2006年1月31日発行
第2巻第1号（通巻5号）
ISSN 1349-8622

SEC journal

5

Software Engineering Center

第4世代のテスト・プロセス
山浦 恒央

企業横断的収集データに基づく ソフトウェア開発プロジェクトの工数見積り
大杉 直樹, 角田 雅照, 門田 暁人, 松村 知子, 松本 健一, 菊地 奈穂美

通信ソフトウェア開発におけるプロセス改善のための フィールド品質に注目した主要な活動要因の抽出
菊地 奈穂美, 安藤 津芳, 水野 修, 菊野 亨

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

SEC journal
Software Engineering Center
No.5目次

巻　頭　言

時代の要請とSEC

経済産業省　商務情報政策局
情報処理振興課長
鍛治　克彦

政策現場の最前線から見えるもの

経済産業省では、2005年8月から産業構造審議会（産構審）に情報サービス・ソフトウェア小委員会を設置し、情報産業を取り巻く諸課題について検討を開始しました。この議論の過程で、情報産業を取り巻く古くて新しい問題の数々が浮上してきています。それらを大きく整理すると、「可視化」、「人材」、「グローバル」の3つのキーワードに収斂してくるように思います。

情報システムの可視化

今日の情報システムは、経済社会の活動を支えるインフラであり、不具合やセキュリティの脆弱性は、日々の活動に大きな影響を及ぼします。また、グローバルな大競争時代の中で、企業経営の不断の革新が重要課題であり、その成否を握るのがERP、SCM、CRM等の最先端の情報システムであると考えられております。

このように、情報システムの性能・品質が企業経営や経済活動に与える影響の増大に比例して、その開発・運用のあり方が改めて問われるようになりました。所謂、「人月工程」主義の積算を万能視してよいのか、あるいは発注側・受注側の双方がお互いの手法と能力に必ずしも全幅の信頼を置けない現状でいいのか、という問題意識が高まり、産構審においても活発な議論がなされています。

ここでのポイントは、情報システムの価値の「可視化」ということです。SECのエンタプライズ系部会が取り組んでいる定量データ収集と分析、見積手法の開発、開発プロセス共有化などの各事業は、ユーザとベンダの情報共有、協働の場を提供し、情報システムの可視化を実現する上での最も有効な手だてであり、今後も具体的な成果を生み出すことが期待されています。

人材の高度化

情報システムの可視化問題と並んで、産構審で多くの識者が指摘する事項が、情報システムやソフトウエアの開発・運用に携わる人材の質の向上です。

IPAの中には、情報処理技術者試験、ITスキル標準などの優れたインフラが存在します。SECは、組込みスキル標準（ETSS）の開発や先進ソフトウエア開発における産学官連携の分野で主体的な役割を担っていますが、これに加えて情報処理技術者試験センター、ITスキル標準センター等との連携強化により、我が国情報産業の人材育成分野でも益々貢献することが期待されています。

グローバルな展開

インドや中国のIT産業の躍進に見られるように、我が国情報システム市場は既に「鎖国」できる時代ではありません。また、組込みソフトウエアに関しては、各国が戦略分野と位置づけており、同分野での我が国の優位性も決して安泰ではありません。産構審の議論でも危機意識が高まっております。

SECで進められている様々な部会での活動には、各々産学官から200名を超える英知が結集し、エンジニアリング、プロジェクトマネジメント、統計など広範な分野で、具体的な成果を挙げてきています。SECのグローバルネットワークも活用して、グローバル大競争時代を勝ち抜く情報産業の育成が一層急務となっています。

成果普及の本格展開を

こうして見てみますと、今日の情報産業を取り巻く主要課題の多くがSECによって果敢に取り組まれていることがわかります。本年はSECの活動も2年目を迎え、今まで以上に積極的に情報発信するとともに、産学からの要望・意見にしっかり応答すべき時期に至りつつあります。

SECの一層の活躍を祈念するとともに、経済産業省としても十分にサポートしていきたいと思います。

複雑化と短期開発要求が進展する組込みソフトウェアの課題を解決する道を示す

組込みソフトウェアは複雑に、そして巨大なものとなっている。その一方で、産業界からは組込みソフトウェアの短期開発要求の度合いが増している。今回は、欧州自動車産業で生まれ、高信頼性ソフトウェア開発のガイドラインとして知られるC言語のコーディング規約「MISRA-C」を作成したMISRAプロジェクトのマネージャであり、MISRA-Cステアリングメンバー委員長のDavid Ward氏とIPA/SECの鶴保征城所長が、組込みソフトウェアをめぐる課題に鋭く切り込み、その解決策について語った。SECは、ソフトウェア工学のメタ的な視点から複数のプログラミング言語を包括することをねらっているのに対し、MISRAは、単一言語からスタートしてC++やモデリング言語へ進もうとしている。そのアプローチに違いはあるが、ゴールは同じということで両者の見解は一致した。今後、両者が交流していくことが考えられ、その成果に期待がかかる。

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

鶴保　欧州における自動車業界や組込みソフトウェアの状況は、日本でも注目をしております。まずは整理の意味も含めて状況をお話しください。

Ward　欧州の自動車業界における組込みソフトウェアの課題を挙げると、車両に搭載される組込みソフトウェア市場が急激に拡大していることです。例えば、欧州のラグジュアリーカーの場合、100個以上のMPUが組み込まれています。金額ベースでも、電気的及び電子的システムが自動車の30～40%を占めていると考えられます。また、自動車の機能の85%くらいがソフトウェアによってコントロールされています。

また、技術的な課題もあります。自動車の制御はこれまでは自律的なもので、個々のエレクトロニックシステムが自動車の個々の機能を担当していました。しかし、現在、自動車の機能は分散システムで構成され、1つのコントローラが複数の機能を担当しています。また、アドバンストドライビングにおいては、車両間のインタラクションを概念としているので車両システムは複雑さが増しています。将来的には、複数の自動車が互いにインタラクションする中で形成される車両同士の分散システムネットワークを正しく運用していくことが課題となります。

自動車業界の課題は、そうした中で信頼性と安全性の高いエレクトロニックシステムを作り上げなければならないということです。かつ、自動車業界は短期間に高品質の自動車を製造しなければならないこと、また継続してコストを下げていかなければいけないという課題に直面しています。また自動車の生産にあたっては、サプライチェーンの一環として自動車メーカー及び様々な企業がシステムを構成しており、車両のシステムはどこかの企業1社だけで担えるものではありません。自動車に関する組込みソフトウェアの課題には以上の点があります。

組込みソフト技術者の育成が急務

鶴保　日本でもまったく同じ状況です。自動車業界や携帯電話業界等の産業界を対象とした組込みソフトウェアの実態調査によると、システム開発費全体に対して組込みソフトウェアが占める比率は40%程度に達しています。ソースコードを見ると、自動車の場合は制御だけで500万ステップ、携帯電話ではすでにそれを超えています。航空機の場合は、ソースコードが4,000万ステップに達しているのですが、そのうちの1/3を組込みソフトウェアが占めています。組込みソフトウェアのコードが膨大になっているのに対して、組込みソフトウェア技術者は17万人程度であり、リソースが足りないのが現状です。我々が直面しているのは、複雑なソフトウェアの安全性・信頼性をいかに高くするかということと、同時に生産性を高くすることです。

