しているISO 9001:2000との関連性を持たせている。テーマ規模に応じたマネジメントすべき必要十分な管理項目を明確にするために、テーマ開始時点でソフトウェア設計作業項目計画書兼実績報告書を用いて実施すべきプロセスや会議体・作成ドキュメント等を宣言し承認を受ける。承認を受けたソフトウェア設計作業項目計画書兼実績報告書は実績値が入力されテーマ完了時点までメンテナンスされる。なお、ソフトウェア設計作業項目計画書兼実績報告書は全てのテーマで作成され、ISOの品質記録となっている。

過去実施した開発テーマのQCD結果を踏まえ、製品品質を優先に考えると共に、コストを常に意識し、テーマ規模・難易度に合わせたバランスの良いプロセスとマネジメントすべき管理項目を選択することが重要である。

## 5 今後の取り組み

### (1) G-SpiceE / MのPDCAスパイラル

今回のG-SpiceE / Mは各部門ベスト設計ドキュメントへ

のリンクを大きな特徴としている。

実例が陳腐化しないように、より良い実例へ随時交換していくとともに、開発現場で発生している問題を顕在化させ、それをプロセスの改善に結び付けていくことが重要である。

- プロセス改善を定着・推進するために
- 経営層の意思が入った推進体制の継続
- 開発現場で発生している問題「見える化」の仕組み作り
- プロセス改善を含めた改善活動結果「見える化」の仕組み作り（実態の伴った定量的指標の設定と計測）

### (2) 効果の測定

今、G-SpiceE / Mが経営者から求められているのは、導入による効果の測定である。現状、品質は「不具合件数 / KLOC」等、生産性は「KLOC / 人月」等を利用して測定しているが、開発規模を表すKLOCが実態を反映していない場合もあり、実態に近い指標への見直しと、継続して指標の自動計測ができるシステム構築を進めている。

### (3) 三位一体の改善

ソフト開発のQCD向上は、プロセス改善、技術向上、人材育成の三要素をバランス良く考えることが必要であるといわれている。遅れていたプロセスの見直しについては、G-SpiceE / Mにより仕込みはほぼ終了したと考える。今後G-SpiceE / Mを真の成果に結び付けるために改善活動を継続するとともに、次は技術者の人材育成に焦点を当て、技術者教育体系としてG-SpiceSSの再整備を検討している。

## 6 おわりに

G-SpiceE / M構築を支援していただいた設計標準委員会メカトロニクス分科会の委員の方々に謝意を表します。

表2 G-SPiceMの利用目的

利用される場面	開発プロセスガイドの利用目的
プロジェクト計画・実施	マネジメントすべき管理項目を明確にし、プロジェクト計画の精度向上を図るとともに、プロジェクトチェックシートを利用して自己評価を行い管理項目の抜けをチェックする。
マネジメント知識体系の習得	マネジメントすべき管理項目およびマネジメント知識体系の習得を行う。
設計委託	設計委託時に管理項目や方法を明確にさせ、受入品質の向上を図る。
管理ドキュメントの利用	標準雛形を利用し、管理レベルの向上を図る。



# 専門職大学院組込み技術研究科の 設置と建学の精神



東海大学 総長・学長  
松前 達郎

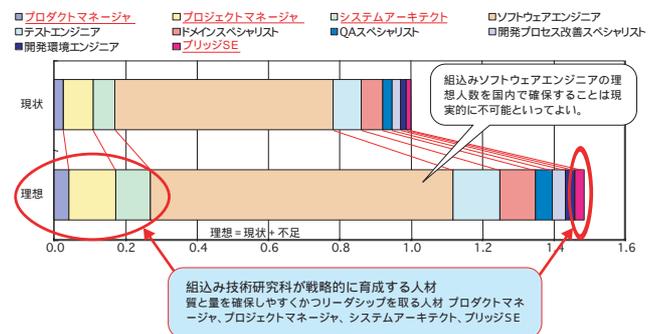
東海大学は、無装荷搬送ケーブル方式の生みの親である松前重義博士によって1942年に創設された。無装荷搬送ケーブル方式は、当時世界の主流であり常識であった装荷ケーブル方式の限界を超えて、電信電話の長距離多重化を可能にした世界的大発明であった。松前重義博士はこの発明に留まらず、日本が科学技術立国として伸びていくように、海外の特許は一切使わず、無装荷搬送ケーブル方式の技術を、製造を含めてすべて国産化させた。東海大学は創設から18年後の1960年、現在のFM東京の前身となるFM東海を実用化試験局としてスタートさせる等、松前重義博士のリーダーシップのもと、情報通信分野の技術開発と人材育成を大きな柱として取り組んできた。

東海大学のチャレンジ精神の背景には、若いうちに思想を培い、体躯を養い、知能を磨いて、明日の日本を作るべくチャレンジしていく松前重義博士の建学の精神がある。こうした東海大学の精神と、松前重義博士が唱えた科学技術立国という日本が進むべき道を考えるならば、製造業の競争力強化に大きく貢献する組込み技術の研究開発と人材育成に取り組むことは、本学にとって必然的なことであったといえる。

## 1 組込み技術研究科の狙い

経済産業省が行っている組込みソフトウェア産業実態調査（産業実態調査）によると、組込み技術者の不足は深刻である。さらに、日本経団連も指摘しているように組込み技術を含む高度IT技術者の不足は、国際的な産業競争力の低下を招くばかりでなく、我が国の安全保障にも影を落としかねない問題である。

一方、我が国はかつてどこの国も経験したことがない少



データは経済産業省 2006年版 組込みソフトウェア産業実態調査 経営者・事業責任者向け調査より

図1 戦略的人材育成による組込み技術者不足問題の解決策

子高齢化の世界に突入しつつあり、さらに若者の理工系離れも起きている。こうした状況における人材戦略は、量的拡大と確保ではなく、人材の質的向上が優先となる。そこで、本学の組込み技術研究科が育成対象とする人材と到達目標とする質的レベルを、組込みスキル標準（ETSS）と産業実態調査に基づいて決定した。すなわち、図1に示すように、ETSSのキャリアのうちプロダクトマネージャ、プロジェクトマネージャ、システムアーキテクト、ブリッジエンジニアを戦略的に育成することを狙いとして位置づけた。以下、これらのキャリアを戦略的キャリアと呼ぶ。

## 2 専門職大学院に組込み技術研究科を設置する意味

組込み技術者は製造業にとって必須の技術者である。科学技術立国を標榜する我が国において、こうした人材を輩出することは大学の社会的使命であるといつてよい。しかしながら、ETSSで定義するレベル2以上（一人前以上）の組込み技術者を育成するには、大学の資源のみでは極めて困難であり、人的資源を含めて企業との連携は不可欠であ

る。

こうした関係において、企業側からも教える人材や機材を提供して、大学と共同歩調をとって組み込み技術者を育成できるかといえば答はNOである。このような取り組みでは成果を期待することはできない。その理由は、産学が求める成果の方向が180度異なるからである。大学は教育研究の成果を論文として公開する論文数確保の方向にあり、企業は開発した成果を論文として公開するよりも知財化もしくは非公開とする方向にある。このように方向の異なる両者が協力できるテーマとなると、両者にとって魅力のないものになる可能性が高くなり、結局は期待する成果につながらないことになるのである。

こうした構造的矛盾を解決し、かつ両者にとって連携する魅力を増幅させること考えたとき、専門職大学院の仕組みを活用することが解を導き出すための有力な手掛かりとなる。専門職大学院は通常の大学院と異なり、修士論文を課さなくてよいという特徴から、学内での研究開発の成果を論文として公開する必要性はない。専門職大学院であれば、上述した産学連携を魅力あるものにし、かつ効果的に組み込み技術開発と実践的な組み込み技術者を育成できる可能性が高くなる。

以上のことから、次のように専門職大学院組み込み技術研究科組み込み技術専攻（以下、本研究科と略記）を設置した。

- 名 称： 東海大学専門職大学院組み込み技術研究科  
組み込み技術専攻
- 場 所： 港区高輪校舎
- 修士課程： 2年間
- 学 位： 和文 組み込み技術修士（専門職）  
英文 Master of Embedded Technology  
(Professional)
- 定 員： 1学年当たり30名（合計60名）
- 修士論文： 不要
- 修了単位： 30単位

なお、本研究科の時間割は、社会人も授業を受けやすいように、昼夜同じ科目を開講している。

### 3 組み込み技術研究科の特徴とSEC成果の活用

#### (1) 育成する組み込み技術者像

本研究科で育成する戦略的キャリアに共通する人材像は次の通りである。

設計要求された家電製品等の簡単な組み込みシステムの

内部構成を技術要素に分解でき、開発技術と管理技術を用いてV字工程に基づいて作業でき、要求を実現するまでの基本的な管理能力を備えている。

簡単な組み込みシステムを設計開発する小規模なプロジェクトチームを編成でき、各キャリアの振る舞いをシナリオとして記述することができる。

組み込みシステムを設計、開発、保守する工程に関し、PDCAを回し、問題点を抽出し、改善改良するための基礎的な開発能力を発揮することができる。

技術・科学に対する思想をもち、技術者としての倫理観をもって平和で豊かな社会の実現に貢献できる。

#### (2) カリキュラム開発へのETSSの応用

ETSSにおける戦略的キャリアは、いずれもキャリアレベル3以上となっている。これらの戦略的キャリアへアップしていく入口は、ソフトウェア開発やテスト等にある。こうした、キャリアアップのプロセスも踏まえて、組み込み技術専攻のカリキュラムを設計した。

以上の設計において、ETSSはカリキュラム設計者の共通基盤として利用でき、カリキュラムの妥当性を見る上でも極めて有効であった。

#### (3) カリキュラムの特徴

本研究科のカリキュラムは、ETSSに準拠して設計したものである。図2に示すように、科目区分の大きな柱は、ETSSの技術要素、開発技術、管理技術である。これに東海大学の建学の精神を継承する「現代文明論特論」と、企業との連携を推進しやすくし、かつ教育効果を上げる科目として「プロジェクト演習」を設けた。以下、各科目区分の特徴について説明する。

##### 専門基礎科目区分

基礎から組み込み技術の知識体系を修得し、さらに技術者

科目	必修・選択	単位	科目	必修・選択	単位
専門基礎科目			管理技術科目		
現代文明論特論	必修	2単位	技術戦略特論	選択	2単位
組み込みシステム構造特論	選択	2単位	製品戦略特論	選択	2単位
技術要素科目			総合教育科目		
組み込みシステム技術要素特論・特別演習	選択	3単位	プロジェクト定型特別演習	必修	2単位
ユーザビリティ特論	選択	2単位	プロジェクト基礎特別演習1	必修	2単位
組み込みネットワーク特論・特別演習	選択	3単位	プロジェクト基礎特別演習2	必修	2単位
デジタル信号処理特論・特別演習	選択	3単位	プロジェクト実践特別演習1	必修	2単位
組み込みOS特別演習	選択	2単位	プロジェクト実践特別演習2	必修	2単位
プロセッサ設計開発特別演習	選択	2単位	企業特別実習	選択	2単位
開発技術科目					
組み込みシステム開発技術特論・特別演習	選択	3単位			
ソフトウェア実装技術特別演習	選択	2単位			

図2 ETSSに基づく組み込み技術研究科のカリキュラム

としての倫理観や思想を醸成することを目的とする。

#### 技術要素科目区分

製品を構成する上で必要な組み込み技術の要素を理解し、基本的な技術要素は自ら設計及び利活用できるようになることを目的とする。

#### 開発技術科目区分

技術要素を開発あるいは応用する上で必要となる開発技術を修得し、実際の開発ができるようになることを目的とする。

#### 管理技術科目区分

開発対象となっている製品に必要となる組み込み技術資源に関する戦略立案と管理、組み込み技術資源に関する戦略の立案と管理を行うための基礎を修得し、マネジメントを実施できるようになることを目的とする。

#### 総合教育科目区分

各科目区分で個別に修得した技術を総合的に利用して、実際の製品開発に応用できる基礎を修得し、さらにその基礎を実際の製品開発や企業の現場で使うことで理論と現実の間のブリッジエンジニアとしての役割を果たせるようになることを目的とする。

## 4 カリキュラム実施に向けた設備と実施状況

### (1) 校舎の機能及び運用

図3は組み込み技術研究科専用に新たに建設した校舎である。この建物は24時間の使用に耐えるように、出入口、図書室、実験実習室、研究室等すべてオートロックとなっている。3階には産学官連携推進室を設け、本研究科に関係する各種会議や打合せなどに利用できる機能を備えている。



図3 専門職大学院組み込み技術研究科用に新築した校舎

### (2) カリキュラム実施と設備

図4は本研究科の建物内部の様子を示したものである。

同図(1)は扇風機の組み込みソフト開発用の設備である。扇風機の開発は、風を作るという物理現象を扱うため、組み込みソフトの実習には極めて有効な材料である。

同図(2)は、車のワイパーやライトの制御スイッチを読み込み制御する実験実習用の装置である。OSはOSECを使用し、ネットワークはCANを使用する。

同図(3)は通常のパソコンルームである。

同図(4)は無人で24時間対応できる図書室である。

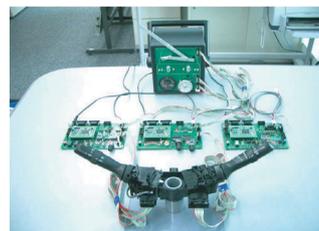
同図(5)はプロジェクト室であり、企業から委託されたプロジェクト等を実施する部屋である。学生にもこれと同じタイプの部屋を2部屋用意しており、24時間自由に利用できるようになっている。

同図(6)は談話室であり、これも学生が自由に利用できる空間である。

このように、本研究科の建物及び設備は、学生のモチベーションを高めると共に、企業との連携を図りやすくする



(1) 実験実習室



(2) 実験実習室



(3) パソコンルーム



(4) 図書館



(5) プロジェクト室



(6) 談話室

図4 組み込み技術研究科の各部屋の様子

機能を実現している。

### (3) 実施状況

カリキュラムと学生の指導内容について述べる。学生は16名在籍し、その内企業に勤務しながら通っている社会人学生は10名である。社会人は、夜間と土曜日に開校する授業に集中しているが、仕事の関係で欠席することもあるため、基本的にすべての授業をビデオに収め後からでも追いつけるようにしている。

### (4) 来年度へ向けた設備の充実

本年度後半から、社会人学生の入学申し込みや機材の提供の申し出が増えつつある。以下はその一例である。

図5は、日産自動車株式会社が本研究科向けに開発中のAFS<sup>1</sup>の実験実習用システムである。AFSは車の向きに応じて、前照灯の上下左右の角度を制御するものである。こ

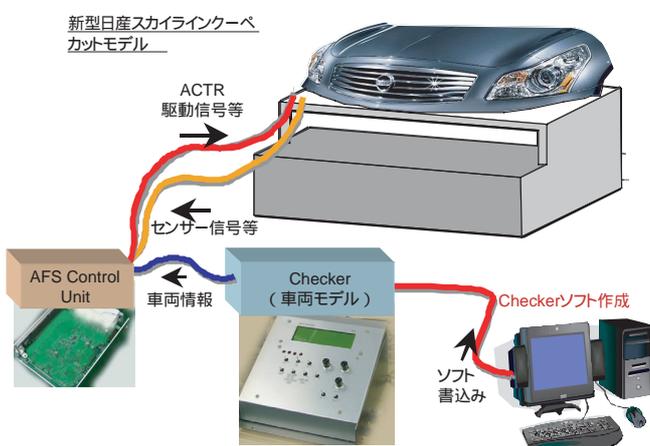


図5 AFS実験教育設備  
(協力：主たる開発環境は日産自動車、  
ソフト開発環境はルネサステクノロジー)

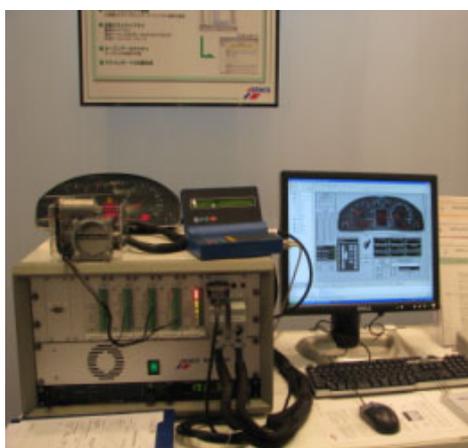


図6 エンジンECU用HILS  
(協力：dSPACE Japan)

のシステムのソフトウェア開発環境は株式会社ルネサステクノロジーが提供する。

図6は、車、航空機、プリンタなど組み込みソフト開発に実際の現場で使用しているHILSというシミュレータである。このシステムを使用することで、摩擦係数など様々な物理条件を設定してソフトウェアの論理的妥当性を検証することができる。図は車のエンジン制御用のECUのHILSである。このHILSについてもdSPACE Japan株式会社から提供していただき、来年度から実験実習用に活用することになる。

## 5 今後の取り組み

ETSSは産業界にも普及の兆しが高まっており、ETSSをベースとした人材育成、人材評価等が活発になりつつある。本研究科はETSSを実際に利用してカリキュラムを開発し、専門職大学院を運営しており、こうしたノウハウを企業へ提供できるものと考えている。そうした意味で、ETSSを共通フレームとして組み込み技術の人材育成手法やカリキュラム開発への協力等、学から産へ提供していくことも考えている。

## 謝 辞

まずは、本研究科の取り組みを高く評価していただき、SEC三周年の記念式典において、栄えあるベストプラクティス賞をいただいたことに感謝申し上げる次第である。また、本研究科の設置・運営に関しては、株式会社東陽テクニカ、株式会社コア、日本インター株式会社、株式会社ガイア・システム・ソリューション、日産自動車株式会社、株式会社ルネサステクノロジー、トヨタ自動車株式会社、株式会社東芝、日本電気株式会社、dSPACE Japan株式会社、日本経団連、組み込みシステム技術協会など多くの民間企業・団体の皆様からご支援をいただいた。また、IPA/SEC、経済産業省、文部科学省からも折に触れ適切なアドバイスやご協力をいただいた。この場をお借りして関係各位に感謝する次第である。

1 AFS : Adaptive Front-Lighting System、配光可変型前照灯システム



# トヨタにおけるソフトウェア開発プロセスの取り組み



トヨタ自動車株式会社 常務役員  
重松 崇

トヨタ自動車は、SEC設立検討時点から関係者として参画してきた。SECは、エンタープライズ系システム、組み込み系システムと実証的な先進ソフトウェアプロジェクトとの3領域を対象として取り組むことになった。前者2領域は、それぞれの課題が明確であったが、後者は具体的で実証的な事例が必ずしもあった訳ではなかった。

トヨタ自動車は、自動車端末と社会インフラとの協調で成立するシステム、すなわち組み込み系システムとエンタープライズ系とで成立するシステムの代表的事例であるITS<sup>1</sup>プラットフォームの構築という目標が、このプロジェクトでは望ましいとの判断で、JARI、インターネットITS推進協議会等の関係機関、自動車メーカ、電気メーカ、経済産業省等と意見調整を進めた。この中で、ITSシステムでは、時間と場所（位置）との時空間データが基本情報であるとの認識に基づいて、この時空間データを処理するプラットフォームソフトウェア、特に馴染みやすい交通情報を扱うことで合意できた。