解決すべき問題は2つあります。1つ目は、ソフトウェアエンジニアリングの分野です。そこには、ソフトウェア工学もあれば分散処理技術の高度化もあります。2つ目の問題は人材の問題です。ターゲット技術が急速に変化する環境において、技術革新に対応できる人材を教育するシステムの確立が急務です。SECでは、この1年間、組込みソフトウェアに関するエンジニアリング手法の確立と人材の教育・確保という2つの問題に取り組んできました。

Ward　鶴保さんのお話は私どもが直面している問題と類似しています。そうした問題を解決する手法としては複数の考え方があると思います。最も大切なのはメソッドで、要求仕様に基づく形でソフトウェアを実装できるようにすることです。2点目として、特に自動車業界で関心が高いモデルベースの開発と、それによる自動コード生成を採用することです。これにより、もっと迅速にソフトウェアの開発を行えるようになると考えています。しかし、信頼性の高い、安全なシステムを構築していくためには、また新たな課題があり、それらを技術的に解決するよう、取り組んでいかなくてはならないと考えています。

鶴保　いろいろな問題を解決していかなければなりませんね。1つ目は、今、おっしゃったように設計から実装の問題ですね。そこに、モデルベースや自動生成という考え方を取り込むべきだと思います。2つ目は、プラットフォームという概念を作って、その上に共通化できる部品を蓄積して活用することが重要だと思います。これは、日本の場合、家電業界が取り組んでいる手法で、自動車業界はこれからという状況です。しかし、現在は家電業界も各社ごとのプラットフォームを利用しているので、業界としてプラットフォームやプログラミングインタフェースの標準化を図っていく時期にさしかかっていると言えるでしょう。3つ目は、コンカレントエンジニアリングの考え方を取り入れることです。日本の産業界、特に自動車業界はセットメーカーと部品メーカーはコンカレントに開発を進め、開発期間の短縮に成功しています。ソフトウェアの場合も、会社を超えてソフトウェアをコンカレントエンジニアリングしていく仕組みが必要だと思います。ソフトウェア業界では、コンカレントエンジニアリングの成功体験がないので、これからチャレンジしていくべきでしょう。最後に指摘したいのはこれらのテーマの解決をすべて支えるのは、人材だということです。

David Ward（デビッド・ワード）
1991年より英国MIRA（自動車産業研究所）に所属。現在は、MIRA電子機器部の先進技術部門グループのマネージャを務める。1995年より、MISRAのプロジェクトマネージャを兼務。MISRA開発ガイドライン、MISRA-Cプログラミングガイドラインの開発責任者。EMCとMISRA開発と普及活動を行っている。ケンブリッジ大学物理学科卒業。1991年ノッテインガム大学でEMC研究により理学博士号。英国物理学会会員、IEEE会員。

ですから、組込みソフトウェア技術者の育成を急がないといけません。

学生が産業界のニーズに触れられるカリキュラムを

Ward　まったくそのとおりだと思います。欧州の自動車業界でもいろいろなイニシアチブがあります。例えば、OSやI/Oドライバといったソフトウェアコンポーネントをソフトウェアアーキテクチャにおいて標準化することによって、エンジニアがアプリケーションソフトウェアの開発に専念できるようにする取り組みが行われています。そしてまた、人材育成も組込みソフトウェアの課題を解決する重要な要素だと考えています。

鶴保　SECでは、大学における組込みソフトウェアの教育について調査を行いました。その結果、学部及び大学院の修士課程において、組込みソフトウェアの教育を行っている大学は非常に少ないということが判明しました。特に、学部の学生が教えられているのは、ソフトウェア分野ではコンピュータ言語、ハードウェア分野では電子回路の単位数が圧倒的に多いということが明らかになりました。ソフトウェア工学や組込みシステムの教育は、コンピュータ言語や電子回路に比べて1/3や1/4以下しか教えられていないということがわかりました。この結果について我々は非常に大きな問題であると認識しています。

Ward　その状況は、欧州と似ています。欧州の大学の学部

MISRAの活動と成果について

MISRAは、Motor Industry Software Reliability Associationの略。MISRAの源流は1990年に英国政府がスタートさせた「システムの安全性及び信頼性に関する研究プログラム」にさかのぼる。「SafeIT」と名付けられたこのプログラムは20～30のプロジェクトで構成され、その1つが自動車業界の車両システムに安全性・信頼性のあるソフトウェアを提供することを支援するMISRAプロジェクト。同プロジェクトでは、自動車業界におけるエンジニアリングシステムの一環としてソフトウェア工学及び自動車工学の観点から安全性のあるシステムをソフトウェアとしてどう実現していくかという点について研究・調査した。

1994年にMISRAは研究・調査の成果を「The MISRA Guidelines」（「MISRAガイドライン」）としてまとめた。このガイドラインに基づいてIEC61508のドラフトが作成されている。MISRAプロジェクトが終了したあとも、プロジェクトにかかわった企業や大学などの組織がボランタリーなベースで、自動車関連における安全なシステムを開発していく上でグッドプラクティスを特定し、エンジニアがMISRAが作成したガイダンスにアクセスできるようにする取り組みを継続している。

自動車業界の技術者がC言語を用いて組込みシステムを迅速に開発できるようにすることを目的にMISRAが1998年に作成したドキュメントは「MISRA-C」と呼ばれる。このドキュメントが発行されて以来、世界50カ国で自動車業界をメインとしつつも、それ以外の業界で広く組込みシステムの開発に利用されている。今後、MISRAでは自動車業界におけるMISRA-C++の規約を作成することをテーマとして掲げている。その成果は自動車業界以外にも有用なものとなると期待される。

でエレクトロニクスやソフトウェアを学ぶ際にはプログラミング言語の教育がポピュラーですが、マルチメディア関連学科が人気を博しており、JavaやWebに学生の関心が向いています。いくつかの大学では、システム工学や安全性システムについて教えているのですが、それは修士レベルの課程であり、学部レベルのコースではありません。私たちは、コンタクトを取っている大学に対してシステム工学における安全性システムについての教育を行って欲しいという要請を出しているのですが、残念ながらまだそういうコースは普及していません。

鶴保　SECの活動の1つに、組込みソフトウェアスキル標準（ETSS）の作成があります。これは、ソフトウェア技術者が組込みシステムの開発に必要なスキルが身に付いているかどうかを確認できる体系になっています。このETSSをベースに大学での組込みシステム教育に適用しようと考えている先生が増えています。私自身も大学でソフトウェア工学を教えているのですが、ビジネスアプリケーションのスペシャリストを育てるのか、組込み系のスペシャリストを育てるのかによってカリキュラムを変える必要があると考えています。私が教えている大学では、初めからJavaを教えています。ビジネスアプリケーションやWebシステムを作成するには、Javaで十分です。しかし、Javaしか知らない卒業生が組込みシステムの業界に入ると非常にとまどうと思います。ですから、ビジネスアプリケーション系と組込み系の教育カリキュラムを分けざるを得ないのではないかと考えています。