この段階で、実証的な先進ソフトウェアプロジェクトの開発スコープとして「ITS各システム間の互換性を確保し、低コスト、高品質、高信頼、高性能で、機能拡張性にも優れた交通情報生成を代表例とするプローブ情報プラットフォームソフトウェア」を開発することになった。

次に、トヨタ自動車は、このプラットフォームソフトウェアを開発する体制をどのように構築するかについて、前述の関係機関、経済産業省（製造産業局 自動車課、商務情報政策局 情報処理振興課）と意見交換し、開発のスタートであるソフトウェア仕様書（要件定義書）を作成する組織と、それに基づいてソフトウェアを開発・実装する組織

とを立ち上げることを推進した。前者は、経済産業省 自動車課がJARIに委託し、官民学でソフトウェア開発組織の立ち上げ前に結成された。後者は、経済産業省 情報処理振興課が、ソフトウェア仕様書に基づいて開発するCOSE<sup>2</sup>に委託・開発する基本スキームを実現した。すなわち、要件定義書の作成からプラットフォームソフトウェア開発というソフトウェア開発プロセスが構築された。

## 1 COSEプロジェクト

### (1) 体制

COSEは、NTTデータ、デンソー、トヨタ自動車、日本電気、日立製作所、富士通、松下電器産業の7社（五十音順）の異業種ベンダが参加し、実用化に耐えうるプラットフォームソフトウェアの開発を行っている。また、このプロジェクトの狙いである「マルチベンダ環境で、いかに効率的かつ高品質なソフトウェア開発を進めるか」という実用的なプラットフォームソフトウェア開発とは異なるソフトウェア・エンジニアリングの部分については、EASE（ソフトウェア工学へのエンピリカルアプローチ）プロジェクトを推進している奈良先端科学技術大学院大学、大阪大学、SEC、経済産業省が学官の立場で参画した。このようにして、COSEプロジェクトの産・学・官での推進体制ができあがった。

トヨタ自動車は、実用化に耐えうるプラットフォームソフトウェアを利用するユーザの立場で、このプロジェクトに参画した。特に、推進上でのCOSEプロジェクトの特徴である以下の3点に関わる領域には、積極的に参画し意見提

1 ITS : Intelligent Transport Systems

2 COSE : Consortium for Software Engineering

案を行った。

## (2) 特徴

COSEプロジェクトの特徴は、次の3点である。

技術委員会、SE部会、評価WG（最終年度には、要件定義タスクフォース、公開デモ実験検討WG）による水平分散型組織

従来のソフトウェア開発は、発注側と受注側との垂直統合型が主流であった。COSEでは、組合参加各社の技術の集合としてプラットフォームを開発するとの基本方針に基づき、技術委員会は「全体の計画検討」と「実用化に耐え得るプラットフォームソフトウェアの仕様検討等の戦略的検討」を組合参加各社が水平分散型で、目標の共有化・情報共有化を図りながら進めた。SE部会は、プロジェクトマネージャを中心にソフトウェア開発全体を管理すると共にソフトウェア・エンジニアリングの実践を行い、その進捗及び課題、開発の方向性等を技術委員会に報告・提案した。評価WGは、利用者の立場から、要件定義書で未規定の内容を含めてソフトウェアの基本設計書、詳細設計書、ソースコードレベルそれぞれについて、ソフトウェアの品質、性能等を実用化に耐えうるか否かの視点からチェックし、それぞれのレベルにフィードバックすると共に、技術委員会に報告・提案し要件定義書の見直しを進めた。

V字モデルの開発プロセスによる効率的、高品質、高性能なソフトウェア開発

ここでの注目すべき事項は、2年数ヶ月間にV字モデル開発プロセスを2度回した点と、SE部会を中心にして進められたソフトウェア・エンジニアリングの実践によりソフトウェアモジュールが構築できた点である。V字モデルの2サイクル化は、当初策定された要件定義書の未規定部分の明確化と充実化及び実用化に向けての検討項目の洗い出しに大いに貢献した。ソフトウェア・エンジニアリングの実践では、EPM<sup>3</sup>ツールの有効性検証、クローン分析によるソフトウェアモジュールの抽出とモジュール化等が実施できた。

プローブ情報プラットフォーム開発プロジェクト推進タスクフォースと客観的評価（学識経験者による評価）チームの立ち上げ

前者は、より広く利活用側からのニーズの吸い上げと組合のスコープの確認に有効であり、後者は、生成された情報の信頼度等、学問的意義の検証を通して、研究組合としてのオープン性が確保されたことが大きな成果であった。

## (3) 成果

以上記述したように、実用化に耐えうる高品質、高性能、高信頼、実用化の投資コストの抑制と、今後の技術革新・機能拡張性に対応できるプローブ情報プラットフォームソフトウェア開発ができた。この中で、ソフトウェア要件定義書の完成度向上（今後の課題を含め）とソフトウェア・エンジニアリングの実践の有効性が確認できた。

このプロジェクトで抽出された時空間データの互換性を確保するための地図の互換性確保、プローブ情報プラットフォームで抽出された渋滞原因の特定等の課題については、今後、別のプロジェクト（例えば、情報大航海プロジェクト等）に引き継がれ、開発されていくことが期待される。

## 2 ETSSを導入した人材育成

以上記述した取り組みに加え、SECの成果である組込みスキル標準（ETSS）を導入した人材育成へのPDCAサイクルの仕組みづくりをトヨタグループ内で推進している。SECの協力の下、ETSSによるスキル診断の取り組みを2006年度から開始した。

### (1) スキルの見える化

「育成の成功事例作り」、「現場に負担をかけない人材育成プロセスの構築」を活動方針とし、2006年度はスキルの「見える化」（定量化・数値化）を実施。スキル診断実施にあたって、標準フォーマットに組込み業界標準であるETSSを採用した。ETSS採用によって、

- ・スキル標準を体系化
- ・スキルを定量化（数値化）

の2点を実現することが、トヨタグループのニーズに合っていると考えた。

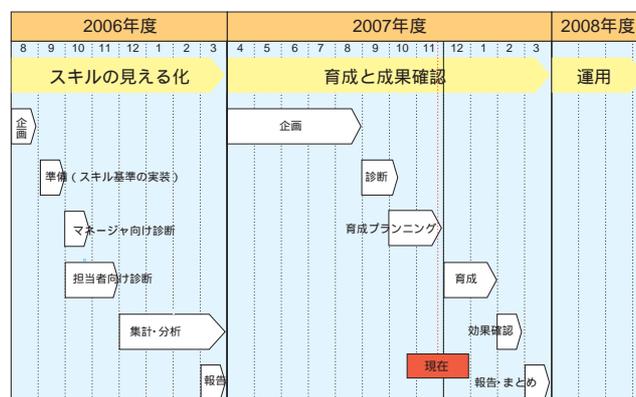


図1 PDCAサイクル確立に向けたステップ

## 2006年度の活動

対象者を管理者と担当者に分け、スキル診断を実施。管理者と担当者を区別することで、下記のメリットがあると考えた。

- (a) 管理者向け診断 (30項目)
- (b) 担当者向け診断 (約200項目)

診断結果の見える化(定量化・数値化)を図るために、

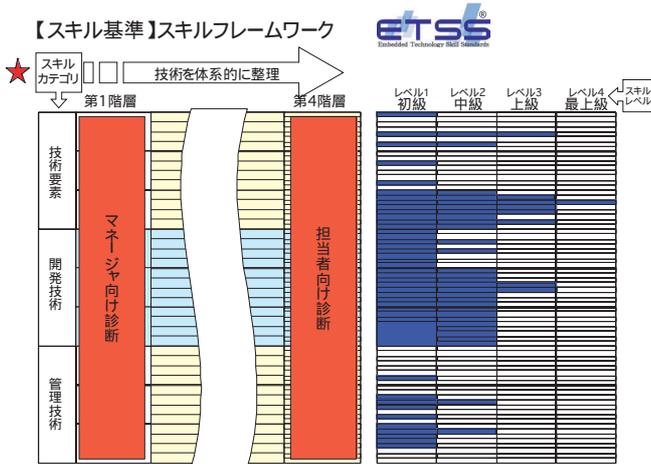


図2 スキルフレームワーク

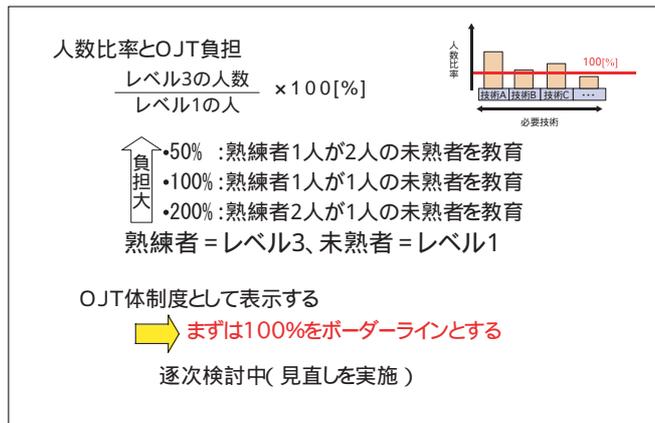


図3 レベル1とレベル3の人数比率の見方

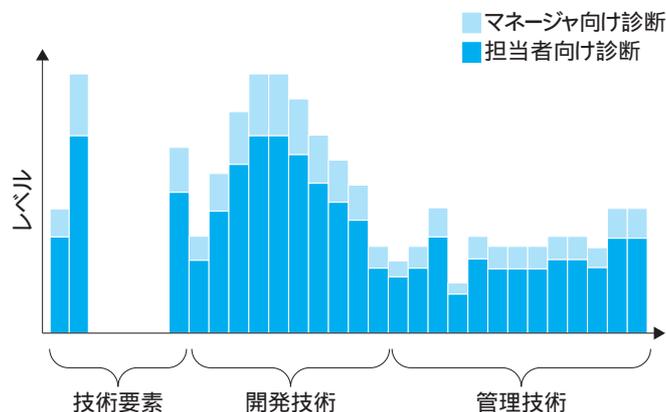


図4 見える化結果の比較

下記事項を実施した。

- (a) スキルプロファイルで集計
- (b) レベル1とレベル3の人数比率を集計

図4のグラフは「スキルの見える化」で得たデータを、管理者と作業者とで比較したものである。

「管理者向け診断の粒度(30項目)で十分」、「問題領域に関係する担当者までは抽出可能」、「活動指針に合った診断方法が選択できない」等のフィードバックが出され、スキル診断の枠組みが完成した。

## (2) 育成と成果確認

2007年に行われたスキル診断では、選定した対象者1人ひとりについてリーダーが診断を実施した。重点テーマに沿って数項目まで絞り込んでから記入し、ピンポイントでレベル判定理由を明記した。プランニングは育成対象者のスキル診断結果から育成メニューを策定し「マイクロトレーニング」を実施した。「現場に負担をかけない育成」、「現場で達成感を実感」、「ピンポイントに対応」、「育成のPDCAサイクルの完結」を活動方針とした。

マイクロトレーニングの実施後、

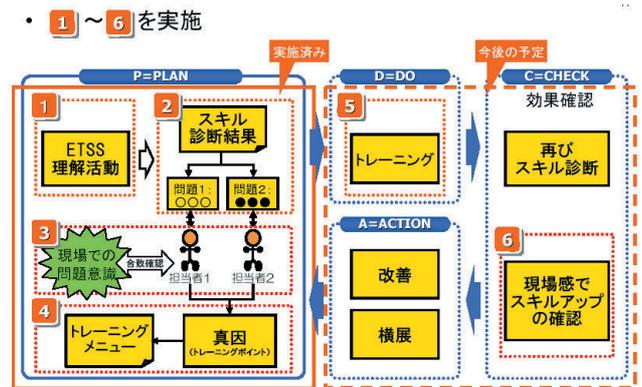


図5 2007年度実施範囲

・ ①～③の詳細

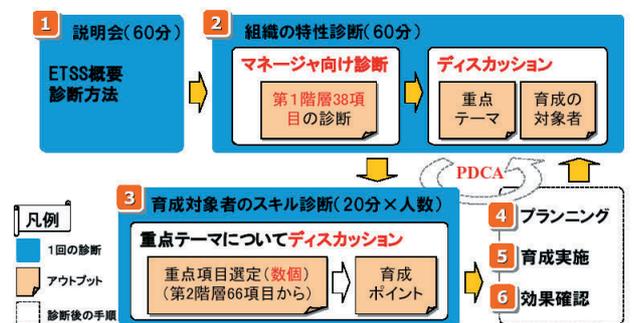


図6 スキル診断

- ・トレーニング後の期待効果が現場で想像できない
  - ・トレーニング受講のモチベーションがあがらない
  - ・トレーニング内容の説明が業務用語でない
  - ・現場は「ユースケース」という言葉を使用しない
- 等の問題点を抽出し、下記の対策を検討した。
- ・業務分析を行い、現場の用語に翻訳

- ・現場作業に近いモデル作業を提示
- ・「ができるようになる」を具体的に表現

### (3) 課題と今後の進め方

PDCAを1サイクル回してみた結果、以下の課題が出てきたので、改善を検討中である。

#### <現状の課題>

- ・一般の技術用語をトヨタグループ内で使用されている現場用語に翻訳
  - ・現場感覚でレベルアップを評価できる手法の企画、現場の作業をモデル化
  - ・確立したPDCAサイクルを全体に適用
  - ・ドメイン非依存の部分については、他の制御システム分野に展開
- また、ETSS導入については、
- ・担当者向けスキル診断の再実装
  - ・項目粒度の改善が必要
  - ・キャリア基準の導入検討

等の課題があり、今後は、人材マップ及び人材育成プランにフィードバックをかけ、実用化に向けて検討を進めていきたい。

## 3 おわりに

COSEプロジェクトでは、産官学連携によるプロープ情報プラットフォーム開発を通して、プロセス・エンピリカル技術の効果有効性検討が実施できた。また、技術者スキル診断については、見える化、育成効果確認フェーズを経て、今後は人材マップ及び人材育成プランにフィードバックする等、実用化に向けた活動を展開していく所存である。

最後に、関係機関ならびにプロジェクトにご参加いただいた関係者の方々のご協力により今回の受賞となった。この場をお借りして深く御礼申し上げます。

図7 マイクロトレーニング (スキル診断・ヒアリングとの相関)

ID	名称	概要	教育レベル	スキルカテゴリ	実施形態	標準時間	備考	人	機	料	費	注
82-03	MEMSコア単体 入門コース	MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理	1級	基礎知識	講義・演習	30分						
82-06	MEMSコア単体の構造と動作原理 (MEMSコア単体の構造と動作原理)	MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理	1級	基礎知識	講義・演習	30分						
82-08	MEMSコア単体の構造と動作原理 (MEMSコア単体の構造と動作原理)	MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理	1級	基礎知識	講義・演習	30分						
82-11	MEMSコア単体の構造と動作原理 (MEMSコア単体の構造と動作原理)	MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理	1級	基礎知識	講義・演習	30分						
82-12	MEMSコア単体の構造と動作原理 (MEMSコア単体の構造と動作原理)	MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理	1級	基礎知識	講義・演習	30分						
82-13	MEMSコア単体の構造と動作原理 (MEMSコア単体の構造と動作原理)	MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理	1級	基礎知識	講義・演習	30分						
82-14	MEMSコア単体の構造と動作原理 (MEMSコア単体の構造と動作原理)	MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理	1級	基礎知識	講義・演習	30分						
82-15	MEMSコア単体の構造と動作原理 (MEMSコア単体の構造と動作原理)	MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理	1級	基礎知識	講義・演習	30分						
82-16	MEMSコア単体の構造と動作原理 (MEMSコア単体の構造と動作原理)	MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理	1級	基礎知識	講義・演習	30分						
82-17	MEMSコア単体の構造と動作原理 (MEMSコア単体の構造と動作原理)	MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理	1級	基礎知識	講義・演習	30分						

図8 マイクロトレーニング (スキル診断結果に基づく育成項目のマッピング)

科目名称	マイクロトレーニング	No	MZ-08
科目の教育レベル	<input type="checkbox"/> 初級 <input type="checkbox"/> 中級 <input type="checkbox"/> 上級 <input type="checkbox"/> 基本級 <input type="checkbox"/> パーナル <input type="checkbox"/> ビジネス		
スキルカテゴリ	<input type="checkbox"/> 技術系 <input type="checkbox"/> 開発系 <input type="checkbox"/> 管理系 <input type="checkbox"/> パーナル <input type="checkbox"/> ビジネス		
概要	本研修は、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理について、10 講の演習課題を解けば、実用に必要なスキルを習得できる。		
受講対象者 (教育対象)	MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理。		
履修条件	MEMSコア単体の構造と動作原理 (クラス、インスタンス、継承)を理解していること。MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理が使えること。		
教育目標	MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理、MEMSコア単体の構造と動作原理。		
実施形態	<input type="checkbox"/> OBT <input type="checkbox"/> 遠隔教育 <input type="checkbox"/> 講義・演習 <input type="checkbox"/> ワークショップ <input type="checkbox"/> 実機演習 <input type="checkbox"/> OJT		
標準時間	6時間(3時間×2回)	開催日程	
教材	テキスト、ワークブック、ハンドブック、ソフトウェア、その他		
教育成果の評価方法	<input type="checkbox"/> 事前試験 <input type="checkbox"/> 事後試験 <input type="checkbox"/> 口頭試験 <input type="checkbox"/> 認定試験の実績		

図9 マイクロトレーニング(トレーニングのシラバス)



ラムを構築。2005年度からの組み込み技術者研修カリキュラム開発に際してはSECの提唱するETSSをベースにカリキュラムの検討を行い、開発委員会への参加もいただきETSSに対応した研修内容を構築した。

### 3 ETSSキャリアフレームと組み込み技術者研修の対応

カリキュラム開発に際し、ETSSのキャリアフレームに準拠した形をとることとし、ソフトウェアエンジニアを基準職種として位置づけ、合わせてテストエンジニアとしての責任も明確にした。さらに、キャリアアップする毎に専門職種の役割を入れていき、本研修の最終レベルをミドルレベルのレベル4とし、プロジェクトマネージャ、システムアーキテクト、QAスペシャリストを目指す基礎知識を持つことを目指した。

職種 / 専門分野とキャリアレベル (出典:IPA / SEC 組み込みスキル標準キャリア基準)

職種	プログラクマネージャ	プロジェクトマネージャ	ドメインベシヤリスト	システムアーキテクト	ソフトウェアエンジニア	アプリ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
ハイ	7	6	5	4	3	2	1	1	1	1
ミドル	4	3	2	1	1	1	1	1	1	1
エントリ	2	1	1	1	1	1	1	1	1	1

注: 図中の黄色い矢印は「中級コース」(レベル3-4)、赤い矢印は「実践コース」(レベル2-3)、緑の矢印は「基礎コース」(レベル1-2)、赤い矢印は「入門コース」(レベル1)を示している。