Ward　全く同感です。プログラミング言語の教育ではC言語を学ぶことが最も多く、学生は例えば、最初に「Hello World」と表示するプログラムをC言語で作成します。しかし、C言語を使って信頼性の高い組込みシステムを作成する際に必要なC言語のさまざまな問題については十分に教えられていません。

鶴保　大学のソフトウェア教育では、プログラミング言語、アルゴリズム、マルチメディアなどのカリキュラムが整備されています。それに対して、ビジネスアプリケーション業界からは、「プロジェクトマネジメントをマスターしてきて欲しい」とか「ソフトウェアエンジニアリングをマスターしてきて欲しい」という要望があります。一方、組込み系の業界からは、

「組込みソフトウェアについてマスターしてきて欲しい」という要望がありますが、大学ではそういう先端的な分野の教育が行われていないのが現状です。産業界が求めるプロジェクトマネジメントや組込みソフトウェア技術が教えられていない背景には、産業界から大学へ人材が移動していないことがあげられます。産業界から大学へいって教えている先生方は、研究所出身者がほとんどで、彼らは産業界に通じているわけではありません。とくに、プロジェクトマネジメントや組込みソフトウェアは、最近のホットな分野であり、大学の先生が教えるのは至難の技です。そのため、産業界の現状に通じた人が直接学生に接する機会をつくって、産業界の問題を教えてもらうようにしないと産業界が求めるソフトウェア技術者育成の問題は解決しないと考えています。

Ward　鶴保所長のお話のように、学生自身が産業界の経験に触れて、産業界の課題について、よりよく知るということは重要だと思います。産業界で組込みシステムを開発する技術者を養成することに関連して、日本で行われたETロボットコンテストを見る機会があったのでその点に触れたいと思います。ETロボットコンテストに参加するエンジニアには、事前に課題となるコースが伝えられていました。そのコースの一部には、分岐があって、一方は近道だがでこぼこ道で走破が難しい設定になっていました。コンテストの参加者は、この悩ましい選択肢を把握して合理的な走行を考える必要がありました。そういう経験をすることは、信頼性があって満足できるシステムを構築するとは、どのようなことなのかを理解するのに役立つと思います。ロボットコンテストは、予測不可能な状況があるかもしれない、ということをエンジニアが考えながら開発しなければならない、ということを認識させるのによい機会だと思います。

組込みソフトにもプロセス改善の発想が必要

鶴保　エンタプライズの分野では、ソフトウェア開発のプロセス標準化を進めなければならないという考え方が時代とともに整理され、常識的になっています。さらに、プロセス改善することが重要です。教科書的なマニュアルだけでは解決できない問題を日常のプロセス改善によって解決する活動にエンタプライズグループが長年取り組んでいます。一方、組込みシステムは大規模になってからの歴史が浅いので、ソフトウェアを開発する上でのプロセスとプロセス改善の活動そのものが、それほど長い歴史をもっていないという認識のもとに、エンタプライズの分野で長い歴史を持っているプロセスとプロセス改善の活動を組込みに応用したいと考えて作業を進めています。組込みの場合でも、基本的にはエンタプライズと同じだと思いますが、組込みシステム特有の事情もあるのではないかと考えています。例えば、組込みシステムのリアルタイム性を考えた場合に、どのような設計をしたらよいのか、ソフトウェア開発のプロセスをどのようにするかということです。組込みシステムの場合、OSとミドルウェア、そしてリアルタイム性の強いアプリケーション、リアルタイム性の低いアプリケーションの3つに分かれます。したがって、組込みのシステムのプロセス改善を進める場合、その対象がOSなのか、リアルタイム性の強いアプリケーションなのか、それともリアルタイム性の少ないアプリケーションなのかということを考慮しながら取り組まなければならないという事情があります。

Ward　イニシアチブという点で、プロセス改善は非常に重要だと思います。特に、自動車業界における組込みシステムとして、そして安全性をもたらすシステムとしてプロセス改善を行うことは必要です。信頼性の高いシステムをそのようなスキームに基づいて提供するということが重要です。

SECとMISRAのアプローチは異なるがゴールは同じ

鶴保　我々の基本的なアプローチは、コーディングといえども技術的な問題だけでなく、プロジェクトそのものの問題が反映されるのではないかという考え方からスタートしています。したがって、最後のコーディングに関しては、さきほど申し上げたように、開発対象がリアルタイムなものか、そうでないのかによっても変わってきます。さらに、プロジェクトの分野や会社の考え方等によっても変わってくると思います。また、より安全性を求めるのか、より性能を求めるのかということによっても変わってくるという認識です。したがって、まず個別の事情を抽象化したものをまとめ、それから個別の部分にアプローチしていこうと考えました。

Ward　なぜ我々がプログラミング言語から組込みソフトウェアの問題解決を始めたのか。それには、2つの理由があります。1994年にソフトウェアエンジニアリングに関するガイドラインを発表するまで、自動車業界の中では、さまざまなプログラミング言語のサブセットが使われており、ガイドラインがありませんでした。そこで、ガイドラインを発表するにあたってC言語を選ぶことによって共通の基盤が提供できないかということで始めました。これが1つ目の理由です。2つ目の理由は、エンジニアが開発を進める上で使える実際のガイドを提供したいと考えたことです。例えば、組込みシステム用のC言語は、ISOの標準にいろいろな定義があるのですが、あいまいな部分や定義されていない部分が多いのです。例えば、「コンパイラに任せます」という表記もあります。そのため、我々としては、エンジニアが文献をよく研究しなくても済むよう、何らかの共通のフォーマットをエンジニアに提示し、エンジニアがそのフォーマットを入手できる環境ができるといいと考えました。それがMISRA-Cガイドラインを作成した2つ目の理由です。

鶴保　SECとMISRAの取り組みは、アプローチが違うと思いますが、目指すゴールは同じでしょう。

Ward　そうですね。現在、多くの企業でMISRA-Cをコーディングのスタンダードとして活用していただいています。なぜかというと、MISRA-Cを使うことのメリットが明快だからです。すなわちMISRA-Cで書かれたソフトウェアがきわめて高品質であるということ、MISRA-Cを活用することによって不具合率が低減されること、その結果として開発期間が短縮されること、さらには開発コストが抑えられるということが達成できるからです。日本の自動車業界においてもMISRA-Cが受け入れられ、活用されていることに満足していますし、感謝しています。MISRA-Cに関して、日本語の翻訳版もありますし、MISRA-Cの研究をしている団体もあり、感謝しています。また、一番大きな課題になるのが、日本人のエンジニアに対して日本語でサポートする、ガイダンスを行うということです。しかしその点に関しては、MISRA-Cを活用している人数が増えてきており、それをサポートしているネットワークも広がってきています。我々も、できる限りのサポートをさせていただければと考えています。

鶴保　これから社会システムの構築が増えると思います。中でも有力な社会システムは、自動車と交通の状況を把握したり、コントロールするシステムです。そうしたシステムは、組込みシステムとエンタプライズシステムが融合したものとなるはずです。そこでは、恐らく巨大なデータベースあるいは分散データベースが構築されるでしょう。そうしたことを考えると、我々もソフトウェアの品質を上げるためにコーディング基準が重要であるとの認識はまったく同じです。SECとMISRAのアプローチの仕方はやや違いますが、最終的なターゲットは近いと考えています。ぜひ我々とMISRAでコラボレーションできればと思います。

Ward　コーディング規約をどのように作成するかという点で、我々とSECのアプローチは違うのですがゴールは同じです。であるが故に、さまざまに協力し合ったり、お互いの経験を共有し合うことができるのでないかと思います。例えば、MISRA-Cの将来バージョンをリリースするにあたって、アプローチが違う故に我々が把握していない問題点を指摘していただくことなどができるのではないかと考えています。また、お互いに共通の利益、あるいは共通の関心事項として持っているのが安全関係のシステム構築だと思います。その点に関しても共通のアプローチを築き上げることができると思っています。

鶴保　我々も、コーディングの中にセキュアコーディングを取り入れようとSECとIPAセキュリティセンターとで協力して取り組んでいます。セキュリティの脆弱性は、Cで発生しやすいという認識のもとで取り組んでおり、そうした技術もお互いに交流できると思います。

SECでは、これからもMISRA-Cの取組みに注目していきます。

論文 Paper

第4世代のテスト・プロセス

山浦 恒央†

ソフトウェア開発で最新のテスト・プロセスが第3世代である．これは，開発部門とは独立のQA部門がテストをする方式で，高レベルの品質保証が可能となる．ソフトウェア開発は，世界的に短納期傾向にあり，高品質を維持しつつ，テスト工数，開発工数，納期を短縮する方法として，第4世代のテスト・プロセスを提案する．これは，①品質開発マネージャを軸にしたテスト・プロセスの導入，②開発側のデバッグと，QA側のテストの一体化による工期と工数の削減を骨子とするものである．

Fourth Generation Test Process

Tsuneo Yamaura)

The most advanced form of the test process in the software development is Third Generation Test Process: a QA team, which is organizationally independent from a development team, drives test activities to get rid of developer's biased mentality to attain high level quality assurance. Current trend in software development requires a shorter period with less cost. In order to satisfy such requirement, this paper proposes Fourth Generation Test Process, whose basics are (1) to assign a quality development manager, and (2) to merge developer's debugging and tester's testing.

1 はじめに

ソフトウェアを出荷する場合，日本では，品質の未成熟なソフトウェアをリリースすることで発生する製品回収やバグ対策による経費や人月を最少にするため，出荷時期を遅らせても，品質を確保してきた．2000年以降の世界的な状況を見ると，ソフトウェア開発で，以下の2つの傾向が非常に顕著である[YAMAURA1999].

①重厚長大から，小型・短納期へ移行.

②組み込み系ソフトウェア等での，出荷時期遅延によるビジネス・チャンスの喪失.

ソフトウェアの出荷時期厳守は必須であり，開発期間短縮は，最優先の命題になりつつある.

一方，ソフトウェアの開発において，開発する機能の総量，開発時間，投入工数，生産性の間には，式　の関係があることが報告されている[PUTNAM2004].

$$\frac{(\text{工数})^{1/3}\cdot(\text{開発時間})^{4/3}}{\text{生産性}} = \beta \times (\text{機能の総量})$$

β：定数

この関係式から，開発時間と工数の間には，トレード・オフがあり，高レベルの品質を維持し，当初の機能を支援し，かつ，開発期間を短縮するには，生産性を急激に上げるか，投入人員を増やすしかない．生産性を急上昇させるのは非現実的なので，投入人員を増やさざるを得ないが，ソフトウェア開発には，人員を無限に増やしても，ある期間より短い時間で開発できない最短開発時間が存在する．予定開発時間が最短開発時間より十分に長い場合，増員すると，それに反比例して開発期間は短くなるが，最短開発時間に近づくと，投入工数に比べ，短縮できる時間が激減する.

以上より，日本型の出荷方式の高い品質レベルを維持しながら，開発時間を短縮するには，開発プロセスを根本的に変える必要がある．とくに，ライフサイクルの半分を占める品質確保やテストのプロセスを改善することは大きな効果がある.

†) 法政大学情報科学部, Information Science Dept., Hosei University

2 テスト・プロセスの進化

ソフトウェアのテスト・プロセスは，他のソフトウェア開発プロセスと同様の進化経路をたどる．開発技術の進化と異なる点は，開発技術の進化は，ソフトウェア産業全体で進むのに対し，テスト・プロセスの進化は，組織により偏差があることである[GLASS2004]．高度なテスト・プロセスが成功している組織がある一方で，デバッグと等価のテストを実施するだけの組織も多い．通常，ソフトウェアのテストは，以下の進化過程をたどる.

①第1世代：開発者自身がテストを実施する.

②第2世代：開発部門の中に，テスト・チームあり.

③第3世代：開発部門と独立に，テスト・チームあり.

3 テスト・プロセスの概要と評価

3.1 第1世代のテスト・プロセスの概要と評価

第1世代のテスト・プロセスは，開発者自身がテストを実施するため，構造に即したテストを実施できるが，テストではなく，デバッグなので，以下の欠点がある.

①単なるデバッグの延長であり，ソフトウェアの不良に対する心理的な姿勢が異なる.

②開発者はデバッグやテストの能力，経験がない.

③自分で自分のプログラムを検証するため，心理的に，「正常に動く」との潜在意識が働き，厳しいテスト項目を無意識に避ける傾向になる

④テスト自体の品質測定を実施していないため，テストでどの程度の品質が確保できたか，不明である.

3.2 第2世代のテスト・プロセスの概要と評価

第2世代のテスト・プロセスは，同じ開発プロジェクトの技術者で構成されたチームがテストを担当するため，ソフトウェアの内部構造に着目したテストを系統的に実施できる利点がある．一方，開発とテストを同じプロジェクトの技術者が実施するため，以下の問題がある.

①テスト担当者は，本来は開発者なので，テストの能力，技法，ノウハウを十分に蓄積できない.

②開発部門内の開発者の意見に影響され，結局は，「ソフトウェアは，正常動作する」という性善説に従ってテストを実施するため，不良を見逃しやすい.

③開発部門内の組織であるため，品質に問題があっても，開発側の意見に押し切られる．とくに，出荷時期と品質が問題になった場合，開発側や顧客のビジネス戦略が優先され，出荷品質に満たない製品を顧客へリリースする可能性が高い.

④プログラマには，「テストはプログラム開発や設計の必要悪」との考えがあり，プロジェクトでテストを担当するエンジニアの士気が上がらない.