図2 ETSSキャリアフレームワークと研修の対応

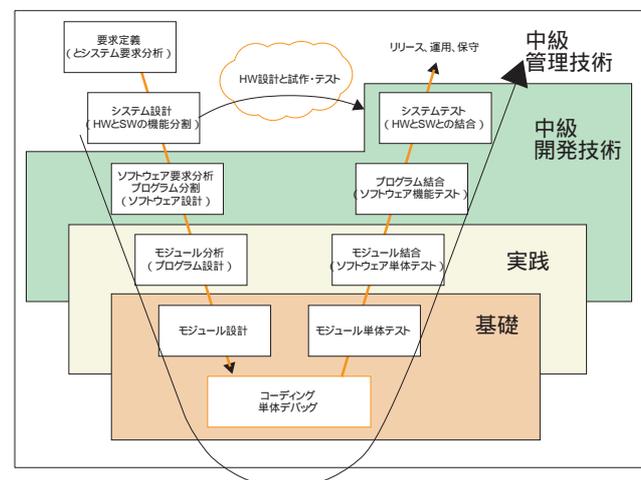


図3 組み込みソフトウェア技術者研修 コース体系

## 4 NICOにおける組み込み技術者研修の内容

### (1) 研修の特長

NICOにおける研修の最終目的は開発現場で仕事ができるようになることとし、そのために「わかる」から「できる」へ、さらに「うまくできる」、「仕事で使える」を目指すものである。中級クラス修了者へは「指導できる」も大いに期待している。

特に心がけたことは、V字モデルの開発プロセスの特徴を理解し、開発の流れと各レベルの役割を認識して、とかく独りよがりになりがちな組み込みソフトウェア開発にプロジェクト管理という考え方を共有することとした。適切な役割分担と各工程での設計文書の必要性の認識、フォーマットの共通化等を通して開発方式を共有し、自社内あるいは企業間協業での開発の効率化と品質向上を狙っている。

さらに、演習を重視し、座学で学んだ知識を繰り返し実際に使ってみて身に付ける方法で、「仕事で使える」を目指した。その究極が各コースの混成チームによるPBL (Project Based Learning) と呼ばれる、実務に即した役割での模擬プロジェクトの体験学習である。この開発演習コースは1日8時間で延べ10日間行われるもので、ハードではあるが受講者の満足度も高く、研修の目玉コースとなっている。

また、研修会場の長岡工業高等専門学校は高専対抗ロボ

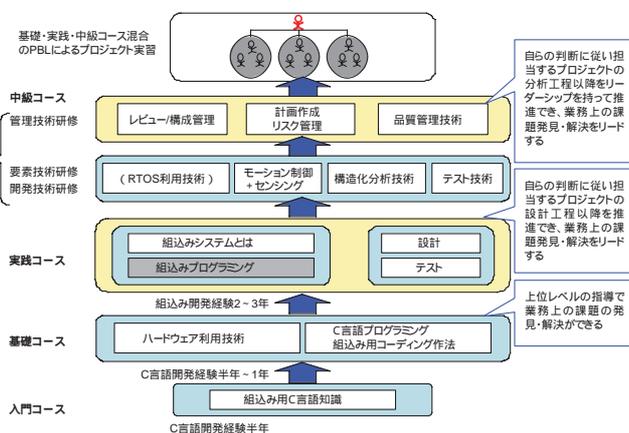


図4 研修コースの構成

表1 NICOの組み込みソフトウェア技術者育成コース例

KM1	組み込みソフト開発 基礎コース: 基礎的な組み込みソフトウェア技術者の育成(6日間)
目標	モジュール設計・実装・テストを中心に演習も交えて組み込みプログラミング技術を学び、ハードウェアの基礎知識も習得することで組み込みソフトウェア開発の基礎を理解する。開発の流れとプログラムの役割を理解し、上位技術者の指導の下でプログラミングができることを目指す。
対象	組み込みソフトウェア技術者としての基礎を固めたい方。業務系からの参加可。
前提知識	C言語プログラミングの基礎知識を身につけており、実際にコーディング経験があること。
内容	1.C言語の組み込みプログラミング作法 5.ハードウェアの基礎知識 (ピンタ、構造体、ビット演算など) 6.オシロスコープ演習 2.組み込みプログラミング基礎Ⅰ 7.組み込みソフトウェアの品質と開発プロセス 3.組み込みプログラミング基礎Ⅱ 8.チーム形式でのプログラミング演習(2日間) 4.組み込みプログラミング演習
KM2	組み込みソフト開発 実践コース: 実践的な組み込みソフトウェア技術者の育成(5日間)
目標	組み込みソフトウェア技術を演習を中心に学び、分析・設計・実装・テストの実際を理解する。独力で設計・実装・テストの業務を遂行できる判断力を付けることで、上流工程の役割とプロジェクト管理に関する知識を習得し、仕事の見積もりができ、複数人のチームで開発設計作業が実施できる能力習得を目指す。
対象	KM1修了者、または基礎コース(KM1)相当の知識と経験があり組み込みソフトウェア技術者として一人立ちしたい方。
前提知識	組み込みソフトウェア開発について詳細設計、プログラミング、テスト環境設定などの基礎技術があること。
内容	1.組み込み向け構造化分析・設計の概要(講義と演習) 5.プログラミング実習 2.開発課題と失敗事例の解説 6.ソフトウェアテストの概要(講義と演習) 3.組み込みプログラムの構造 7.組み込みソフトウェア開発管理の基礎 4.ハードウェアの特性 8.チーム形式でのソフトウェア設計・テスト設計実習(2日間)
KM3	組み込みソフト開発 中級/プロジェクト管理コース: 組み込みソフトウェア開発プロジェクトのリーダー育成(2日間)
目標	組み込みソフトウェアプロジェクトにおけるリスク管理・工程設計・品質管理・構成管理などプロジェクト管理の基礎技術を一部演習を交えながら学ぶことで、自らの判断に使い担当するプロジェクトを推進できる知識を習得する。業務上の課題発見・解決をリードするプロジェクトリーダーを目指す。
対象	組み込みソフトウェア技術者として一人立ちしており、リーダーとしてプロジェクトをまとめる役割を担うことを期待されている方。
前提知識	組み込みソフトウェア開発(要求分析から設計、テスト工程まで)で自立したスキルがあること。
内容	1.プロジェクトマネージメントの概要 4.設計・コードインスペクションのポイント 2.プロジェクト計画立案・運用とリスク管理の概要 5.構成管理の詳細とその実践方法 3.サンプルプロジェクトを基にしたPJ管理のグループワーク 6.組み込みソフトウェアの品質管理理論と演習
KM4	組み込みソフト開発 中級/開発技術コース: 組み込みソフトウェア開発の中核技術者の育成(2日間)
目標	構造化分析、リアルタイムソフトウェアでのテスト技術を学ぶことで品質設計に根ざした開発技術を身に付け、プロジェクト内での技術リーダーを目指す。さらに実習に必要なモータ制御やセンシング、データ処理といった技術要素を学ぶ。
対象	組み込みソフトウェア技術者として一人立ちしており、プロジェクトの技術的リーダーを目指す方。
前提知識	KM2修了者、または構造化分析・設計、テスト設計について実践コース(KM2)相当の知識と経験がある。
内容	1.構造化手法による要求分析 4.リアルタイムソフトウェアのテスト(講義と演習) 2.要求分析の演習 5.プロジェクト実習(KM6)にむけた要素技術解説(音声処理) 3.マイコンドライバの作成作法 6.プロジェクト実習(KM6)にむけた要素技術解説(モータ制御)
KM5	組み込みソフト開発 中級/要素技術(RTOS)コース: 組み込みソフトウェア開発の中核技術者の育成(2日間)
目標	組み込みシステムのプラットフォームとして最近使用機会が多くなっているリアルタイムOS(RTOS)の基本を学び、システム設計におけるRTOSの有用な使用方法を理解する。
対象	組み込みソフトウェア技術者として一人立ちしており、プロジェクトの技術的リーダーを目指す方。
前提知識	KM2修了者、または組み込みC言語プログラミングとテスト設計について実践コース(KM2)相当の知識と経験をお持ちの方。
内容	1.リアルタイムOSの基礎 4.ITRON同期・通信機能プログラミング実習 2.ITRON基本プログラミング実習 5.ITRONベースでのリアルタイムソフトウェア実装実習 3.リアルタイムOSの同期・通信機能 (RTOSとしてTOPPERS/JSPカーネルを使用します)
KM6	組み込みソフト開発 プロジェクト実習コース: 組み込みソフトウェア開発技術者合同のプロジェクト実習(10日間)
目標	PBL(プロジェクトベース・ラーニング)手法での役割分担に基づいた実践的な開発を体験する。参加者自身で①プロジェクト推進のための計画と必要な技術を設定し、②開発の実践を通じ技術の習得を目指す、③問題解決に立ち向かうことで、実際のプロジェクト開発に必要な能力を得ることを目指す。
対象	リーダー担当: 中級プロジェクト管理コース(KM3)と中級技術(KM4,KM5)修了者 サブリーダー担当: 中級技術(KM4,KM5)修了者 実務担当: 基礎コース(KM1)修了者
前提知識	対象者としてのそれぞれのコースにおける知識を身につけている。
内容	1.モジュール設計 2.プログラミング 3.モジュール、単体テスト設計とテスト実施 4.システムテスト実施

(2) 各コースの概要

本研修の具体的なカリキュラムは SESSAME<sup>2)</sup>のテキストをベースに実習内容を独自に作成し、プロジェクトで実際に使用するドキュメント類の雛形も提示して、実践に即した内容としたものである。

(3) PBLの内容

PBLは、スキルレベルの異なる人員からなる組み込みソフト開発模擬プロジェクトを形成し、実務に即した役割での開発実習を通して、実際のプロジェクト開発に必要な能力を得ることを目指すものである。具体的には1チーム6名で日数は10日間(約2ヶ月間)、その間のチームメンバー内のメール等による時間外作業も発生するため密度の濃いものとなっている。目指すものはチーム自ら見積もり、計画し、分析・設計・プログラミングといった作業を計画に沿って遂行するプロジェクト管理遂行能力と、コミュニケーションの重要性を認識し、上流工程での品質の作り込み、下流工程での品質の検証という品質管理能力を身につけることである。

コンの強豪校として知られるが、本年度からはNICOと共催の形で協力をお願いしている。実習教材の製作や、オシロスコープ等の機材提供や先生方に講師として参加いただく等、研修に深くかかわり、研修拠点として最高の環境を提供していただいている。

教材は長岡工業高等専門学校の先生方から協力いただき、市販の自走ロボットにマイクを取り付けた音声回路ボードを作成して、音声(拍手)によりリモート運転を行う機能を搭載し、最終的にはチーム単位でのタイムトライアルを行って機能を競うという内容で実習を行った。

音声(拍手音)によるリモート運転を行う  
搭載の反射センサーで障害物を検知し自律回避する

要素技術

- ・制御 モータPWM制御
- ・センシング 反射センサ(A/D)、マイク(A/D)
- ・データ処理 A/Dからのデータ処理
- ・割り込みベースの非同期処理またはRTOS利用(チームで選択)

開発技術

- ・構造化分析、構造化設計に基づく実装
- ・テスト技術

管理技術

- ・計画とレビュー、リスク管理
- ・サブチーム間のコラボレーション
- ・品質管理

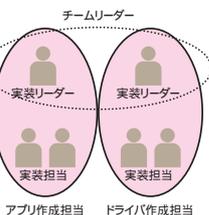


図5 PBLの内容

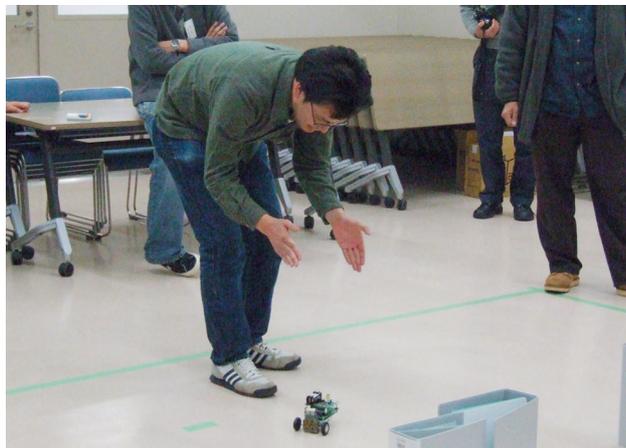


図6 PBLレースの様子

2 SESSAME : NPO法人組み込みソフトウェア管理者・技術者育成研究会

コース		2006年	2007年
中級	PBL	24人(11社)	17人(10社)
	RTOS		14人( 8社)
	開発技術		11人( 6社)
	管理技術		13人( 7社)
実践		20人( 9社)	23人(16社)
基礎		23人(15社)	33人(20社)
入門		14人( 5社)	
合計		79人(18社)	111人(28社)

2007年  
業種別  
受講状況

事業区分	企業数	基礎	実践	管理	開発	RTOS	PBL	合計
SW開発(業務系)	6	5	4	1	0	0	2	12
SW開発(組込み系)	8	14	6	11	8	9	6	54
電子/機械製造業	14	14	13	1	3	5	9	45
総計	28	33	23	13	11	14	17	111

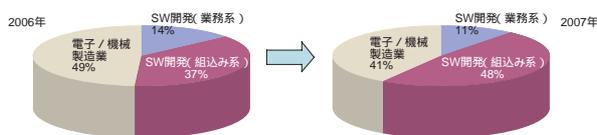


図7 組込み研修参加状況

## 5 研修参加状況とその考察

2006年から始まった本研修は当初の予想を上回って好評に推移しており、今年度は111名の参加を見ている。コース別、業種別の詳細は図7の通りであるが、当初自前での開発が少ないのではと危惧していた製造業からの参加が昨年、今年とほぼ同数の40名程度あり、自社の製品の付加価値を高める努力が窺えて「ITを競争力の源泉とした産業群の形成を促進」という、システム開発産業群創出活動の基盤事業としての役割を果たし始めていると判断できる。また、IT系企業でも業務系SEからの参加が各年10%を超えており、ITゼネコン下請体質と揶揄され続けてきた本県IT企業が、製造業のパートナーとして製品開発にも関わる付加価値追求型に変身する姿が見えてきた。

さらに、参加企業同士のマッチングが進行しており、県内製造業とIT企業の連携及びIT企業間での協業が昨年の研修終了直後から進み、昨年度の受講企業が本研修の受講を地場協力企業に薦めるケースが目立っている。ヒアリングで確認したところ、開発プロセスの共通化と標準フォーマットでのドキュメント利用による効率化を挙げる企業が多かった。これも当初の目論見通りであり、我々が予想したよりも早く、初年度から企業間連携の動きが促進され、事業成果が直ちに目に見える形で現れてきたことは大いに評価できることである。

## 6 今後の取り組み

### (1) ETSS研究会の開催

研修事業が立ち上がりつつあることを受けて、参加企業からETSSの企業内での運用を計画する動きが出始め、NICOがこの動きをまとめて「新潟県ETSS研究会」を2007年10月に立ち上げた。ETSSを実際に人材育成、コンピテンシ管理、外注スキル管理に「使える」ようにするための研究会で、第1段階として従業員がスキルレベルを判断する基準の明確化・共通化をテーマに活動することになった。この成果は、以下の内容で活用していく予定である。

- ・個人スキル、企業スキルの客観評価
- ・企業の強みを明示するツールの整備と、それによるビジネスマッチングの促進

### (2) 講師の確保と研修の継続

人材研修は継続こそが力である。これまではSESSAMEからカリキュラム作成を指導していただくと同時に講師についてもテーマ別に適任者の派遣をお願いしていたが、いつまでも好意に甘え続けるわけにもいかず、今後は自前で講師を育成する必要がある。まずは基礎、実践及びPBLコースについては本研修修業者にサブ講師を務めてもらう体制を固め、一部テーマについてもメイン講師を選定し、無理のない範囲でSESSAMEにご協力いただき、研修結果や課題等を共有することで連携するメリットを探っていく。

また同様に、IPA/SECに研修内容と結果、ETSS活用実績を報告することで、今後の活動のアドバイスをいただき、研修の内容のレベルアップを図っていく。長岡工業高等専門学校とは今後とも研修の共催を継続し、研修内容を高専教育にも生かしていただけるような検討を進めていく。具体的には基礎コースの一部についてIT教育カリキュラムへの採用がなされており、今後、上位学年の学生をPBLへ参加することでインターンシップの役割を持たせることも計画している。参加企業に対しては技術者のレベルアップが企業の業績にリンクしてくるよう、企業マッチングをNICOが積極的に進めることで参加の継続と増加を目指し、人材育成と県内企業の付加価値向上が図られる体制を構築していきたい。



# テストアセスメントによる 品質向上への取り組み



日本ユニシス株式会社 品質保証部  
飯田 志津夫

情報システムの信頼性を確保し向上させるため、検証と妥当性確認（V&V）の活動であるテストの重要性が増している。日本ユニシス株式会社品質保証部では、社内で蓄積したデータだけでなく、定量データ分析[SECADATA]等の公開情報も活用し、プロセスとプロダクトの品質を客観的に評価している。この定量的品質管理の根幹である品質メトリクスの信頼性を確保するため、現地・現物調査及び、調査の一環としてテスト専門技術者によるテストアセスメントを実施している。本稿では、テスト設計成果物のアセスメントと、その結果に基づくプロジェクト支援を通じたテスト品質の向上と要員育成に関する取り組みについて紹介する。

## 1 品質管理の現状と テストアセスメントの必要性

品質管理では、代用特性である障害検出率等の品質メトリクスを用いて品質の良否を判断する。品質管理が小規模プロジェクトまで徹底し、品質を定量的データで語るようになってきたが、最初にメトリクスを見て安心し、現物（成果物）を見ないで品質を判断してしまう弊害が、一部のプロジェクトで出てきた。このように真の品質特性である稼働後のお客様の要求の充足度と、代用特性である品質メトリクスとの間の強い相関性が確認されていないと、品質メトリクスを用いた品質管理の意味が失われてしまうことになる。

また、テストによる検証と妥当性確認の成否は、テストケースの品質に依存する。収集・分析・評価される品質メトリクスの信頼性が確保されないと、上位管理者による管理レビューや品質保証レビューの実効性が担保できない。このため、プロジェクトから独立した第三者として、品質

保証部のテスト専門技術者が具体的なテスト内容を客観的に評価し、テスト品質の向上を支援する「テストアセスメント」を実施している。

## 2 テストアセスメントの 概要

テストアセスメントの実施に際しては、留意すべき点が三つある。アセスメント候補の選定、入力とする要求仕様書・設計書の品質の見極め、そしてアセスメント結果に基づく品質向上の実現である。

### (1) 実施時期

アセスメントの実施によってテスト設計の不十分な箇所が初めて明確となり、大幅な手戻りが発生する事態を回避するため、プロジェクトが先行的に重要部分のテスト設計に着手した時点で実施する。