3.3 第3世代のテスト・プロセスの概要と評価

ソフトウェア開発におけるテスト・プロセスで，最も進化した形態が第3世代である（図1）．本方式には，第1世代，第2に比較して以下の利点がある.

①設計者との「親密な関係」がなくなり，性悪説に従って，テストの目的である「ソフトウェアが正しく動作しないことを証明すること」を進められる.

②テスト担当者は，テストを専門に実施するエンジニアなので，デバッグやテストの能力，技法，ノウハウ，経験を十分に備えている.

③出荷時期が迫っているのに品質が不十分な場合，早急に出荷したい開発側と，対等の立場で，品質・出荷時期の問題を議論できる.

一方，第3世代のテスト・プロセスにも，以下の問題や課題がある.

(1) 設計者が設計・実施したテスト項目の再利用

デバッグの目的は，「プログラムが正常動作することを証明する」であり，テストは，「プログラムが正しくないことを証明する」である．デバッグとテストでは，目的やメンタリティが全く異なるが，実作業での共通部分は非常に多い．デバッグとテストの違いと類似性を的確に理解していれば，両者の共通部分を有効利用することが可能となる.

(2) 開発部門における品質管理のテスト部門への依存

品質管理で，開発部門とテスト部門が独立しており，

技術的な交流がないことと，開発部門の品質管理能力がテスト部門に比べて低いため，開発部門は，品質管理をテスト部門に依存する傾向があり，品質管理技術，能力を向上させる姿勢が見られなくなる．

(3) デバッグとテストでの類似作業

品質保証に要するコストと期間は，要求される品質のレベルにより大きく異なる．本論文での議論や効果算出のため，以下の標準的なモデルを取り上げる．

仮定：10人の開発担当者が，12ヶ月間にC言語で100KLOCのソフトウェアを新規開発する．

(a) 各開発フェーズに要する月数

①仕様分析・仕様決定：2ヶ月×3人
②設計：2ヶ月×10人=20人月
③コーディング：2ヶ月×10人=20人月
④デバッグ：3ヶ月×10人=30人月
⑤テスト（QA技術者）：3ヶ月×3人=9人月
⑥不良修正，品質向上：3ヶ月×5人=15人月

(b) 3ヶ月のデバッグの詳細（10人の開発者が対応）

①100,000行のソフトウェアに10,000件のテスト項目が必要（1,000件/プログラマ）．
②10,000件のテスト項目を設計するのに10日（1日1人あたり100件）
③1,000件の机上デバッグに10日（1日1人あたり10件）
④10,000件のマシン・デバッグに40日（1日1人あたり25件）

(c) 3ヶ月のテスト・フェーズの詳細

・テスト側（3人のQAエンジニアが担当）
①5,000件のテスト項目を設計するのに10日（1日1人あたり，167件）
②5,000件のマシン・デバッグに50日（1日1人あたり33件実施）
・開発側
①テスト・チームが指摘したバグの修正，類似バグの見直し等　3ヶ月×5人=15人月

上記で，デバッグに3ヶ月費やし，その後で，デバッグと類似作業であるテストに同じく3ヶ月の期間を割り当てている．これを共通化できれば，12ヶ月の開発期間を1〜3ヶ月短縮できる可能性がある．

(4) テスト担当者のメンタリティ

テストは高度に創造的である．対象プログラムの脆弱性を系統的に摘出するには，設計したプログラマよりも高い能力が必要であり，システムを容易に開発できる能力，創造力がないとテストはできない．ソフトウェア業界における現実的な認識はこれと逆である．テスト担当者だけでなく，設計者も，「テストは，創造的な仕事ではない」，「テスト担当者は，システム開発能力がなく，プログラマより能力が低い」と考える傾向にあり，士気が下がる．この対策として，「テストは，品質を開発するという前向きの作業」との意識付けが必要となる．

(5) テスト部門とノウハウの共有

開発担当者は，構造的で，変更にも柔軟に対応できるプログラムを効率よく作成する能力はあるが，ソフトウェアの検証技術はテスト技術者より劣る．開発担当者が検証技術やテスト技法を体系的に学習することは稀で，自分で編み出したり，同じプロジェクトの上級プログラマから，徒弟方式で学ぶ場合が多い．上級プログラマも，自己流でデバッグをしてきたので，効率がよく，バグの検出効果が高いテスト手法を習得していない．

第3世代のテスト・プロセスの骨子は，開発チームとテスト・チームを組織的に独立させ，両チームの「癒着」を廃することで，バグの摘出率を上げることにあるが，組織の「断絶」により，テスト担当者が蓄積してきた品質向上・バグの未然防止のノウハウも，設計側には，十分には伝わらない結果となった．

4　第4世代の品質保証プロセス

4.1　第4世代の品質保証プロセスの概要

第1世代，第2世代，第3世代のテスト・プロセスの欠点を是正し，より効率的な品質保証を実施するため，「第4世代の品質保証」を提案する．第4世代のテスト・プロセスとは，「開発側が実施するデバッグと，QA側が実施するテストを統合し，品質開発マネージャがそれを指揮する」というものである．第4世代のテスト・プロセスの概要は以下の通りである（図2）．

①ソフトウェア開発プロジェクトに，QA部門から派遣された品質開発マネージャを1人設置する．
②最終製品以外の中間成果物（要求仕様書，設計書，ソースコード等）の場合，開発者がそのデバッグ，テストを設計，実施する．その場合，品質開発マネージャが，開発者に対し，デバッグの内容，テスト方式，テスト結果の評価方法を指揮する．
③最終製品の場合，第3世代のテスト・プロセスでは，開発側とは独立に，テスト・チームが最終製品のテストを実施したが，第4世代のテスト・プロセスでは，開発側の技術者がデバッグとテストを実施し，品質開発マネージャがそれを指揮する．最終製品の品質保証