またテスト工程の後半で品質問題が顕在化したプロジェクトに関しては、単体テストや統合テストの十分性を検証するため、品質保証レビューの実施に先立って実施する。

### (2) アセスメント範囲の絞り込み

候補の選定では、プロジェクト全体の品質の作り込み状況を的確に把握するため、以下の観点から調査して偏りを避ける。

- ・システムを構成する主要サービスを網羅する
- ・外部ベンダは全てを対象とする
- ・重要な機能及びリスクの高い機能を優先する

品質問題が発生したプロジェクトの場合は、上記に加えて、以下のポイントでレビュー時とテスト時の品質記録と品質メトリクスを分析して、検証活動のプロセス品質を判

定する。

- ・レビューのスキルは十分か
- ・レビュー時間及びテスト件数が目標値と乖離している場合、その理由が明確に説明されているか
- ・当該フェーズで検出すべき障害が抽出されているか
- ・各フェーズの品質メトリクスに偏りがなにか

さらに、問題プログラム（メソッド）の抽出にMcCabeの循環的複雑度のツールによる計測結果も利用する。プログラム内のロジックのパスが増えるほど、整数で示される循環的複雑度が大きくなるため、プログラム構造に問題が存在する可能性が高くなる。したがって、循環的複雑度の大きなプログラムに関する設計書及び実施したテストケースを分析することが、プロジェクトの設計とテストのプロセス品質を把握するプローブの役割を果たすことになる。

Capers Jones [CAPERS1996] は、循環的複雑度が30を超え

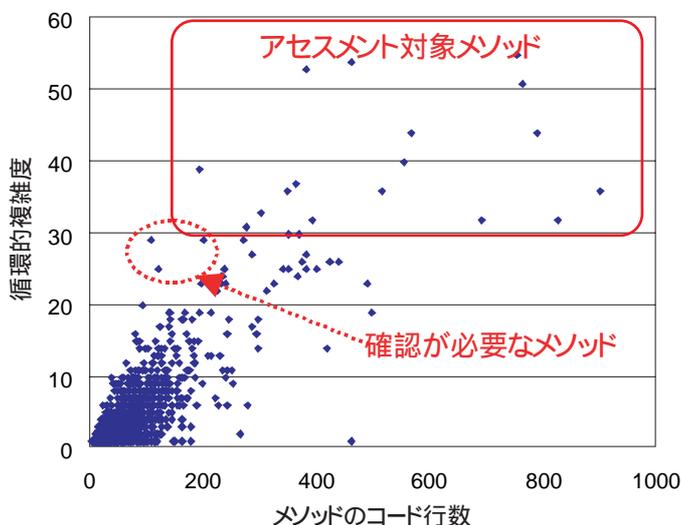


図1 メソッドの規模と循環的複雑度の相関グラフ

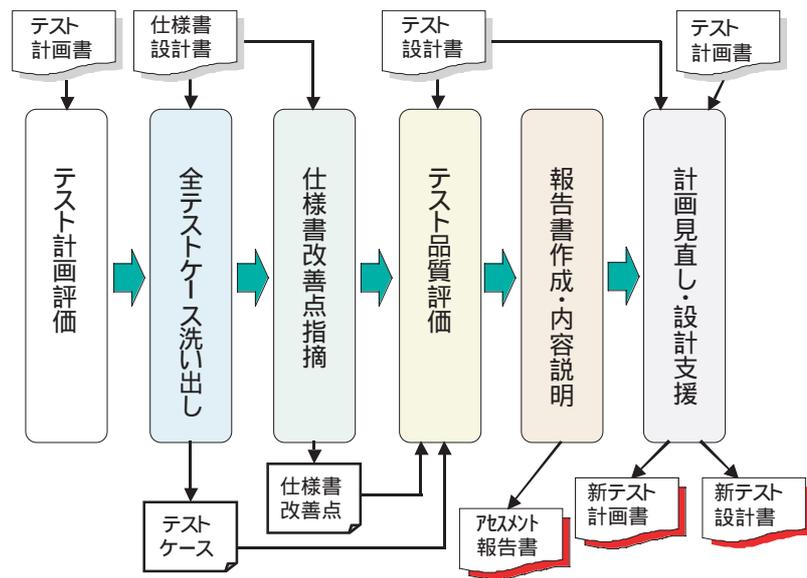


図2 アセスメント手順

ると構造に疑問があるとしており、これまでの品質保証部でのアセスメント結果からも循環的複雑度の有効性が実証されているため、図1に示すように、30を超えるものを対象として抽出する。アセスメントで構造に問題があることが判明した場合は、リファクタリングの対象とする。また、30以下でも、コード行数が少ないにもかかわらず循環的複雑度が高いものは、コーディング規約を遵守していない可能性があるため確認しておく必要がある。

### (3) アセスメント手順

以下の手順でアセスメントを実施する（図2）

テスト計画評価

最初にテスト計画を以下の観点で分析する。

- ・採用しているテスト手法（ブラックボックステスト、ホワイトボックステスト、独自手法等）
- ・テスト基準（テストケースの粒度、実施レベル等）

テスト計画の精度、プロジェクト側のテスト技術の理解度、エンジニアリングプロセス標準の遵守度を評価し、プロジェクトの全体感を把握して、調査の範囲と深度、特に着目するポイント、作業日程の見直しの必要性の判断等のアセスメント方針を確認する。

問題プロジェクトでは、計画自体が貧弱であるか、または別プロジェクトの計画を内容を理解せずに引き写している場合が多く、実現性に乏しいテスト計画となっているため、我々は三現主義（現地・現物・現実）でプロジェクト内に入り込んで実態を確認している。

全テストケース洗い出し

ブラックボックステスト手法でテストケースを設計するため、要求仕様書・設計書を入力として、入力データ仕様とビジネスロジックを整理する。

入力データ仕様から、同値分割、境界値分析、エラー推測の技法で入力条件を明確にして検証に必要なテストデータを抽出する。ここで同値分割ではデータの値だけに着目するのではなく、例えば取消しキーを押す前、1回押した後、2回押した後のような、時間的な同値クラスや、機種やバージョン等の動作環境の違いにも着目して、同値クラスの境界条件を用いてテストデータを抽出する。

またエラー推測技法は、直感と経験から起こりそうなエラーの型を推測して、テスト項目を設計する方法である。テスト設計者の経験への

依存度を極力低下させるため、アプリケーションアーキテクのノウハウを導入して業務分野毎の事例を収集・整理している。

ビジネスロジックに関しては、仕様書・設計書に基づき入力と出力の関係をデシジョンテーブルに展開する。設計書に記載された機能にあいまいな点や矛盾する部分が存在し、このままの記載内容ではテスト設計ができないと判断した時は、フローチャートや状態遷移図等を作成して詳細機能を検証している。これらの情報を用いてテストシナリオを作成する。

テストデータとテストシナリオの作成が終わったら、それらを組み合わせて検証に必要な全テストケースを導出する。

#### 仕様書改善点指摘

上記 では、仕様書・設計書の記載内容がテスト設計

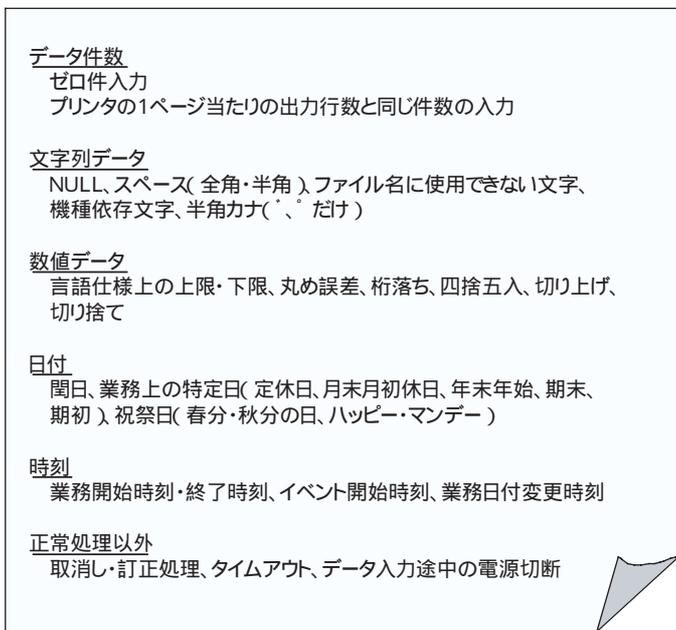


図3 エラー推測事例集の一部

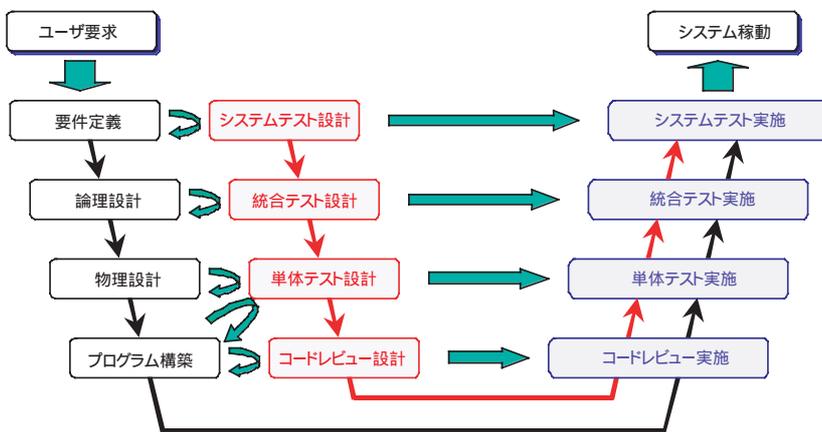


図4 Wモデル

のインプットとして十分な品質に到達しているか検証している。そこで明らかとなったあいまいな記述、矛盾した記述、記述漏れ、記述不足等の箇所に関して、本来記載すべき正しい内容を別途とりまとめている。

これによってテストアセスメントには、機能設計と同期してテスト設計を実施することで、テストの観点から設計内容を検証できる「Wモデル」と同様の品質向上効果が期待できる。

#### テスト品質評価

アセスメントチームが作成したテストケースとプロジェクトが作成したテストケースを対比して、不足する部分や当該テストケースで検出可能な範囲等のテスト設計の品質を評価する。既にテストケースに対応するソースコードが作成済みの場合、プロジェクトが設計したテストケースをもとに、C0カバレッジ(命令網羅)、C1カバレッジ(分岐網羅)を測定し、コードの網羅性も検証する。

図5の事例では、プロジェクト側が設計したテストケースが本来実施すべき件数と比較して相当数不足していることを表しているが、会計業務の画面の正常系テストケースには無駄なテストが存在している。これは、同値分割が適切に実施されず、同じ同値クラスを検査するテストケースが重複して存在しているためである。

一般的に開発担当者の心情から正常ケースのテストが中心となり、異例ケースとエラーケースのテストが不足することが多いが、このテストアセスメントを実施することによってイレギュラーなテストケースが強化され、システムの頑健性が向上する。プロジェクトの特性により異なるが、これまでの事例では、イレギュラーケースが、全テストケースの40

~50%を占めることが多い。

#### 報告書作成と内容説明

評価結果を以下の通り報告書にまとめ、不足する理由も含めてプロジェクト側に説明する。

- ・テストカバレッジの評価(正常系、異常系別)
- ・機能テスト/非機能テストの評価
- ・適用すべきテスト技術
- ・機能仕様と合致しないテストケース
- ・仕様書に追加すべき情報
- ・全体評価

アセスメントで改善点を指摘しても、

プロジェクト側に受け止める力がないと実効性が確保できない。そのため、テスト専門技術者にアセスメント実施後のフォロー活動を義務付けている。

#### 計画見直しと設計の支援

アセスメント結果に基づいて実施するプロジェクト側のテスト計画の見直し作業とテスト設計作業を支援する。

テスト計画の見直しでは、プロジェクトの特性を念頭に置き、サンプリングに対するアセスメント結果から全体を類推し、各機能の重要度と制約条件を考慮して、必要なテストケース数を見積もる。

次に、テストケース設計・テスト準備・テスト実施・テスト結果検証・障害部分修正・再テスト・テスト結果再検証の各々の作業の生産性と、過去の障害発生率実績からテストに必要な工数を算出する。最後に、投入可能な要員数と納期の制約からテスト計画を引き直し、現行テスト計画の見直しの必要性を判断する。

## 3 評価とまとめ

### (1) テストの十分性の保証

テスト専門技術者による体系的な指導に基づき、テストケース設計方針を確認し、テスト設計基準を見直して、不足するテストケースを設計するため、明示的だけでなく暗黙の要求仕様・設計仕様も網羅した漏れのないテストが実施され、テストの十分性が保証される。

これによって品質メトリクスの信頼性が向上するため、品質判定の精度が高まるだけでなく、組織のプロセスパフ

フォーマンスデータとして幅広く活用できている。

### (2) テスト技術への理解の深まり

開発技術は、話題性があり、間断なく新技術が登場し、実務で経験する機会も多いが、テスト技術は、画期的な新技術はあまり登場せず、地味な印象が強いため、特に若手技術者の興味を引く機会が少ないように感じられる。

一連のテストアセスメントの過程を通して、実務で体系的なテスト技術を経験することで、プロジェクト要員のテストに関する理解が進み、技術力の向上が見られるため、要員育成の観点からも有用なOJTの機会となっている。

### (3) 課題と今後の展開

本来は見積り時点でテスト設計方針を明確にすべきであるが、現実には実施できているプロジェクトは少ない。テストアセスメントの有効性は明確であり必要性の理解は深まっているが、対象プロジェクトの拡大に伴うテスト専門技術者不足の解消が緊急の課題である。このため、テストアセスメントを通じた現場での要員育成の機会を増やすと共に、現在は不定期で実施しているテスト技術教育を、次年度からは正規コースに格上げし、質量両面から充実をはかる計画である。

あえてアセスメントを実施しなくても、テスト技術が基本動作としてプロジェクトに早期に定着化するよう、更に活動を強化していきたい。

#### 参考文献

[CAPERS1996] Capers Jones : Applied Software Measurement, Second Edition, The McGraw-Hill Companies Inc., 1996 (「ソフトウェア開発の定量化手法 (第2版)」  
鶴保 征城・富野 壽 監訳, pp.230, 共立出版, 1998)  
[SECDATA] SECが2005年より発行しているソフトウェア開発データ白書

サブシステム名	種別	テストケース数					
		プロジェクト		品質保証部		過不足	
		正常系	異常系	正常系	異常系	正常系	異常系
発注	画面	225	91	257	259	32	168
	バッチ	140	234	2,176	831	2,036	597
発注	画面	246	81	183	370	-63	289
	バッチ	33	10	33	38	0	28

図5 テストケース数の評価例



ソフトウェアエンジニアリング ベストプラクティス賞

# CoBRA法に基づく工数見積りモデル構築への取り組み

株式会社日立製作所



幕田 行雄 小玉 剛久 宇津木 顕彦 福地 豊 谷川 晃一

株式会社三菱総合研究所



石谷 靖 (SEC) 塩田 英雄



江口 良司 渡辺 嘉也 村瀬 武志 定祐 壽美男

特定業種の不特定顧客を対象にしたビジネスアプリケーションソフトウェアを新規に開発するプロジェクトをドメインとして、見積りプロセスを改善するために、CoBRA<sup>1</sup>法に基づく工数見積りモデルを構築することにした。CoBRA法は「オーダーメイド」で工数見積りモデルを構築する手順を示しており、選定した。本報告では、CoBRA法に基づいた工数見積りモデル構築への取り組みを報告する。本報告が今後の見積りプロセスの改善のために少しでも役に立てば幸いである。

## 1 CoBRA法の実証実験の取り組み状況の概要

IPA/SECからの技術指導により、2006年度に株式会社日立製作所 情報・通信グループ（以下、日立）では、特定顧客向けにCoBRA法に基づく工数見積りモデル（以下、モデル）を構築した。日立内PMOメンバである著者らは、このモデルを構築した経験により、CoBRA法の考え方、モデルの構築の手順、及びそのメリットを体感した。

そこで、日立の他事業部門へ横展開することにした。特定業種の不特定顧客を対象にしたビジネスアプリケーションソフトウェアを新規に開発するプロジェクトをドメイン（以下、Aドメイン）として、CoBRA法に基づくモデルを構築した。このAドメインを中心にした、モデルを構築するタスクフォースの取り組みを中心に報告する。

## 2 見積りに関する問題意識とモデル構築手法の選定

### (1) 見積りに関する問題意識

ソフトウェア開発の見積り手順としては、まず、機能要件を入力にして、その確定レベルも考慮して開発規模を見積もる。次に、非機能要件、人的要因、プロジェクト制約、プロセス要因などの複数の工数影響要因を考慮して生産性を見積もる。最後に、開発規模を生産性で割って工数を見積もる。Aドメインの見積りに関する問題意識としては、大きく下記の3点であった。

- 要求仕様があいまいで、見えていない部分の見積り
- 複数の工数影響要因を考慮した論理的な工数見積り
- 見積り作業そのものが短期間

著者らはAドメインのプロジェクトの成功の確率を向上するために、工数見積りは重要な要素の一つであり、を

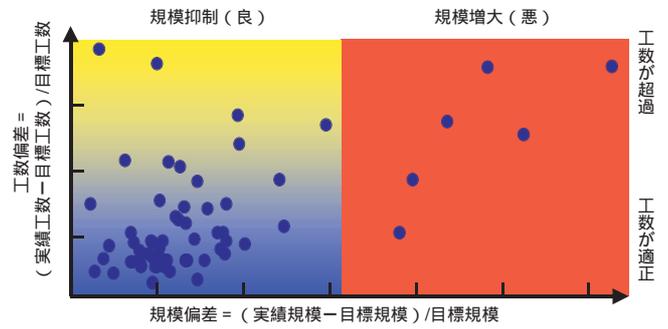


図1 規模偏差と工数偏差の関係

<sup>1</sup> CoBRA は、ドイツ・フラウンホーファ協会実験的ソフトウェア工学研究所 (IESE) の登録商標。[BRIAND1998] Cost Estimation Benchmarking and Risk Assessment CoBRA法の技術解説は[TAKAHASHI2007]等参照。

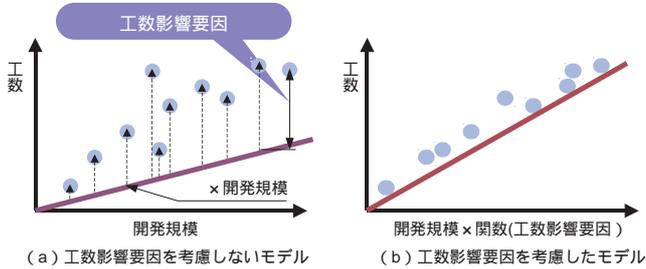


図2 工数影響要因が工数に与える影響

改善したいと考えた。そこで、Aドメインの現状を把握するために、過去プロジェクトの規模偏差と工数偏差の実績データを図1に示すようにプロットした。規模増大しているプロジェクトよりも、規模抑制しているプロジェクトが多かった。一方で、規模を抑制しているプロジェクトが、工数が適正なプロジェクトよりも多いことが確認できた。この結果より、工数が超過しているプロジェクトを、工数が適正なプロジェクトにすることが重要であることが再認識できた。