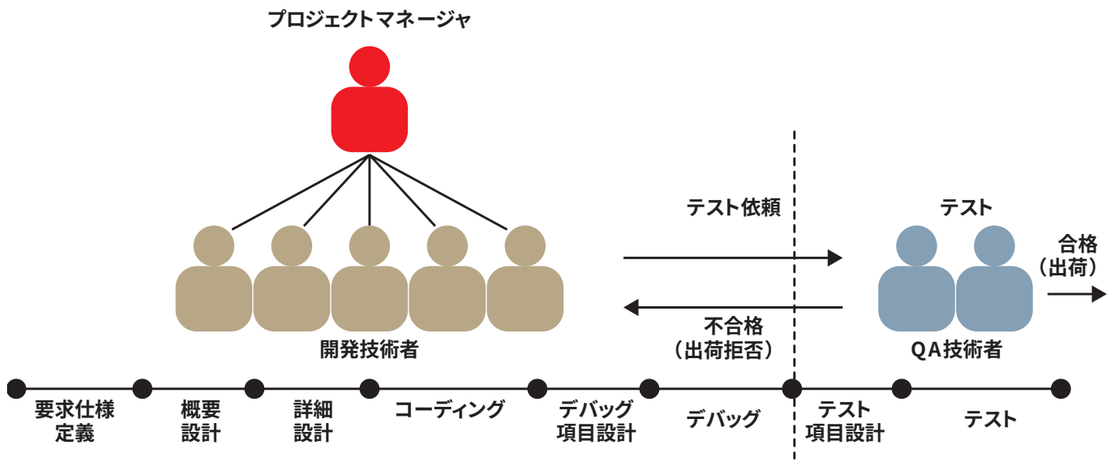


図1　第3世代のテスト・プロセス

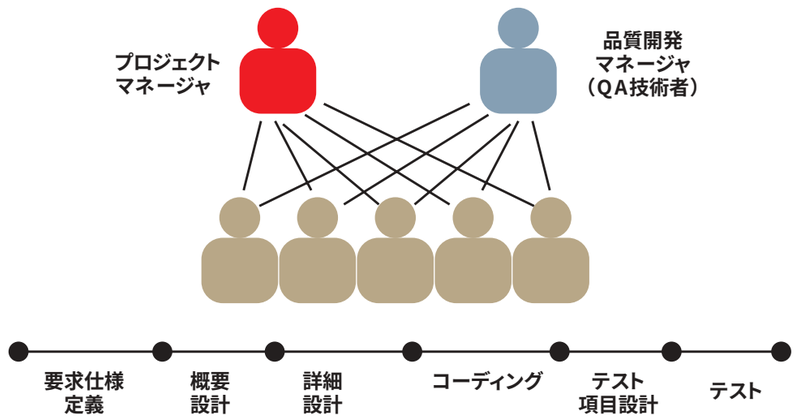


図2　第４世代のテスト方式

プロセスは以下の通り．

・開発者が作成したデバッグ項目の内容をテストの専門家としての視点で精査し，デバッグ項目の追加，変更，削除を指示する．
・開発者のデバッグ項目消化状況，バグ発生状況から品質を予測し，デバッグ項目の追加，変更，削除を指示する．
・デバッグが終了した時点で，品質開発マネージャは，最終製品の品質を評価する．これが，事前に設定した終了基準に合致している場合，意図した品質が確保できたとみなし，従来のテストを実施することなく，出荷する．

5　第4世代テスト・プロセスの期待効果

開発側のデバッグとQA側のテストをまとめ，品質開発マネージャが管理すると，次の効果を期待できる．

テスト期間の短縮による早期出荷の実現
テスト項目の設計・実行における資源の削減
開発担当者のテスト能力向上
テスト要員の削減
テスト担当者の士気向上

各項目について，以下，詳細を検討する．

5.1　早期出荷の実現

第3世代テスト・プロセスのソフトウェア開発の例として，開発作業が9ヶ月，QA作業が3ヶ月というプロジェクトを取り上げた．12ヶ月のうち，開発側のデバッグに3ヶ月，QA側のテストに3ヶ月かけている．6ヶ月の品質保証期間を統合し，同じ品質を確保できれば，理論上は開発期間を25%削減できる．実際には，品質開発マネージャがテスト内容のチェック，レビュー，追加・修正をするため，4ヶ月かかると考えるのが妥当であり，2ヶ月短縮でき，開発期間を17%削減可能となる．

5.2　テストにおける資源の削減

第3世代テスト・プロセスでは，開発側のデバッグ項目をQA側は参照せず，独立にテストを実施していた．デバッグ用の項目をQA側が再利用することで，時間，テスト資源，人員の重複を半分近くまで削減可能となる．

5.3　開発エンジニアのテスト能力向上

効率よくデバッグやテストを実施するには，品質管理，テスト技法の知識，経験が必須である．現状は，徒弟制度や，試行錯誤により，テスト能力を身に付けている．テストの専門家である品質開発マネージャが，プロジェクトの品質管理を指揮・管理し，開発担当者が作成したテスト計画やテスト項目をチェック，指導することで，開発担当者は，系統的にテスト技術を習得できる．

5.4　テスト要員の削減

第4世代のテスト方式では，QA部門に所属する品質開発マネージャがデバッグ，テスト戦略を決め，開発担当者は，品質開発マネージャの指揮下，デバッグやテストを実施する．これにより，テストを実施するQA担当者が不要となり，人員削減が可能となる．QA関係の技術者数は，全技術者の5%から10%といわれており[YAMAURA 1998]，このQA担当者を半減させることが可能となる．

5.5　QA担当者の士気向上

第3世代では，QA担当者は，開発担当者の作ったプログラムをテストするという「後戻り的な作業」というイメージが強く，QA担当者は，高い技術を持ちながら，モチベーションは低かった．第4世代のテスト方式で品質開発マネージャを導入することにより，①品質保証は，品質を開発するという前向きで創造的な作業となる，②品質に関し，全責任を持つ，③製品出荷時期の決定では，プロジェクト・マネージャと対等の立場で議論できる．第4世代のテスト方式により，QA担当者の士気が，飛躍的に向上すると期待できる．

6　適用プロジェクトの概要

第4世代テスト方式の効果を検証するため、実際のプロジェクトで適用した．5プロジェクトでは，従来の第3世代のテスト方式を，8プロジェクトで，第4世代のテス

表1　プロジェクトのデータ（各開発フェーズの期間と工数）

プロジェクト名	期間・工数	要求仕様定義	概要設計	詳細設計	コーティング	デバック項目設計	デバック	テスト項目設計	テスト	合計
プロジェクトA（適用前その1）	期間（週）	8	5	4	7	3	7	2	4	40
	工数（人週）	41	63	69	101	44	101	15	22	456
プロジェクトB（適用前その2）	期間（週）	1	1	2	2	2	2	1	2	13
	工数（人週）	1	2	2.5	5	2.5	5	1	1.5	20.5
プロジェクトC（適用前その3）	期間（週）	1	1	2	2	2	2	1	2	13
	工数（人週）	1	3	3.5	6	3.5	6	1	1.5	25.5
プロジェクトD（適用前その4）	期間（週）	1.5	1.5	2	3	1.5	2	2	3	16.5
	工数（人週）	3	3	5	8	4	5	2.5	2.5	33
プロジェクトE（適用前その5）	期間（週）	1.5	1.5	2	3	1.5	2	2	3	16.5
	工数（人週）	2	2	3	6	2	4	1.5	1.5	22
プロジェクトF（適用後その1）	期間（週）	2	2	3	3	2	3	—	—	15
	工数（人週）	6	9	10	17	8	13	—	—	63
プロジェクトG（適用後その2）	期間（週）	3	2	3	3	3	4	—	—	18
	工数（人週）	5	7	9	13	6	12	—	—	52
プロジェクトH（適用後その3）	期間（週）	2	2	2	2	2	3	—	—	13
	工数（人週）	4	5.5	7	10.5	5	9	—	—	41
プロジェクトI（適用後その4）	期間（週）	3	3	4	3	2	4	—	—	19
	工数（人週）	6	7	11	15	4	14	—	—	57
プロジェクトJ（適用後その5）	期間（週）	1	1	2	2	1	3	—	—	10
	工数（人週）	2	2.5	5	5	1.5	10	—	—	26
プロジェクトK（適用後その6）	期間（週）	1	1	2	2	1	3	—	—	10
	工数（人週）	3	2.5	5	5	2	6	—	—	23.5
プロジェクトL（適用後その7）	期間（週）	3	3	2	2	3	4	—	—	17
	工数（人週）	5	5	5.5	6	5	8	—	—	34.5
プロジェクトM（適用後その8）	期間（週）	1.5	1.5	2	2	1	3	—	—	11
	工数（人週）	3	3	6	5	2	5.5	—	—	24.5