そこで、タスクフォースの目的を、「工数見積りの妥当性をプロジェクトマネージャ（以下、プロマネ）が合理的に判断・説明するための道具を整備することにより、工数見積りの見える化の品質と効率を向上したい」とした。従来は図2(a)に示すように、工数影響要因を考慮しないモデルでは、開発規模が同じでも必要な工数に違いがあることを論理的に説明できなかった。一方、図2(b)に示すように、工数影響要因を考慮したモデルを構築することができれば、過去プロジェクトを論理的に説明できると考えた。

## (2) モデル構築手法の選定

過去プロジェクトの実績データを収集し、工数影響要因を考慮したモデルを構築するための手法が多くある。モデル構築手法として、COCOMO [COCOMO]、協調フィルタリング法、CoBRA法から選定することにした。

COCOMO では、161件のプロジェクトの実績データをベースにして、工数を目的変数とし、開発規模、スケールドライバ、コストドライバを説明変数にしたモデルを提供している。COCOMO のスケールドライバとコストドライバは工数影響要因に略対応する。一方、協調フィルタリングに基づく工数見積り手法 [KAKIMOTO2006] は、データの欠損を含んだ過去プロジェクトの実績データに対して、欠損率が增大しても見積り精度が大きく低下しないので有益な手法である。COCOMO または協調フィルタリング

に基づく工数見積り手法は、ドメイン内の多くの実績データを用意することができれば、ドメインに特化したモデルを構築することが可能である。

しかし、一般的な組織では、統計的な分析ができるほどに、ドメイン内の多くの実績データを用意できないのも現実である。また、ドメイン内の多くの実績データを用意してモデルを構築したとしても、情報技術の進歩のスピードが速いので、構築したモデルが情報技術にマッチしない可能性も発生する。

これらの課題に対して、CoBRA法は、ドメインに特化したモデルを構築する手順を与えるものである。少数（10個程度）の過去プロジェクトの実績データと経験豊富なプロマネ（以下、熟練者）の暗黙知を活用してハイブリッドなモデルを構築することができる。したがって、CoBRA法では、ドメインの熟練者の暗黙知を活用することにより、ドメイン内の多くの実績データを用意しなくてもよいので、ドメインに特化したモデルを容易に構築できることが特徴である。COCOMO を「既製服」のモデルだとすると、CoBRA法は「オーダーメイド」でモデルを構築する手順を示しているといえる。著者らはAドメインでCoBRA法に基づくモデルを構築することができると考えた。

## 3 CoBRA法に基づくモデル構築結果

### (1) モデルの評価結果 [MAKUTA2007]

「アセット」を活用したCoBRA法に基づくモデル構築の手順により、モデルを構築した。Aドメインの9個のプロジェクトでモデルを構築し評価した。モデル評価は、過去プロジェクトの実績データに対してクロスバリデーション法（1つのデータを残りのデータで構築したモデルにより見積もる方法）で検証した。相対誤差を下記の式で算出した。

$$\text{相対誤差} = |\text{見積り工数} - \text{実績工数}| / \text{実績工数}$$

評価項目は、平均相対誤差（MMRE）相対誤差の標準偏差（Std.Dev）相対誤差25%のプロジェクトの割合（Pred.25）を用いた。モデルの評価結果を表1に示す。MMREが17.6%となりタスクフォースの目的が達成できると判断した。

表1 モデルの評価結果

MMRE	Std.Dev	Pred.25
17.6%	12.6%	77.8%

## (2) モデルの適用結果

CoBRA法に基づいて構築したAドメインのモデルをプロマネに適用させた。意見をまとめると、下記の3点であった。

- ・工数影響要因毎の影響工数を確認し、見積り中に気づき  
が得られ、工数見積りに関するリスクを低減することが  
できた(図3)
- ・3点見積り(最小値、中央値、最大値)で表示(算出)  
されるので、プロジェクトのリスク許容度などを考慮  
して見積りに活用することができた(図4)
- ・モデルの適用は、工数影響要因のレベルの判定が主な作  
業なので、それほど手間ではなかった。

## 4 モデル構築で工夫した点

### (1) 「アセット」を活用したCoBRA法に基づくモデル 構築手順

従来のモデル構築手順では、モデルを構築するために、  
熟練者を多く関与させなければならなかった。第一線で活  
躍していて忙しい熟練者の協力を得るためには、CoBRA  
法の有効性をあらかじめ示す必要がある。

著者らは、熟練者の暗黙知が形式知化された「アセット」  
を有効活用することにより、熟練者を全く関与させること  
なく短期間に仮モデルを構築する手順を提案した  
[MAKUTA2007]。なお、「アセット」とは「組織アセット」  
と「外部アセット」の2種類がある。例えば、「組織アセ  
ット」には、Aドメインに蓄積されているプロジェクトマ  
ネジメント研修で利用した演習事例とプロジェクトアーカ  
イブ情報、及び見積りガイドブック等がある。「外部アセ  
ット」には、[SEC2006]、[COCOMO ]等がある。

「アセット」を活用したCoBRA法に基づくモデル構築手  
順を表2に示す。まず前工程で仮モデルを構築し、続いて  
後工程でモデルを構築するものである。

#### 前工程：仮モデルの構築手順

前工程では、熟練者の暗黙知が形式知化された「アセ  
ット」を有効活用することにより、熟練者を全く関与させ  
ることなく、著者らは5人日(1人×5日)で仮モデルを構築  
することができた。ただし、著者らがCoBRA法に基づい  
たモデル構築の経験があったこと、及び作業ステップ5を  
サポートするモンテカルロシミュレーション<sup>\*2</sup>が実装され  
たCoBRAツールがあったことが前提条件である。

仮モデルを構築し、その評価結果としてMMREが29.5%  
となり、タスクフォースの目的が達成可能であると判断し  
た。この実証結果を熟練者に説明したところ、熟練者たち  
は、自分たちの暗黙知を更に提供すれば、もっと良いモデ  
ルが構築できるとすぐ感じてくれ、多くの熟練者の積極  
的な協力を得ることができた。そこで、本格的に、CoBRA  
法に基づいたモデルの構築に着手した。

#### 後工程：モデルの構築手順

次に、後工程では、多くの熟練者の積極的な協力と議論

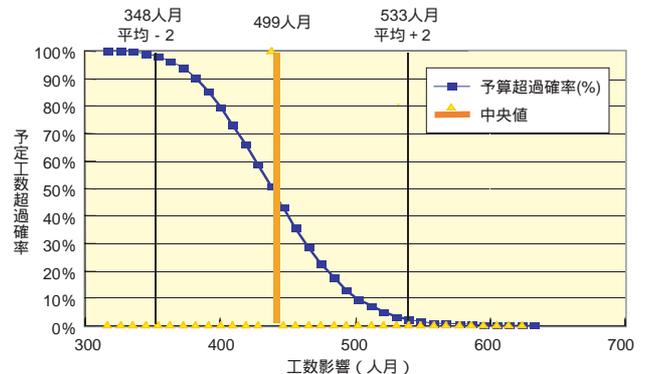


図4 予定工数超過確率分布(事例2)

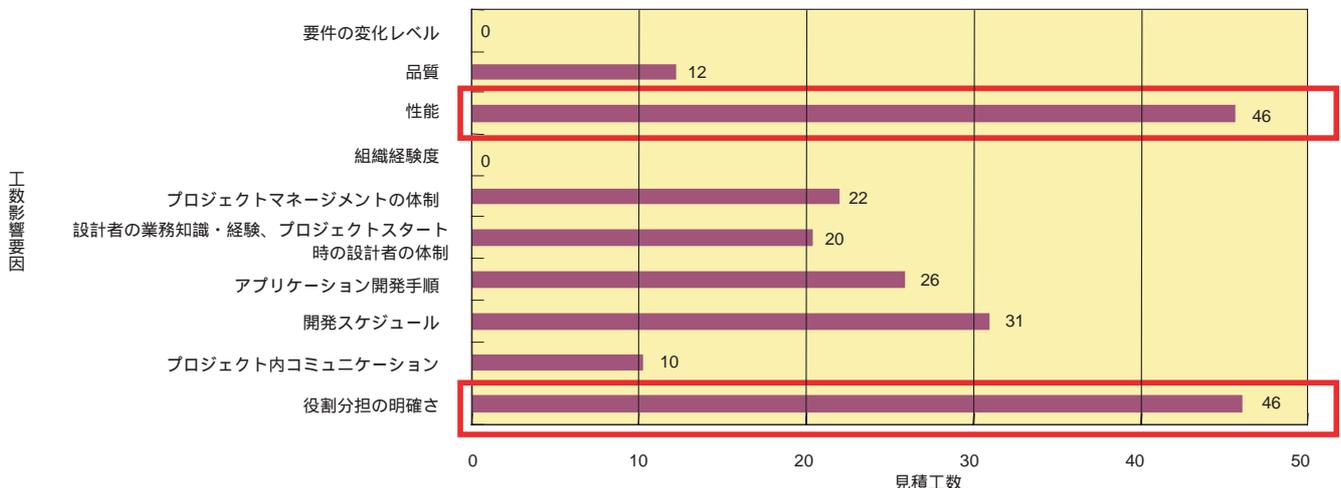


図3 工数影響要因の影響分析(事例1)

2 モンテカルロシミュレーション：乱数を用いて繰り返し計算することにより近似解を求める計算手法

表2 モデル構築手順

作業ステップ	工数影響要因の 関係図の設定	工数影響要因と そのレベルの定義	工数影響要因の影響度の設定	過去プロジェクトの 実績データ収集	見積りモデルの構築と評価
工程					
前工程	ステップ 1	ステップ 2	ステップ 3	ステップ 4	ステップ 5
後工程	ステップ 6	ステップ 7	ステップ 8	ステップ 9	ステップ 10

により、前工程で構築した仮モデルをブラッシュアップして、モデルを完成させた。前工程で構築した仮モデルをブラッシュアップするだけなので、熟練者の関与を最小化できた。10人の熟練者が関与した作業ステップは6～8であった。モデル構築のために熟練者の関与した総実績時間は45時間（270分/人×10人）であった。

## (2) 工数影響要因とそのレベル定義で考慮した点

工数影響要因とそのレベル定義（ステップ8）で考慮した点を説明する。

複数の利用場面で、工数影響要因のレベルを定義

工数影響要因として、「要件の変化レベル」の定義の事例で説明する。利用場面として、新規見積り時と再見積り時の二つを想定してレベルを定義した。新規見積り時は、各業務ごとの業務要件キーマンの取りまとめ能力と体制でレベルを定義した。また、再見積り時は、システム全体に影響を与える要件の変更がある工程で、レベルを定義した。プロマネがその時点の情報で客観的なレベルの判定ができるようにするために、複数の利用場面を想定して、工数影響要因のレベルを定義した。

複数の項目で評価するように工数影響要因を定義

工数影響要因として、「設計者の業務知識・経験、プロジェクトスタート時の設計者の体制」の事例で説明する。評価する項目を、「業務カバー率（＝割り当て可能業務数/全業務数）」と「プロジェクトスタート時の体制を設計者の適正人数」の二つとした。この二つの項目を独立に評価した後に、加重平均などで総合評価することにより、工数影響要因のレベルを判定するようにした。

また、工数影響要因のレベルの定義は、例えば業務カバー率のように定量的な定義となるように心がけた。

## (3) 3点見積りの実践

図4に示すように予定工数超過確率分布に最小値、最大値を平均±2σで表示することにより、熟練者の暗黙知を活用した3点見積りが実践できるようにした。

## 5 今後の取り組み

プロジェクトコントロールの「モノサシ」の一つとして「工数影響要因」をマネジメントに活用することにより、コミュニケーションの品質と効率の向上が可能であると考えられる。プロジェクトの成功の確率を向上するために下記の2点を中心にして取り組んでいく。

### (1) プロジェクトマネジメントへの活用

工数影響要因がまだリスクである間にできるだけ工数影響要因のレベルを下げると同時に、工数へ大きな影響を与える工数影響要因は影響を小さくするようなマネジメントが重要である。工数へ大きな影響を与える工数影響要因は、標準的な対策方針をベースにして対策することも重要である。

### (2) ユーザ企業とのコミュニケーション

ユーザ企業とベンダ企業間で、工数へ大きな影響を与える工数影響要因を互いに認識して、プロジェクトを推進することが重要である。

## 謝 辞

この取り組みに積極的にご協力を頂いた日立のプロマネ、及びPMOに感謝する。また、日立でのCoBRA実証実験に関して、技術指導を頂いた関係各位に感謝する。

## 参考文献

- [BRIAND1998] Briand, L.C., El Emam, K., Bomarius, F. : COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking and Risk Assessment, Proceedings of the 20th International Conference on Software Engineering, pp. 390-399, 1998
- [COCOMO ] COCOMOモデル : <http://csse.usc.edu/csse/>
- [KAKIMOTO2006] 柿元健、角田雅照、大杉直樹、門田暁人、松本健一：協調フィルタリングに基づく工数見積り手法のデータの欠損に対するロバスト性の評価、電子情報通信学会論文誌、Vol.J89-D, No12, pp.2602-2611, 2006
- [MAKUTA2007] 幕田行雄、福地豊、谷川晃一、江口良司、渡辺嘉也、村瀬武志、石谷靖、塩田英雄：CoBRA法に基づくソフトウェア開発プロジェクトの見積りモデル構築手順の改善、プロジェクトマネジメント学会2007年度秋季研究発表大会予稿集、pp.110-115, 2007
- [SEC2006] IPA SEC : ソフトウェア開発見積りガイドブック、オーム社、2006
- [TAKAHASHI2007] 高橋茂：ソフトウェア開発見積り評価モデルCoBRA, SEC journal, No11, pp.26-31, 2007



ソフトウェアエンジニアリング ベストプラクティス賞

# 組み込みシステムにおけるモデルベース開発 (MBD) 技術者のスキル標準

## 事例:自動車分野のMBD技術者に必要なスキル(ETSS-JMAAB)



JMAAB MBDエンジニア育成ワーキンググループ 幹事  
三菱電機株式会社 姫路製作所 制御機器第一製造部  
山田 元美

自動車は常に改良され続けているが、近年は、特に環境技術、安全技術における革新が強く求められている。また、従来にもまして利便性、快適性の改善も求められる。これらの要求に応えるために、電子制御による高機能化、多機能化が急速に進んでいる。高機能化、多機能化によりソフトウェアが量的に拡大すると開発工数が増大する一方、品質保証も大きな問題となっている。

そのため、自動車業界において制御系の開発効率向上及び品質向上のために、MATLAB/Simulinkをコアツールとしたモデルベース開発 (MBD<sup>1</sup>) が量産にまで展開され始めた。

### 1 JMAABとは

JMAAB<sup>2</sup>は、日本の自動車メーカーとECUサプライヤが参加しているMATLABのユーザ会である。世界の主要自動車メーカーやECUサプライヤ、The Math Works (MATLAB製品開発元) が参加する

MAABもあり、JMAABと連携して活動している。「開発環境構築は協調し、競争は製品で！優れた環境でレベルの高い競争をしよう！」をスローガンに、MBD推進と開発環境をより発展させるために様々な活動を行っている。図1にJMAABの組織構成を示



図1 JMAABの組織構成

す。JMAABは、ユーザ会の発起人中心で構成されるボードメンバ、実際の活動を行うワーキンググループ (以下、WGと略称)、WG活動に参加しているコアメンバ、JMAAB活動に関心を持ちインターネットを通して会員登録した一般メンバで構成されている。

### 2 MBDとは

JMAABでは、「MBD」を下記のように定義している。

- ・モデル：  
対象の機能が図示されており、一意的に解釈できるもの
- ・MBD：  
複雑化・高度化した現代の自動車制御システム開発において、MATLAB / Simulink等のCAEツールによって制御装置と制御対象の機能をモデル化し、開発の初期段階からソフトウェア設計の段階までモデルを活用することで、製品ライフサイクル全般に亘った品質向上と開発効率向上を目指した開発手法のことである。シミュレーション技術を駆

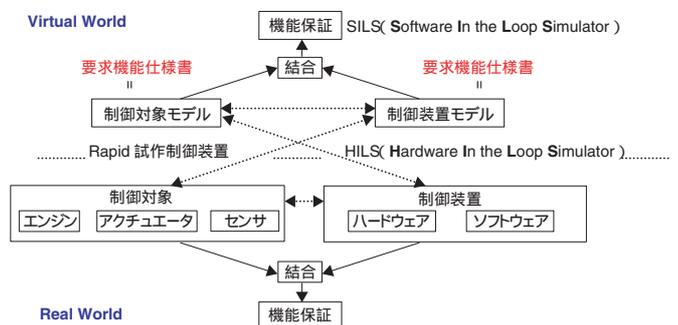


図2 MBDの定義

1 MBD : Model Based Development  
2 JMAAB : Japan MATLAB Automotive Advisory Board

使うことで高精度な機能確認を実施でき、かつ、複雑な開発行程のルーチンワーク化を促進することで、自動化・省力化にも貢献する。

### 3 MBD技術者育成の重要性

MBDの推進にあたり、ツールの機能改善と共に重要となるのがエンジニアの育成である。CAEツールや設計技術の開発改良が進む中で、自動車制御系開発における各種ツールの知識や使いこなし、制御システムの設計検証技法、設計プロセスにおける管理ツールの活用手法など、自動車業界のエンジニアにとって必要となる知識やスキルの範囲が広がり、何らかの指針が必要になってきている。優れたツールも、最新の技術や手法を十分に取り扱える人材がいなければ成果を出すことができない。そのためMBD技術者育成は重要な、また急務な課題となっている。

### 4 MBDエンジニア育成WG

MBD技術者育成を推進するために「JMAAB MBDエンジニア育成WG」というワーキンググループ（以下、育成WGと略称）を2005年に発足した。育成WGのねらいは、自動車メーカーとサプライヤの視点から、MBDにおけるモデリングやツールに関する知識・設計プロセスにおける活用手法など、マネージャ/技術リーダー/実務担当者を対象とした共通課題を明らかにして、MBDエンジニアに必要な教育プログラムの企画、及び共通指標となる認定レベルの制定を行うことである。ETSS-JMAAB策定においても図3に示すような観点で活発に議論を行った。

育成WGは、JMAAB中でも参加企業がもっとも多いWG（現在13社）となっている。

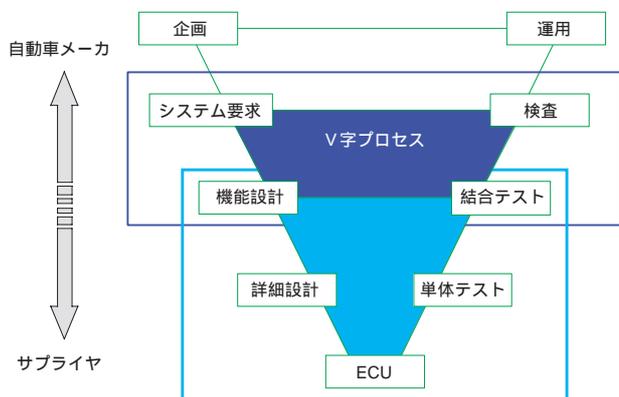


図3 自動車業界の役割分担（ソフトウェア）



図4 MBDエンジニア育成WGメンバー（前列中央が筆者）

### 5 ETSS-JMAABについて

育成WGは、2006年にETSS-JMAABの4種類の文書のVer1.0を作成完了し、2007年6月よりJMAABホームページ（<http://j-maab.cybernet.jp/>）で一般公開している。ETSS-JMAABは、IPA/SEC発行の「組込みスキル標準 ETSS」（ETSS：Embedded Technology Skill Standards）2006年度版をもとにカスタマイズされている。

ETSS-JMAABは、自動車制御系開発におけるMBDエンジニアを対象として、開発技術、要素技術、管理技術に関するスキルとキャリアをまとめたものであり、以下の領域での活用を期待している。

- 自部署のスキルレベル把握と人材育成計画の立案
- 外注者のスキルレベル把握（受け入れレベル規定等）
- 教育カリキュラム構築 ...他

各々の要素は、図6に示すようにマッピングして関連付けている。

スキル基準：MBDによる制御系開発に必要なスキルを体系的に整理するフレームワーク

キャリア基準：MBDによる制御系開発に関わる職種名称や職掌とそれに求められるスキルを定義するフレームワーク



自動車メーカーとサプライヤの全開発プロセスを含む

図5 ETSS-JMAABの4つの文書

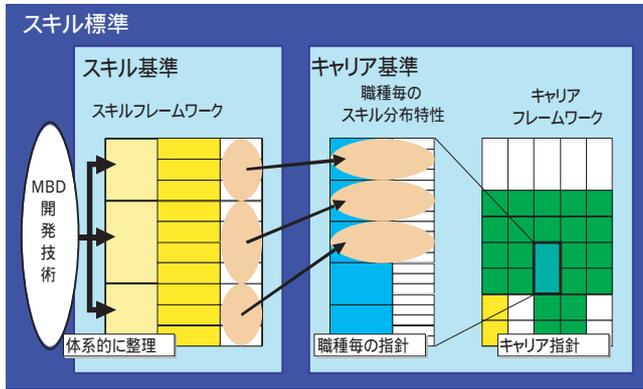


図6 ETSS-JMAABの全体構成

## 6 ETSS-JMAABのポイント

### (1) スキル基準

スキル基準は、MBDによる制御系開発に必要なスキルを明確化・体系化したものであり、人材育成・活用に有用な「ものさし」(共通基準)を提供する。スキルは「技術要素」「開発技術」「管理技術」のカテゴリで整理・階層化されている。図7にETSS-JMAABのスキルフレームワークを示す。

ETSSと同様に、スキルとは作業の遂行能力を指し、「～ができること」を表現するものであり、知識を有するだけではスキルとは扱わない。知識はスキルを発揮するために必要な構成要素で、特に自動車の製品知識と工学基礎知識はMBDエンジニアに必要な知識である。利用観点毎のスキルと知識の関係は図8のように示される。

スキル基準で定義する技術の範囲は、共通的に利用されるものを想定し、各企業や応用ドメインで利用される特有の技術に関しては扱っていない。また、スキル粒度の第3

スキルカテゴリ	スキル粒度			スキルレベル					
	第1階層	第2階層	第3階層(例)	説明	レベル1	レベル2	レベル3	レベル4	レベル5
					初心	初級	中級	上級	最上級
技術要素									
開発技術									
管理技術									

図7 ETSS-JMAABのスキルフレームワーク

階層は例にとどめている。運用する各企業で具体化・追加等が必要となる。技術要素と開発技術カテゴリについては、図9、図10でETSSとの違いを示す。

スキルレベルは以下の0～5の6段階で定義している。

- ・レベル0： 内容を知らない。
- ・レベル1：初心 内容を知っている。
- ・レベル2：初級 支援のもとに作業を遂行できる。
- ・レベル3：中級 自立的に作業を遂行できる。
- ・レベル4：上級 作業を分析し改善・改良できる
- ・レベル5：最上級 新たな技術を開発できる。

### (2) キャリア基準

キャリア基準は、MBDに従事する技術者の主な職種、その内容、レベル、求められるスキルを明示したものである。MBDによる制御系開発に従事する技術者の職種を14

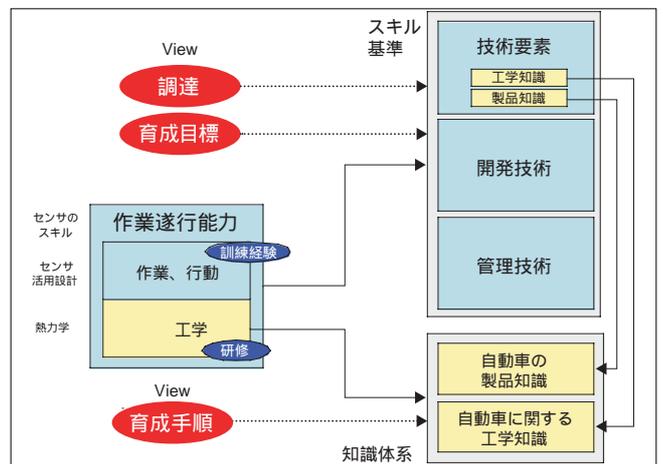


図8 利用観点 (View) ごとのスキルと知識の関係



図9 技術要素スキルカテゴリの第一階層

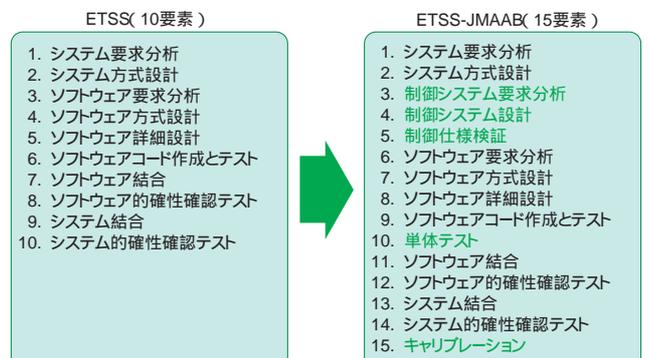


図10 開発要素スキルカテゴリの第一階層

種類定義している(図11、表1)

キャリアレベルは、モデルベース開発の各職種において、技術者が専門性を持った人材として価値創出に応じたスキルの度合いを表したもので、3段階のレベルで定義している(図12)

制御システム設計に直接携わるエンジニア(ドメインスペシャリスト、システムアーキテクト、モデルエンジニア、制御アーキテクト、ソフトウェアエンジニア、適合エンジニア、テストエンジニア)8職種については、具体的なスキルレベルマップを示している。

## 今後の展開と期待

ETSS-JMAABは、経営者層の人材・リソース戦略立案や

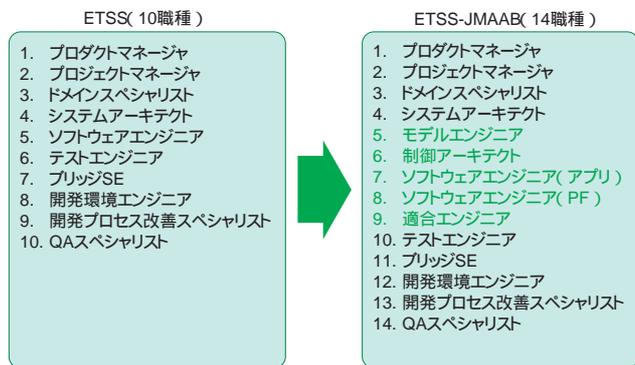


図11 職種

表1 職種と責任及び役割の対応表

職種名	業務の概要	責任	業務	関係システム
プロジェクトマネージャ	製品開発の推進	組織・業務共有	製品を市場に届ける準備、プロジェクトの進捗管理	製品投入に係る関係の業務
プロジェクトマネージャ	システム開発・ソフトウェア開発	プロジェクトの品質・コスト・納期	プロジェクトの進捗管理、品質管理、コスト管理	製品投入に係る関係の業務
ドメインスペシャリスト	開発の業務	プロジェクト(企画) 開発の業務	開発業務の推進、品質管理、コスト管理	製品投入に係る関係の業務
システムアーキテクト	システム開発	開発業務(コスト削減)・品質	システム開発の推進、品質管理、コスト管理	システム開発の業務
モデルエンジニア	モデルの開発	品質・納期	システム開発の推進、品質管理、コスト管理	システム開発の業務
制御アーキテクト	システム開発	品質・納期	システム開発の推進、品質管理、コスト管理	システム開発の業務
ソフトウェアエンジニア	ソフトウェア開発	ソフトウェアの品質・納期	ソフトウェア開発の推進、品質管理、コスト管理	ソフトウェア開発の業務
ソフトウェアエンジニア	ソフトウェア開発	ソフトウェアの品質・納期	ソフトウェア開発の推進、品質管理、コスト管理	ソフトウェア開発の業務
適合エンジニア	システム開発	プロジェクトの品質・コスト・納期	システム開発の推進、品質管理、コスト管理	システム開発の業務
テストエンジニア	システム開発	プロジェクトの品質・納期	システム開発の推進、品質管理、コスト管理	システム開発の業務
PLM	プロジェクトの品質	プロジェクトの品質・コスト・納期	プロジェクトの推進、品質管理、コスト管理	プロジェクトの業務
開発環境エンジニア	開発環境の構築	開発環境の構築・保守	開発環境の構築、保守	開発環境の業務
開発プロセス改善スペシャリスト	開発プロセスの改善	開発プロセスの改善	開発プロセスの改善	開発プロセスの業務
QAスペシャリスト	プロジェクトの品質	開発業務の品質	プロジェクトの推進、品質管理、コスト管理	プロジェクトの業務



図12 ETSS-JMAABのキャリアレベル

評価、プロジェクトマネージャによるコストマネジメントや組織マネジメント、個人のスキル向上などへ活用が考えられる。また、ETSS-JMAABは、MBDによる自動車制御系開発を対象としているが、他の制御系開発にも適用可能な部分も多々ある。

JMAABの目的は、業界においてMBDの普及を進めることにより、高品質の製品を短期間に顧客に提供することであるとみえる。そこで、特に重要と考えているのは、要員教育に対する活用であるが、現在のところ体系化された教育カリキュラム・コンテンツが不足している。

育成WGはETSS-JMAABを図13のように位置付け、各企業や教育機関の皆様と連携し、より質の高い技術者教育の実施を目指している。現在は、今後の維持改定の仕組みづくり、及び教育カリキュラムなどについての討議を行っており、今後カリキュラム・導入手順書などを含めた改訂版を公開していく予定である。

## 謝辞

ETSS-JMAAB作成並びに本稿の作成にご尽力頂いたWGメンバーを含め、関係各位にお礼を申し上げます。

## 参考

- ETSS-JMAABのダウンロード/アンケート <http://j-maab.cybernet.jp/>
- 問い合わせ先: JMAAB事務局 [j-maab-jimukyoku@cybernet.co.jp](mailto:j-maab-jimukyoku@cybernet.co.jp)

## JMAAB MBDエンジニア育成WG歴代参加企業

アイシン精機、アイシンAW、アドヴィックス、いすゞ自動車、ジャトコ、スズキ、デンソー、トヨタ自動車、日産自動車、日産ディーゼル、日立製作所、本田技術研究所、マツダ、ミツバ、ヤマハ発動機、サイバネットシステム、三菱電機

## 参考文献

[SEC2006] IPA SEC : 組込みスキル標準ETSS概説書[2006年版], 翔泳社, 2006

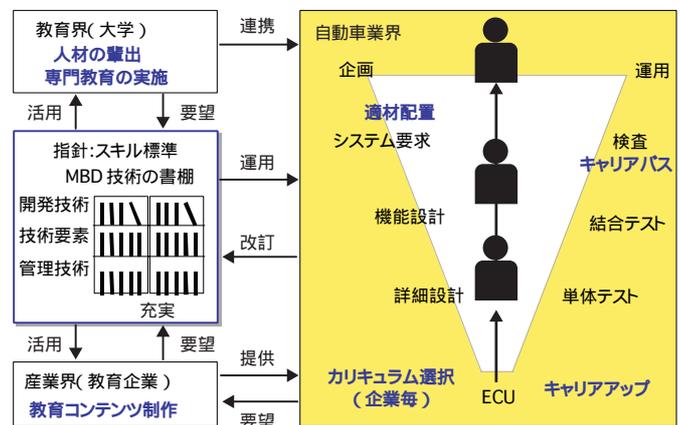


図13 MBDエンジニア育成 目指す世界



貢献賞

# 定量データ提供企業グループ

- 株式会社アルゴ21
- NECソフト株式会社
- NTTソフトウェア株式会社
- 株式会社NTTデータ
- 沖ソフトウェア株式会社
- 沖電気工業株式会社
- 株式会社構造計画研究所
- 株式会社CSKシステムズ
- 新日鉄ソリューションズ株式会社
- TIS株式会社
- 東芝情報システム株式会社
- 日本ユニシス株式会社
- 株式会社野村総合研究所
- 株式会社日立システムアンドサービス
- 株式会社日立製作所
- 日立ソフトウェアエンジニアリング株式会社
- 富士通株式会社
- 松下電器産業株式会社
- 三菱電機インフォメーションシステムズ株式会社
- リコーソフトウェア株式会社

(50音順)



表彰式後、審査委員を交えての記念撮影



定量データ分析部  
会合田主査(右)  
とIPA藤原理事長

## 貢献賞授賞の理由

「定量データ提供企業グループ」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 定量データ分析部会(主査:合田治彦氏(富士通))のメンバ企業である。

SEC発足(2004年)以前、日本国内にリファレンス可能なプロジェクトデータは存在せず、各企業がベンチマーク等を行おうとしてもできなかった。また、情報システム開発の受発注の二者間においても定量的な参考情報がないため相互理解が困難であった。

そのような状況のなか、同グループは、各社の情報やプラクティスを提供することにより、日本における定量データによる科学的・工学的なアプローチへの礎を築いた。各社がデータ分析の観点、ノウハウ、データの提供等を率先して実施したことにより集積した1,770件のプロジェクト

データは、貴重な企業横断データベースとなった。そして、それは、「ソフトウェア開発データ白書<sup>1)</sup>」という書籍として広く一般に利用されるようになり、「プロジェクト診断支援ツール<sup>2)</sup>」というシステムを実現し、日本国内での定量データに基づくプロジェクトマネジメント活動を加速した。また、収集データは国内の大学でも分析実証を可能とし、結果的に10件以上の論文発表につながり、実証ソフトウェア工学の分野で日本の国際的な認知を高めた。

このように同グループがわが国のソフトウェアエンジニアリング活動の推進に貢献したことを称え、IPAより「貢献賞」を授賞したものである。

現状の収集データはマクロな傾向をつかむために有効であるが、さらに精度が高く詳細なデータを収集することにより、信頼性や生産性等の向上につながるであろう。

(「SEC journal」編集部)

1 ソフトウェア開発データ白書: SECが2005年より発行しているプロジェクトデータ集  
 2 プロジェクト診断支援ツール: [https://sec.ipa.go.jp/project\\_assessment/TopMenu.do](https://sec.ipa.go.jp/project_assessment/TopMenu.do)

# ISO/PC 236第1回会議参加報告

SEC 研究員 新谷 勝利

## 1 ISO/PC236とは？

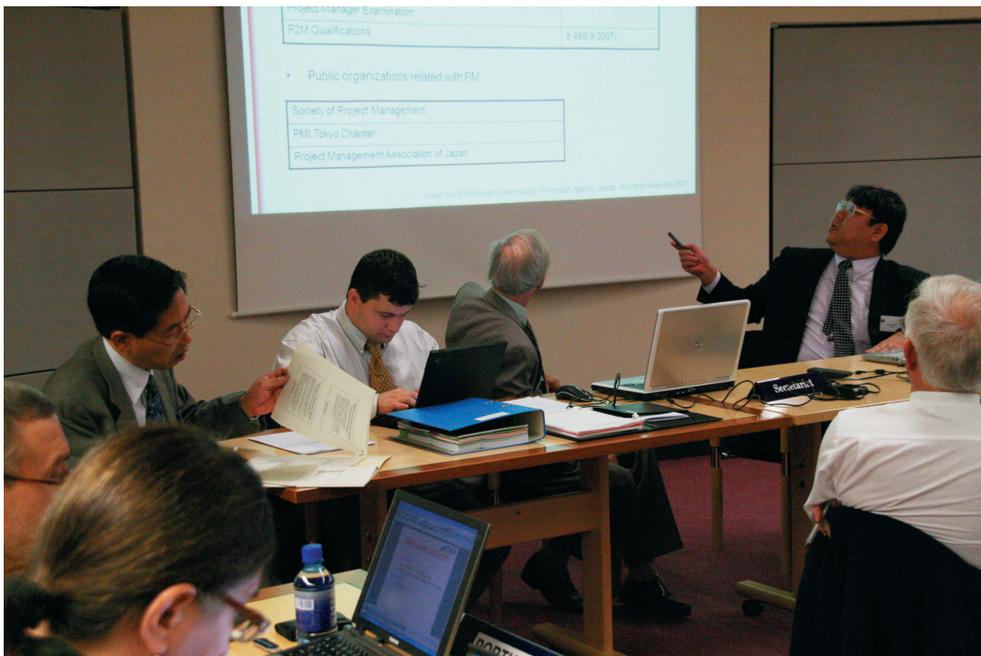
2007年1月、英国規格協会（British Standards Institution：BSI）が新規案件としてISOに提案した「プロジェクトマネジメントに関わる国際標準作成」が賛成多数で承認された。これを受けて2007年2月、ISO/PC236委員会が設置され、プロジェクトマネジメントの国際標準化の作業が開始された。日本では2007年6月、独立行政法人 情報処理推進機構（IPA）にPC236日本委員会事務局が設置された。3年以内にこの作業、つまり国際標準としての発行を完了するために、英国は議論のベースとしてそのプロジェクトに関する国家標準であるBS6078:2002を、米国はANSI\*/PMI99-001-2004であるPMBOKガイド（A Guide to the Project

Management Body of Knowledge）第3版の用語集を提供することになった。また、PC236の委員長は英国が、事務局長は米国が受け持つ体制となった。

## 2 ISO/PC236第1回会議

PC236委員長を出している英国のBSIロンドン事務所にて、上記の決定に沿った事前の委員長、事務局長打ち合わせに従い、WG0～WG3に分かれ、それぞれ次のような目的で活動を開始した。

- WG0：全体会議
- WG1：用語集
- WG2：プロセス
- WG3：その他の項目で、主としてアネックスに記述するもの。例：スキル



右から日本代表団の副委員長、Dr. Jim Gordon（ISO/PC 236 Chairman）、Mr. Jason Knopes（ISO/PC236 Secretary）、IPA今清水理事。

\* ANSI：American National Standards Institute

他の国際標準開発の手順と同じように、以下の要領で実施された。

- ・ 10月29日：全体会議（WG0）
- ・ 10月30日：WG1～WG3に分かれての分科会
- ・ 10月31日午前：WG1～WG3に分かれての分科会  
午後：全体会議（WG0）
- ・ 11月1日：WG1～WG3に分かれての分科会
- ・ 11月2日午前：同上  
午後：全体会議（WG0）