表2　プロジェクトのデータ（デバッグ、テスト・フェーズでのテスト件数と摘出バグ数）

プロジェクト名	テスト件数 摘出バグ数	机上デバッグ（テスト）	単体デバッグ（テスト）	組合わせデバッグ（テスト）	システムデバッグ（テスト）	テスト	合　計
プロジェクトA（適用前その1）	実施テスト件数	870	4902	3488	1026	4023	14309
	検出バグ	172	339	211	54	125	901
プロジェクトB（適用前その2）	実施テスト件数	53	199	161	21	135	569
	検出バグ	7	10	9	3	11	40
プロジェクトC（適用前その3）	実施テスト件数	63	237	213	29	174	716
	検出バグ	6	16	15	5	14	56
プロジェクトD（適用前その4）	実施テスト件数	88	239	183	43	243	796
	検出バグ	7	24	11	4	7	53
プロジェクトE（適用前その5）	実施テスト件数	70	202	164	28	213	677
	検出バグ	5	18	8	3	4	38
プロジェクトF（適用後その1）	実施テスト件数	394	580	243	210	—	1427
	検出バグ	32	55	35	9	—	131
プロジェクトG（適用後その2）	実施テスト件数	254	516	201	142	—	1113
	検出バグ	26	33	31	18	—	108
プロジェクトH（適用後その3）	実施テスト件数	311	419	103	83	—	916
	検出バグ	21	21	19	9	—	70
プロジェクトI（適用後その4）	実施テスト件数	195	571	188	107	—	1061
	検出バグ	19	40	33	17	—	109
プロジェクトJ（適用後その5）	実施テスト件数	164	306	121	82	—	673
	検出バグ	16	18	11	7	—	52
プロジェクトK（適用後その6）	実施テスト件数	171	203	122	92	—	588
	検出バグ	23	17	9	5	—	54
プロジェクトL（適用後その7）	実施テスト件数	229	301	215	120	—	865
	検出バグ	46	15	8	2	—	71
プロジェクトM（適用後その8）	実施テスト件数	161	225	115	93	—	594
	検出バグ	28	11	9	5	—	53

ト方式を採用した．各プロジェクトの概要を以下に示す．

全13のプロジェクトは，通信サービス管理をWebで実施するクライアント・サーバをベースにしたソフトウェアであり，同一プログラムを開発・改造した．開発手法は，ウォーターフォール・モデルで，使用言語は，Java，C++である．以下に，各プロジェクトの概要を示す．また，表1，表2に各プロジェクトのデータを示す．

・プロジェクトA
　新規開発（第3世代テスト・プロセス適用）
・プロジェクトB～E
　機能追加（第3世代テスト・プロセス適用）
・プロジェクトF～M
　機能追加（第4世代テスト・プロセス適用）

7　第4世代プロセスの効果の検証

5プロジェクトで従来の第3世代のテスト方式を適用し，8プロジェクトで第4世代のテスト方式を採用し、各プロジェクトの結果を分析して，第4世代テスト方式の効果を検討した．

表3に，13のプロジェクトのテスト関連データ（プログラム開発規模，全工数，品質確保工数，総デバッグ・テスト項目数，摘出バグ数，プログラム開発効率，テスト密度，バグ密度，バグ摘出打率）を示す．これをもとに，第4世代テスト方式の効果を考察する．13のプロジェクトとも，同一アプリケーションのソフトウェア開発であるため，プログラム開発上の難易度に大きな違いはない．しかし，プロジェクトAは，プロジェクトB～Mに比較し，(1) 開発規模が7～22倍違う，(2) プロジェクトAは100％新規作成だが，プロジェクトB～Mは，変更であり，開発形態が異なる，(3) 開発担当者は，プロジェクトB～Mまでは同じであるが，プロジェクトAは異なる，の理由により，プロジェクトB～Mのデータをもとに，第4世代テスト方式の効果を検証する．

7.1　品質レベルの比較

表3から，プロジェクトB～Mのプログラム開発規模あたりの摘出バグ数を図3に示す．第3世代テスト方式の1KLOCあたりの不良の数は，平均8.9個／KLOCだが，第4世代テスト方式は，8.1個／KLOCであり，品質レベルは向上している．これは，品質開発マネージャがプログラム開発の初期段階から，品質管理の観点で，プログラム開発者の品質管理プロセスに介入した効果である．最後のプロジェクトが稼働状態に入り、1年以上が経過したが、プロジェクトAに関するユーザ指摘は数件あったが、BからMまでのバージョンは、不良、障害等の報告を受けていない。プログラム開発者との事後インタビューにおいても，品質レベル比較を確認した．

表3　プロジェクトのデータ (生産性、バグ摘出効率、テストの有効性など)

プロジェクト名	開発規模（KLOC）	全工数（人数）	品質確保工数（デバッグ＋テスト）	デバッグ＋テスト項目数	摘出バグ総数	開発効率（工数／KLOC）	品質確保工数率（品質確保工数／全工数）	テスト密度（テスト項目／KLOC）	摘出バグ密度（摘出バグ総数／KLOC）	バグ摘出打率（摘出バグ総数／テスト項目数）%
プロジェクトA（適用前その1）	103.3	456	182	14309	901	4.41	0.40	138.5	8.72	6.30
プロジェクトB（適用前その2）	4.6	20.5	10	569	40	4.46	0.49	123.7	8.70	7.03
プロジェクトC（適用前その3）	5.6	25.5	12	716	56	4.55	0.47	127.9	10.00	7.82
プロジェクトD（適用前その4）	5.9	33	14	796	53	5.59	0.42	134.9	8.98	6.66
プロジェクトE（適用前その5）	5.0	22	9	677	38	4.40	0.41	135.4	7.60	5.61
プロジェクトF（適用後その1）	15.1	63	21	1427	131	4.17	0.33	94.5	8.68	9.18
プロジェクトG（適用後その2）	12.0	52	18	1113	108	4.33	0.35	92.8	9.00	9.70
プロジェクトH（適用後その3）	10.1	41	14	916	70	4.06	0.34	90.7	6.93	7.64
プロジェクトI（適用後その4）	15.1	57	18	1061	109	3.77	0.32	70.3	7.22	10.27
プロジェクトJ（適用後その5）	6.6	26	11.5	673	52	3.94	0.44	102.0	7.88	7.73
プロジェクトK（適用後その6）	5.8	23.5	8	588	54	4.05	0.34	101.4	9.31	9.18
プロジェクトL（適用後その7）	8.4	34.5	13	865	71	4.11	0.38	103.0	8.45	8.21
プロジェクトM（適用後その8）	6.2	24.5	7.5	594	53	3.95	0.31	95.8	8.55	8.92