PC236の第1回会議は正式にはこの全体会議で終了し、今回の会議での合意事項を確認した。

- ・ 11月3日午前：WG1～3に分かれての分科会

2008年4月に予定されている次回PC236国際会議に向けて、各作業部会では進め方等を議論する。日本からは、IPAの今清水理事、PC236日本委員会の関委員長をはじめ7名が参加し、WGを分担した。ホテルに帰ってからは、毎夜ロビーにて、それぞれが自分の分担した部分を説明し、翌日のアプローチ、発言内容等を調整した。

### 3 PC236第1回会議の決定事項

最終全体会議における今後のスケジュール説明では、以下のように合意された。

- ・ 2008年2月：WD（Working Draft）をPC236会議参加国に配布、各国はコメントを国際会議までに提出する。
- ・ 2008年4月21～25日：PC236会議、ワシントンD.C.にて上記を解決し、WDを更新する。
- ・ 2008年6～9月：改定されたWDへコメントを反映する。
- ・ 2008年11月3～7日：PC236、ミュンヘンあるいはその近辺、WDへの解決案を取り込み、CD（Committee Draft）を作成する。
- ・ 2008年12月：CDを作成する。
- ・ 2008年12月～2009年3月：CDへのコメントを提出する。
- ・ 2009年03月～06月：CD投票を行う。

各WGでの主たる議論は、以下のように全体会議に



日本案を説明するIPA 今清水理事（左端）。

報告された。

#### WG1：用語集

- ・ WG1のベース文書として、ANSIから提案されたPMBOKの用語集を用いる。
- ・ 2008年4月のワシントンでの会議まで個別活動。
- ・ 2007年11月30日：各WGは、用語集に採用する候補をWG1に提出する。
- ・ 2008年2月1日：各WGは、WDに含めるために、テキスト等をWG1へ送付する。

#### WG2：プロセス

- ・ プロセス定義をするために、以下の観点をベースとする。

プロジェクトに限定する。

プロセスのカテゴリーは以下とする。

立ち上げ

計画

実行

監視とコントロール

終結

サマリー/イントロダクション

- ・ 2008年4月のワシントンでの会議まで個別活動。以下の2項目をこの間に達成する。

2008年1月：委員はアサインされたプロセスを定義し、WDの作成を開始する。

2008年2月15日：WD発行

#### WG3：国際標準の全体構造およびアネックス作成

- ・ プロセスマネジメントの国際標準はISO 21500といい、WG3は目次を作成する。
- ・ 国際標準の構造の一部として「プロジェクトマネジメントにおけるスキルとコンピテンシー」に関する文章を本文及びアネックスに記述する。
- ・ この部分は、ITSSの考え方を反映する記述になる予定（54頁の写真は、ITSS体系をPM用に再定義したPMSS体系を説明している場面である）。

#### 4 今後の活動について

日本のプロジェクトマネジメントの国際規格化に貢献できる内容は、英国及び米国のように、すでに国内にプロジェクトマネジメントの標準を持っている国とは異なることになる。以下のような貢献に加えて、提案されるドラフトに積極的にコメントし、タイムリーに返事をする必要があるであろう。

- ・ WG1においては、他のWGの作成する文書から抽出することになるが、後日、日本語に翻訳されることを想定し、翻訳するにあたり適切な訳語が用いられるように、また、わかりやすい英語による用語説明がなされるように議論に参加する。
- ・ WG2においては、日本で比較的共通に用いられているであろうプロセス定義を今回のプロセス定義に反映する。
- ・ WG3においては、ITSSの考え方を反映するように、積極的にアネックスの文面を提案していくことになる。

ドラフトを書く作業は、どの国際規格作成においても、Subject Matter Expertとしての委員個人であり、日本国内委員会ではない。しかしながら、投票の時には、委員個人ではなく、日本国内委員会である。せっかく委員個人が書いたドラフトも日本国内委員会で修正されたのでは何にもならず、このバランスを考えた委員の活動アサインメントが必要であろう。

コンピテンシーは確かにプロジェクトマネジメントそのものではないが、プロジェクトマネジメントを活性化するものである。当国際標準に取り込むことに反対の人もいるが、オフラインで話してみると、コンピテンシーの重要性については、否定する者はいない。この部分は、IPAとして、その活動を国際標準に反映する機会であり、関係国と会話を進めながら規格本文にも、アネックスにも積極的にドラフトを作成、提案する必要があるであろう。

# プロセス改善を成功させるためのESxRの使い方

三菱電機メカトロニクスソフトウェア株式会社 和歌山支所  
技術1課

岩橋 正実

ESPR( Embedded System development Process Reference )、ESMR( Embedded System project Management Reference )、ESCR( Embedded System Coding Reference )を用いてソフトウェア開発プロセス標準を構築しただけでは、QCD(品質、コスト、納期)の改善に繋がらない。対象組織の課題を分析して課題解決のためのタスクの計画/管理と実行スキルの向上が重要である。本稿では、QCD改善を目的に特定の組織に対するプロセス改善の事例としてソフトウェア開発プロセス標準の構築とその運用方法について解説すると共にESxR( Embedded System development eXperimental Reference ) / ETSS ( Embedded Technology Skill Standards )の活用方法を紹介する。

## 1. はじめに

インターネット等の社会インフラの加速度的な発展に伴い、組込みソフトウェアが急激に高機能化、多様化した結果、品質を確保するのが困難な状況になってきている。このような状況を改善するためにESxRが開発され、展開が進んでいる。

ESPRには、開発プロセスと支援プロセスがあり、それぞれのプロセスに対してアクティビティとタスクが定義されている。支援プロセス [ SUP1 プロジェクトマネジメント ] のタスク [ SUP1.1 プロジェクト計画書作成 ] の詳細がESMRに定義されている。開発プロセス [ SWP4 実装及び単体テスト ] のタスク [ SWP4.3 実装及び単体テスト結果の確認 ] の詳細がESCRに定義されている。そしてETSSではこれらの開発プロセスと支援プロセスに必要なスキルを定義している。特定の組織を対象としたソフトウェア開発現場のプロセス改善に、これらをどのように活用したかについて述べる。

## 2. 導入事例

### 2.1 標準プロセスの構築

ESxRに限らず標準化されたリファレンスをそのまま社内標準に適用してもQCDの改善には繋がらない。対象組織の課題を分析して課題を押しさ込むことでQCDの改善に繋がる。課題の分析はテストやレビューで検出された不具合の要因、作り込んだアクティビティ、検出したアクティビティ、1kline当たりの成果物や不具合の量等から課題を分析する。課題が見えて初めてどのようなタスクの実行を強化すべきかが見えてくる。

課題を整理すると様々なパターンの課題が発生していることに気付く。組込みソフトウェアの開発は、開発担当者毎に開発作業の内容が異なり、成果物(ソフトウェアのアーキテクチャやドキュメント)が異なっている。不具合が発生しても特定の不具合には歯止めをかけることができるがソフトウェア開発プロセスが不安定であるため異なるパターンの不具合が発生する。開発のプロセスを安定させるためには、組織のソフトウェア開発プロセス標準と不具合のパラツキを抑えるためのタスクの設定が重要である。

ある組織を対象に、過去に流出した不具合の分析を実施した。全体の15%が仕様自体の不具合で、70%が設計、15%が実装の不具合である。ソフトウェア開発部門で改善可能な設計と実装の不具合を分析して、不具合の流出を防止できるタスクの設定を実施した。そして、継続的なプロセス改善を進めるために、プロジェクト完了後に開発段階で発生した不具合を分析し、必要に応じタスクの追加更新を実施し、予防処置を継続して実施している。さらに、安定して開発を進めるために、要求から設計、実装への変換と、要求から試験設計への変換をパターン化して実行タスクとして組込みトレーサビリティを確保

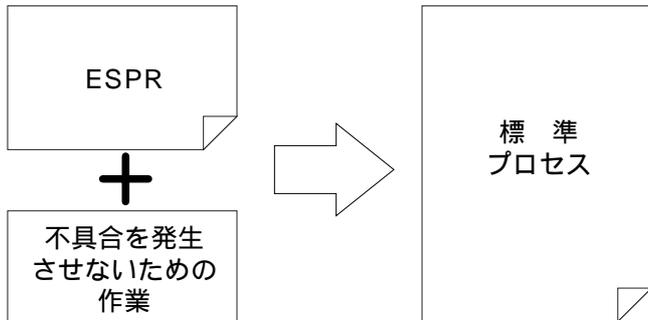


図1 標準プロセスの構築

した。これにより作り込まれる不具合もパターン化するため、課題を発生させなくするためのタスクの実行が有効に機能する。

次に、上記のソフトウェア開発プロセス標準の構築にESPRをどのように活用したかについて述べる。ESPRには組込みソフトウェア開発に必要なタスクを設定している。対象組織の不具合を分析してESPRを参照して不具合の流出防止に有効なタスクの設定を実施した。不具合を未然に防ぐために、どのアクティビティでどのような作業を実施すべきかをESPRから読み取り、必要なタスクを設定した。また、ESPRのアクティビティごとに対象組織に不足しているタスクを検討して必要に応じてタスクの設定を実施した(図1)。ESPRではタスクを複数のサブタスクに分離しているが、不具合を再発させない作業レベルのタスクを設定するために、サブタスクよりも詳細化したタスクを設定した。これにより、構築したソフトウェア開発プロセス標準を実行することで不具合の作り込みが抑制されQCDの改善が望める。

## 2.2 タスクの選択実行

組込みソフトウェアの開発では、既存製品への変更開発や新規開発等、開発の規模が様々であるため、ESPRで定義されたタスクの選択実行が必要になる。さらに、物理現象をコントロールするような組込みソフトウェアの開発では、最適な制御を実現するために何度も仕様変更

アクティビティ	タスク	サブタスク	開発規模			計画	実績
			新規	追加	変更		
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							
.....	タスク						
.....	タスク						

図2 タスクの選択実行と計画管理

が発生する。この変更のスピードを上げて仕様構築を加速させる必要があるが、このような変更作業を実施する際にESPRで定義された全タスクを実行することは工数が掛かり過ぎて実用的な使用方法ではない。このような状況に対応するために、開発の規模、作業フェーズでタスクを選択実行する。開発の規模は、新規、機能追加、機能変更で必要なタスクを設定する。さらに、作業フェーズとしては、開発の開始フェーズとしてソフトウェアアーキテクチャ設計などの基本設計部分を実施、中間フェーズでは仕様構築段階であるので実装及び単体試験を繰り返し実施する。そして、最終フェーズではソフトウェア要求定義から再度設計検証を実施して品質を確保して対象組織に合った工程設計を実現させた。

ESPRのアクティビティ、タスク、サブタスクから対象組織に必要なタスクの選択方法に関しては前節で解説したので、本節のタスクの選択実行に関して新たにESxRを活用した内容はない。自組織に合致したソフトウェア開発プロセス標準を定義して開発規模や作業フェーズに応じてタスクを選択実行する枠組みを構築することでプロセス改善の効果が期待できる。

## 2.3 プロセス実行計画と管理

プロジェクトの立ち上げ時は、ソフトウェア開発部門でソフトウェア開発計画書を作成して、以下のような項

目の計画管理を実施して開発を進めている。

- ・プロジェクトの概要と組織の編成と役割
- ・コア資産、環境の調達計画
- ・リスク計画管理
- ・ホールドポイントの計画と工程管理
- ・開発プロセス計画管理と計測管理

この中で最も重要な項目は、開発プロセス計画管理である。2.2節で解説した「開発規模に応じた実行タスクのパターンを計画時に選択してタスクの実行を管理すること」で品質を確保させている(図2)。これにより、対象組織での課題を解決するタスク実行をプロジェクト毎に計画/管理することになり、着実に効果を上げることができる。

#### 2.4 ESxRの具体的な活用

ソフトウェア開発プロセスの改善にESxRを適用する目的は、開発作業の品質レベルを一定のレベルに維持させ予算内で、かつ納期通りに開発作業を完了させることにある。作業内容が一定のレベルで実行されると各作業に必要な時間も安定するため、品質、コスト、納期を計画通りに実行することが可能になる。前節まではソフトウェア開発プロセスの構築に関するESxRの適用に関して解説したが、本節では、運用時の適用についてESPRを軸として、ESMR、ESCR、ETSSをどのように活用しているかを解説する。

ESPRのプロジェクト計画時に、ESMRをリファレンスとして使用する。実行する各アクティビティの内容や注意点をESPRから読み取り開発を進める。プログラムの実装段階ではESCRをリファレンスとして使用して実装精度を上げている。組織全体へのソフトウェア開発プロセス標準の説明は実施するが、十分な教育をするには多くの時間を必要とする。そのため、対象組織内の教育を実施しながらESxRを参照させてスキルの向上を図っている。さらに、SEC主催のESxR関連セミナーに参加させて組織全体のプロセス実行スキルの向上を進めている。

#### (1) ESPR

受注、発注先のアクティビティにバラツキが発生するため作業範囲の取り違いや作業品質のバラツキが発生する。そこで、ESPRを使用してソフトウェア開発作業のバラツキをなくすため、2.1節で解説した組織のソフトウェア開発プロセス標準を使用して受発注の作業内容と成果物を明確にしている。これにより、異なる発注先でも一定レベルの品質を確保することが可能になった。

組織のソフトウェア開発プロセス標準の構築にもESPRは有効である。部署間で異なるアクティビティ名称を使用している組織は、組織標準を構築してもアクティビティの捉え方が異なる場合が多い。そこで、各部署のアクティビティとESPRの紐付けの検討を進めている。これにより組織全体としての整合がとれるため、組織標準を構築する際にも有効活用が期待される。

#### (2) ESMR

ESPRの支援プロセスであるプロジェクト管理のタスクを実行時に参照させている。2.3節のプロセス実行計画と管理で解説したソフトウェア開発計画の設定時にESMRを参照させてプロジェクト管理の実行精度の向上を進めている。

#### (3) ESCR

ESPRの実装及び単体テストのアクティビティの実行精度を向上させるために実装作業実施時にESCRの信頼性の章を参照させてコード品質の改善を進めている。現時点では参照のみであるが、将来は、コーディング規約に取り込む予定である。実装のサブタスクとして規約の確認タスクを設定してサブタスクの実行を管理することでコード品質の改善を進める計画である。これにより、実装上の課題で不具合が流出した場合は、コーディング規約の更新を実施して、サブタスクの実行管理で継続的な品質改善が可能になる。

#### (4) ETSSとの連携

ESPR、ESMR、ESCRの参照で期待できるのはタスクの実行品質の向上である。ソフトウェア開発プロセス標準が定義されてもタスク実行スキルが低ければQCDの改善は望めない。そのためESxRとETSSとを連携させる意味は非常に大きい。ESPRとETSSのアクティビティの名称は異なっている。そこでESPRの開発プロセスと支援プロセスをETSSの開発技術と管理技術のアクティビティに置き換えてESPRを軸にETSSのスキル項目を定義した。ESPRではアクティビティ内のタスクの実行内容を詳細に定義している。さらに、必要な手法及びツールが定義されているため、ESPRからスキル項目の抽出は容易に実現できスキル評価項目が安定するため組織間でのスキル評価のバラツキがおさえることが可能になった。現在は、品質/生産性を分析して強化すべきスキルを分析定義して、戦略的な教育の取り組みを進めている。

#### 2.5 プロセス改善の効果

プロセス改善を実施した対象組織の改善前と改善後のプロセス改善の効果を分析する。プロセス改善前と改善後のアクティビティの工数割合を分析すると改善後は、ソフトウェア分析定義、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の上流アクティビティの作業割合が約3倍に上がっている(図3)。品質に関してはソ

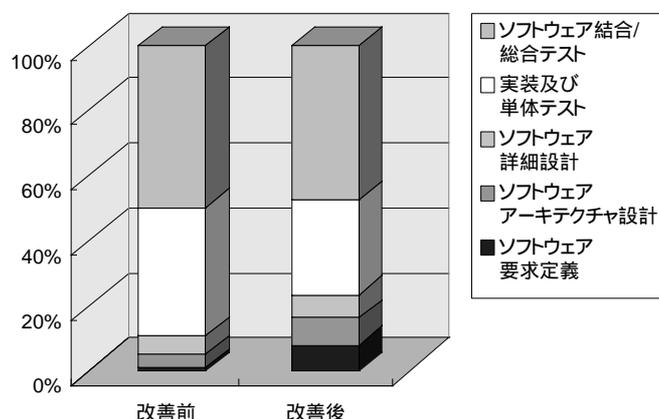


図3 アクティビティ単位の工数割合

フトウェア開発作業で不具合を作り込んでいたアクティビティは、設計が70%、実装が15%で、仕様の不具合が15%であったが、改善後は、ソフトウェアの開発作業で作り込んだ不具合は現時点では0%となった。本状態を継続するにはプロセスの計画管理の徹底と、タスク実行スキルを向上させるためのDR時のOJTの徹底が必要不可欠である。

## 3. まとめ

ソフトウェア開発プロセスは、ソフトウェアの品質を確保するために必要なタスクの集合である。ソフトウェア要因の不具合がどのようにして作り込まれたかを分析して、タスクの実行により不具合を発生させなくすることが重要である。さらに、ソフトウェアの開発を定型作業に近づけることで、品質特性が安定してプロセス改善が有効に機能するようになる。そして、ESPRを軸にETSSを定義することで、プロセスの実行の側面とスキルレベルの二つの視点から管理改善が可能になり品質と生産性の向上が期待できる。

#### 参考文献

- [ESPR] IPA SEC：組込みソフトウェア向け開発プロセスガイド, 翔泳社, 2006
- [ESMR] IPA SEC：組込みソフトウェア向けプロジェクトマネジメントガイド[計画書編], 翔泳社, 2006
- [ESCR] IPA SEC：組込みソフトウェア向けコーディング作法ガイド[C言語編], 翔泳社, 2006
- [ETSS] IPA SEC：組込みスキル標準ETSS概説書, 翔泳社, 2005

# 車載OS開発に関するESMRの活用事例

株式会社ヴィッツ 組込みソフトウェア開発部  
リアルタイムシステム課

長江 宣宗

株式会社ヴィッツでは、経済産業省と中小企業基盤整備機構の委託事業「戦略的基盤技術高度化支援事業」（通称「サポイン事業」）の「機能安全対応の自動車制御用プラットフォームの研究開発」プロジェクトに取り組んでいる。また、経済産業省の産学連携ソフトウェア工学実践事業「高信頼組込みソフトウェア開発」をJasParが委託を受け、自動車用制御基盤ソフトウェアの仕様策定等を実施するプロジェクト（JasPar / 国プロ）のOS部分の仕様策定・開発メンバとして参加している（なお、株式会社ヴィッツはJasParプロジェクトの正会員である）。

## 1. 事例の概要

サポイン事業では、我々は機能安全の国際規格であるIEC 61508の安全度基準（SIL）のSIL3に対応したOSの開発を行うことを目標としていた。しかし、機能安全対応で、IEC 61508の製品認証を得るためには、総数50弱のドキュメントを作成する必要があることがわかった。その内の実施計画書の作成について、国内での車載OSの開発分野での適用例が見つからず、参考になる統一された書式がなかった。しかし、IPA/SEC発行「組込みソフトウェア向けプロジェクトマネジメントガイド[計画書編]」[ESMR]の存在を知って、機能安全対応での適用可能性を検討した。そこで、IEC 61508の規格が要求する項目を機能安全系の項目、機能安全対応には関係しないが、実施計画書には必要な内容を非安全系として分類して、ESMRの全項目との検討を行った。すると、全記載項目31項目中、機能安全系の項目が21項目、非安全系の項目が9項目あり、適用が必要な項目が合計で30項目に及ぶことがわかった（表1）。

また、JasPar / 国プロでは、高信頼組込みソフトウェア開発の一環として、SECの成果物を評価する必要があった。そこで、ソフトウェア開発上でESMRや、他にも「組込みソフトウェア向け開発プロセスガイド」[ESPR]等を評価するために、今回のプロジェクトに適用実施した。

弊社ではESMRとESPRについて今回は車載OS開発の分野での有用性を確認したいと考えた。

表1 ESPRとIEC 61508の比較

大項目	小項目	計画書記載
プロジェクト概要	プロジェクトの目的	
	プロジェクトの目標	
	目標達成のための方針・手段	
	プロジェクトの範囲	
	プロジェクトの前提条件	
	プロジェクトの成果物	
	スケジュールと予算	
参照・定義情報	計画の更新	
	参照	
プロジェクト体制	定義	
	製品開発プロジェクト体制	
	外部インタフェース	
リソース計画	ソフトウェア開発プロジェクト内部体制	
	役割分担	
	開発規模と工数の計画	
	人員計画	
	設備、機器等の調達計画	
作業計画	プロジェクトの人員研修計画	
	予算計画書	x
	開発作業の洗い出し	
	開発作業の順序付け	
品質保証計画	開発作業担当者の割付	
	作業計画	
	品質目標	
安全保証計画	品質保証の体制と仕組み	
	品質保証に関する主要なイベント	
	すべての運用及び保全計画	
リスクマネジメント	すべての安全妥当性確認計画	
	すべての設置及び引渡計画の作成	
	リスクマネジメントの方針と仕組み	
	リスク一覧表	

:機能安全が関係 :機能安全には直接関係しないが必要 x:不要

SIL : Safety Integrity Level

## 2. 事例の特徴

ESMRを用いた計画書作成の導入の事例として各項目についての詳細を述べると共に、ESMR導入の際に気付いた点や問題点を紹介する。

### Chapter 1 . プロジェクトの概要について

導入当初、目的に「車載OSの開発」、目標も「OSの開発」とし、長々とした文章で説明する等、わかり難い内容となってしまった。目標は簡単に書き、開発期間や品質で当然と思えることでも書いた方がわかりやすく人に伝わりやすい。

問題点：目標や目的の項目が明記されていないため、自由な記述ができる一方で、記述すべき具体的内容がわかり難いので、より簡明な説明が必要である。

### Chapter 2 . 参照・定義について

計画書内で使われる用語については記述があった方がよいが、用語定義についてはプロジェクト中にも更新が必要となる可能性があるため、問題点の解決方法としては、メンバが自ら追加・参照できるようにすることで、常に最新の情報を共有した方がよい。また、プロジェクト中に記載情報の更新が必要になってくると、新旧の情報のずれが発生する可能性がある。さらに、他プロジェクトでも相互に参照できるように、計画書とは別文書にすることで、社内共有できる形式にした方がよい。

### Chapter 3 . プロジェクトの体制について

プロジェクト体制の全体の把握と外部との接点を明確化できる。

問題点：リソース計画に情報がリンクしているので、リソース計画の変更が生じるとこの章の内容も更新が必要である。

### Chapter 4 . リソース計画について

プロジェクト管理ソフトで代替して記述することで、進捗管理も一元化できて効率よく進めることができた。更にIEC61508で規定する力量確認の項目を追加したが、他のプロジェクトでも力量を確認する項目は必要と思われるので、ESMRの項目に追加することを勧める。

### Chapter 5 . 作業計画について

実施内容の洗い出しや、作業期間、リソースの割り当て等が整理して記述できる。

問題点：類似する内容や情報が複数のシートに存在するので、情報をまとめて変更箇所を減らして欲しい。

### Chapter 6 . 品質保証計画について

目標とする品質レベルが明確になる。

問題点：達成目標値としての具体例がなく、記述し難いので、品質目標値の設定方法についての詳細説明があるとよい。

### Chapter 7 . リスクマネジメントについて

トラブル発生時の影響が明確化できる。

問題点：発生頻度、重要度の5段階評価は定量的な尺度との関連が示されていないので、必要に応じて他の尺度を用いることも検討する。

## 3. 今後の展開

今回の取り組みでは、ESMRに規定されている項目を計画書に記載していくことで、円滑に作成することができた。また、既に自社の計画書の雛形が整備されている企業にとっても、自社で日頃作成している計画書とESMRの規定している計画書を照らし合わせて、記載項目を検討する契機とすることができる。ESMRを用いて計画書を作成した結果、異なるプロジェクト間でも文書のフォーマットを統一して記載することが容易にできたことがわかった。しかし、今後の文書改訂時にも、ESMRのフォーマットを統一のとれたものに維持していくには、文書管理が必要であると思われるので、次年度以降のプロジェクト内容に今回検討した点をフィードバックして文書整備を行いたいと考える。

#### 参考文献

[ESMR] IPA SEC：組込みソフトウェア向けプロジェクトマネジメントガイド[計画書編]，翔泳社，2006

[ESPR] IPA SEC：組込みソフトウェア向け開発プロセスガイド，翔泳社，2006  
[SEC-Webサイト] <http://sec.ipa.go.jp/>

# BOOK REVIEW

## ソフトウェアエンジニアリング講座 <1> <2> <3> <4>

ITトップガン育成プロジェクト 著

ISBN : 978-4-8222-8305-6 日経BP社刊  
A5判・350頁・定価3,360円(税込)・2007年2月刊

### 全巻通読で、はじめてできるベースライン

全1,264頁、37章、各章毎に専門家、すなわち延べ約40人の書き手を動員した大著である。ソフトウェアエンジニアリングの全領域について記述した1冊の書を読みたいというニーズに応えてくれる、頑張れば読み通せる稀有な書になっている。さすがに広汎な分野のカバーに4巻構成となってしまったが、本書には脚注はあっても巻末に参考文献が示されていない。つまり完結した読みものを目指している。リンク集のようなSWEBOOK(ソフトウェアエンジニアリング基礎知識体系)へのアンチテーゼといえる。参考文献がいくら並んでいても読者はそれらを全部読むことなど出来ないのだから、この努力は素晴らしい。この分野では多くの人が「断片の知識」「断片の経験」をもとに「断片の議論」をしている。断片同士が

同期すれば深い議論になるが、周りの者に理解できない、いわゆる「オタク」議論に陥り、同期しなければ文字通り「すれ違い」議論になる。まず皆で本書を通読して認識のベースラインをつくり、そこからパランスのとれた議論を始めたいものである。第1巻でこの分野の基礎を概観した後、第2巻で「システム開発プロジェクト」、続いて「プログラミング」、「オープンシステム技術」となるが、通読すると各巻の示す領域で実に多彩な活動が行われていることがよくわかる。本書のように全体を俯瞰できる著作を機会に、皆のベクトルが揃ってゆくことを展望したい。

(神谷芳樹)



## 企業進化論 情報創造のマネジメント

野中 郁次郎 著

ISBN : 978-4-532-19111-5 日本経済新聞社刊  
A6判・332頁・定価840円(税込)・2002年2月刊

### 形式知の共有拡大とスキルは企業進化の要

みなさんの会社は、エクセレントカンパニーであるといえますか。

日本のソフトウェア開発企業において優良企業だなど感じることや、マスメディア等で報道されることは少ない。

この書籍には、企業が進化し、競争力や価値を高めるためのエッセンスが豊富に盛り込まれている。MBA的な経営分析だけでなく、企業文化やセルフオーガナイズング等、企業の地力に関する解説は新鮮であり、自社での戦略において有意義な知識を提供してくれる。

この「有意義な知識」は企業の進化を支える源ともいえる。巻末の補論として掲載されるSECIモデルは、知識の流通が企業進化の源であることを証明している。

暗黙知を共有すること、暗黙知を形式知とすること、形

式知を広く共有すること、そして形式知を暗黙知として腹に落とし込むことがサイクルとして表わされている。

開発現場のスキルをOJT等によって共有することや、スキルである暗黙知をソフトウェアエンジニアリングとして形式知化することが必要である。そして、形式知をグループという小さい単位でなく、もっと大きな範囲で共有することの必要性を説く。まさに、SECでのETSSをはじめとする各種取り組みが重要であることを再認識させられる書籍である。余談であるが、この書籍は文庫本であり、持ち運びしやすいことも大きなメリットである。

(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

開催年月	開催日	イベント名	主催	開催場所	URL
2月	26(火)	【SEC主催セミナー】 プロジェクト「見える化」(名古屋)	IPA/SEC	愛知県名古屋市・ 名古屋国際会議場	<a href="http://sec.ipa.go.jp/">http://sec.ipa.go.jp/</a>
3月	12(水)	【SEC主催セミナー】 組込みソフトウェア向け ESPR/ESMR概説(札幌)	IPA/SEC	北海道札幌市・ さっぽろ産業新興センター	<a href="http://sec.ipa.go.jp/">http://sec.ipa.go.jp/</a>
	19(水)	組込み技術者へのスキルチェンジ ～エンタプライズ系技術者から組込み系 技術者へのスキルチェンジの勘所～	財団法人 にいがた産業創造機構	新潟県新潟市・ NICOプラザ会議室	<a href="http://www.nico.or.jp">http://www.nico.or.jp</a>
5月	14(水)～ 16(金)	SODEC / 第17回 ソフトウェア環境開発展	リード エグジビション ジャパン 株式会社	東京都江東区・ 東京ビッグサイト	<a href="http://www.sodec.jp/">http://www.sodec.jp/</a>
	14(水)～ 16(金)	ESEC / 第11回 組込みシステム開発技術展	リード エグジビション ジャパン 株式会社	東京都江東区・ 東京ビッグサイト	<a href="http://www.esec.jp/">http://www.esec.jp/</a>
	27(火)～ 28(水)	IPAX2008 SEC Forum 2008	IPA	東京都文京区・ 東京ドームシティ・プリズムホール	<a href="http://sec.ipa.go.jp/">http://sec.ipa.go.jp/</a>
6月	5(木)～ 6(金)	Embedded Technology West 2008 / 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・ インテックス大阪	<a href="http://www.jasa.or.jp/etwest/">http://www.jasa.or.jp/etwest/</a>
	5(木)～ 6(金)	ソフトウェアテストシンポジウム 2008 関西(JaSST'08 Kansai)	特定非営利活動法人 ソフトウェア テスト技術振興協会(ASTER) JaSST'08 Kansai 実行委員会	大阪府大阪市・ インテックス大阪	<a href="http://www.jasst.go.jp/">http://www.jasst.go.jp/</a>
7月	17(木)～ 18(金)	SPES2008	社団法人 情報サービス産業協会	東京都江東区・ 日本科学未来館	<a href="http://www.jisa.or.jp/seminar/spes2008/">http://www.jisa.or.jp/seminar/spes2008/</a>
9月	4(木)～ 5(金)	第27回ソフトウェア品質シンポジウム	財団法人 日本科学技術連盟	未定(東京都内)	<a href="http://www.juse.or.jp/">http://www.juse.or.jp/</a>
10月	1(水)	情報化月間記念行事	IPA	東京都港区・ ANAインターコンチネンタルホテル 東京	<a href="http://www.ipa.go.jp/">http://www.ipa.go.jp/</a>
	28(火)	IPA Forum 2008 SECコンファレンス (SEC journal論文発表会)	IPA/SEC	東京都港区・ 明治記念館	<a href="http://sec.ipa.go.jp/">http://sec.ipa.go.jp/</a>
	29(水)～ 31(金)	ESS2008 (組込みシステムシンポジウム2008)	情報処理学会 組込みシステム 研究会 IPSJ SIGEMB	東京都渋谷区・ 国立オリンピック記念青年総合 センター	<a href="http://www.ertl.jp/SIGEMB/">http://www.ertl.jp/SIGEMB/</a>
11月	19(水)～ 21(金)	Embedded Technology 2008 / 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	<a href="http://www.jasa.or.jp/et/">http://www.jasa.or.jp/et/</a>

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。

## ET2007報告

SECは、2007年11月14日(水)～11月16日(金)パシフィコ横浜にて開催された「Embedded Technology 2007」にIPAブースとして他部署と共に出席し、パネル展示、ブース内セミナー、EPMツールのデモ、産業実態調査を実施いたしました。また、アネックスホール及び会議センターにて、「組込みシステム開発地域活用セミナー」として講演とパネルディスカッションを、「IPAセミナー」としてセキュリティセンターとSECの講演を行いました。

### < SECセッション >

11月14日 15:00～17:00 「組込みシステム開発地域活用セミナー ～高度人材拠点・分散開発拠点としての地域の活用～」  
講演「『企業立地促進法』と地域における組込み産業育成への活用」：経済産業省経済産業政策局 地域経済産業グループ 立地環境整備課 課長補佐(企画調整担当) 祝谷和宏氏  
パネル「地域における組込み人材育成・産業育成の取り組み～課題と工夫～」  
モデレーター：SEC組込み系プロジェクトサブリーダー 田丸喜一郎  
パネリスト：経済産業省経済産業政策局 地域経済産業グループ 立地環境整備課 課長補佐(企画調整担当) 祝谷和宏氏、青森県商工労働部新産業創造課 情報産業振興グループリーダー 佐々木明夫氏、塩尻市経済事業部商工課 塩尻インキュベーションプラザ担当 太田幸一氏、株式会社名古屋ソフトウェアセンター 代表取締役専務 高嶋雅樹氏、株式会社セイデン 代表取締役社長 有賀一師

氏、東海大学専門職大学院組込み技術研究科 研究科長 教授 大原茂之氏  
参加人数：91名

### < IPAセミナー >

11月15日  
10:00～12:00 「IPAセキュリティセンター」  
講演「セキュアな組込みシステムに向けたIPAの活動について」：IPAセキュリティセンター 情報セキュリティ技術ラボラトリー ラボラトリー長 小林偉昭氏  
講演「組込みシステムの具体的な脅威と対策の提案」：IPAセキュリティセンター 情報セキュリティ技術ラボラトリー 非常勤研究員 鶴飼裕司氏  
13:00～16:50 「SEC2006年度成果レビュー」  
講演「高品質な組込みソフトウェアを創るには」：IPA SEC組込みプロジェクト研究員 猪狩秀夫  
講演「ETSSがスキルマネジメントの常識を覆す」：IPA SECリサーチフェロー 大原茂之氏  
参加人数：198名

なお、SECで主催した各セミナーの資料のうち講師の許諾を得たものについては、SEC-Webサイト・イベント情報・2007年11月14日「Embedded Technology 2007」(<https://sec.ipa.go.jp/event/2007/20071114.php>)にて公開しています。

## 編集後記

SECは、昨年10月で設立から3年を経過し、多くの方に支えられての三年間の成果を三周年成果報告会で報告しました。その間「SEC journal」も号を重ね、本号で13号となりました。論文誌としてのプレゼンスも少しずつ高まり、投稿論文数はまだ多いとはいえませんが、掲載論文を楽しみにしている読者も着実に増えてきています。また、SEC-Webサイトの利用登録者も15,000人を超え、SEC journalのダウンロード数も10,000件(2006年4月～2008年1月)と、読んで頂いている方が増えてきていることが実感でき、感謝しております。

本号では、ソフトウェアエンジニアリング ベストプラクティス賞をご紹介いたしましたが、受賞された各組織は、SEC成果をソフトウェア開発現場での課題解決に向けて自分たちの環境に適したかたちにして、地道に展開・適用され、効果を出しておられます。本号に掲載した受賞各組織の取り組みでは、その様子をご紹介いたしており、必ず読者の琴線に触れるところがあると確信しています。「SEC journal」でのベストプラクティス掲載についても、論文と同様に力を入れていきたいと思っております。

本journalに対してのご意見とSECへの情報提供・問題提起等には、SEC-Webサイト(<http://sec.ipa.go.jp/>)内の「SECへのお問い合わせ」をご利用ください。

(Y.O)

## SEC journal 編集委員会

編集委員長

奥 保正

編集委員(50音順)

猪狩 秀夫

大山 眞弓

田丸喜一郎

樋口 登

神谷 芳樹

門田 浩

渡辺 登



### お詫び

本誌No.12において下記の誤りがありました。お詫びしてご訂正申し上げます。

- ・26頁 右段1つめの論文  
誤: 2006年優秀賞受賞 正: 「SEC journal」創刊記念論文 優秀賞受賞論文
- ・26頁 右段4つめの論文  
誤: 賞の記載なし 正: 2006年優秀賞受賞論文

SEC journal® 第4巻第1号(通巻13号) 2008年2月29日発行

© 独立行政法人 情報処理推進機構 2008

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。

# SEC journal 論文募集

独立行政法人 情報処理推進機構  
ソフトウェア・エンジニアリング・センターでは、  
下記の内容で論文を募集します。

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

## 論文テーマ

ソフトウェア開発現場のソフトウェア・エンジニアリングをメインテーマとした実証論文

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

## 論文分野

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

## 論文の評価基準

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

## 論文賞

「SEC journal」では、毎年「SEC journal」論文賞を発表しております(今回は2007年10月30日SECコンファレンス)。受賞対象は、「SEC journal」掲載論文他投稿をいただいた論文です(論文賞は最優秀賞、優秀賞、SEC所長賞からなり、それぞれ副賞賞金100万円、50万円、20万円)。

## 応募要項

### スケジュール

- ・15号(2008年7月発行)
  - 応募締切 2008年5月23日
  - 投稿締切 2008年5月30日
  - 採否通知 2008年6月20日

- ・16号(2008年10月発行)

- 応募締切 2008年7月18日
- 投稿締切 2008年7月25日
- 採否通知 2008年8月22日

採録決定後、1週間程度のカメラレディ期間があります。

詳細は別途通知されます。

採録の場合には「SEC journal」への掲載およびSEC-Webサイトやイベント等で発表します。

詳しくはSEC-Webサイトよりお問い合わせください。

### 提出先

独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター内「SEC journal」事務局

eメール: [sec-ronbun@ipa.go.jp](mailto:sec-ronbun@ipa.go.jp)

### その他

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

## SEC journalバックナンバーのご案内

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



No.1

No.2

No.3

No.4

No.5

No.6

No.7

No.8

No.9

No.10

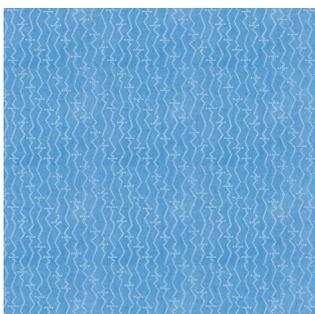
No.11

No.12

SEC Journal No.13  
第4巻第1号(通巻13号)  
2008年2月29日発行 ©独立行政法人 情報処理推進機構

編集兼発行人 千113-6591 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  
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