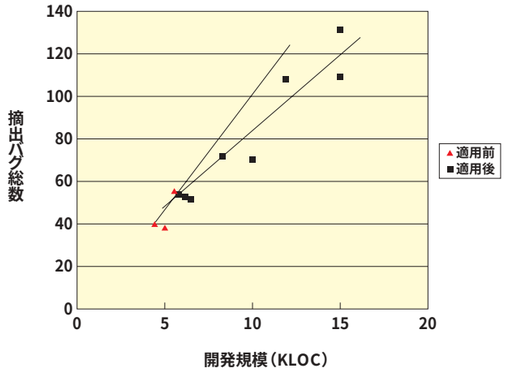


図3　品質レベルの比較

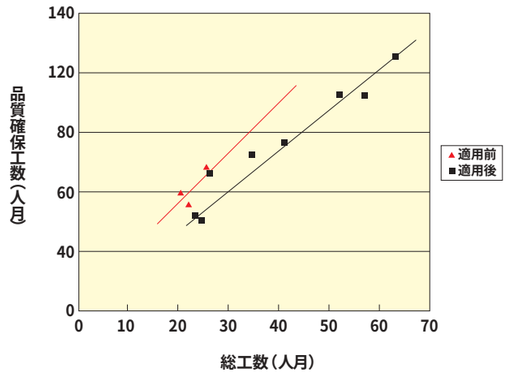


図4　品質確保工数率の比較

7.2　テスト期間の短縮による早期出荷の実現

表3に示したように，第3世代のプロジェクトB～Eでは，品質確保工数率（品質確保工数／全工数），すなわち，全工数中のデバッグ，テストの割合は，各49％，47％，42％，41％であり，第4世代のプロジェクトF～Mでは，プロジェクトB～Eより高い品質レベルを保ちながら，品質確保工数率が33％，35％，34％，32％，44％，34％，38％，31％に低下している（図4）．

第3世代テスト方式の品質確保工数率が45％と仮定すると，プロジェクトF～Mに第3世代方式を適用した場合，21人週，18人週，14人週，18人週，11.5人週，8人週，13人週，7.5人週が，それぞれ，28.5人週，23.5人週，18.5人週，25.5人週，11.5人週，10.5人週，15.5人週，11人週に増加するものと予測できる．これにより，プロジェクトF～Mは，第4世代のテスト方式により，総工数が従来の74％，77％，76％，70％，100％，76％，84％，68％に減少できたと考察できる．

以上により，第4世代のテスト方式は，テスト・リソースの共有により、テスト期間短縮が可能となり、早期出荷（総工数削減）に効果が認められると考えられる．

7.3　テスト項目の設計・製造・実行での資源削減

表3の「バグ摘出率（摘出バグ総数／テスト項目数）」は，テスト項目1件で検出するバグ数をパーセントで示す（図5）．第3世代のテスト方式（プロジェクトB～E）では，テスト項目1件でバグを摘出できる確率は，5.6％～7.8％（平均6.8％）であるが，第4世代のテスト方式（プロジェクトF～M）では，7.7％～10.2％（平均9.0％）まで上昇している．これは，テスト項目の効率がよいことを示し，品質開発エンジニアの指揮によるデバッグ，テストが効果を出していることを示す．

図6にプログラム・サイズとテスト項目数の比較（テスト項目密度）を示す．第3世代テスト・プロセスのプロジェクトB～Eでは，ソースコード1KLOCあたり，テスト項目をそれぞれ，123.7件，127.9件，134.9件，135.4件実施している（平均130.5件）．一方，第4世代のテスト・プロセスのプロジェクトF～Mでは，それぞれ，94.5件，92.8件，90.7件，70.3件，102.0件，101.4件，103.0件，95.8 件（平均93.8件）と大幅に低下した．品質レベルで，プロジェクトB～Eより，プロジェクトF～Mの方が高いため，プロジェクトF～Mで品質を確保するため実施したテスト項目は，品質確保の効果が高く，デバッグとテストを一体化したこと，品質開発エンジニアのノウハウが設計者に効率的に移転されていることによる資源削減

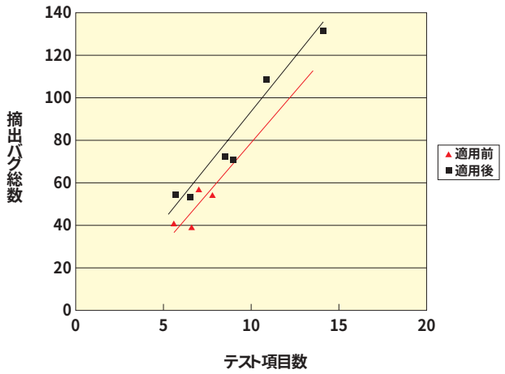


図5　テスト項目のバグ摘出率の比較

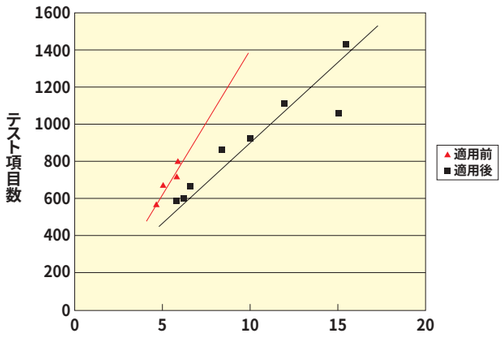


図6　プログラム・サイズとテスト項目数の比較

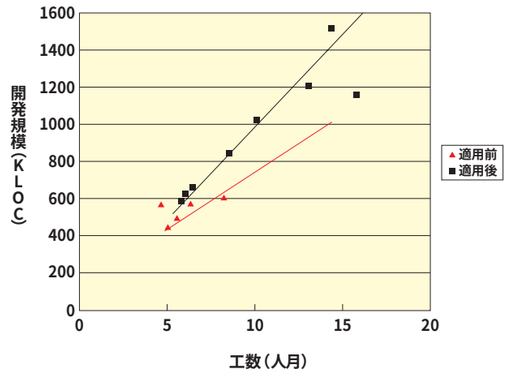


図7　生産性の比較

の効果が出ていると考察できる．

7.4　生産性の比較

KLOCあたりに実施するテスト項目数が減少し，デバッグ・テスト効率が向上したことで，生産性も向上した．図7で，第3世代のテスト・プロセスのプロジェクトと，第4世代テスト・プロセスのプロジェクトの生産性を比較した．第3世代では，1人週あたりの開発プログラム行数は，平均0.21KLOC，第4世代では，1人週あたりの開発プログラム行数は，平均0.25KLOCであり，テスト項目数の減少により、約16%の生産性改善が見られる．

7.5　開発エンジニアのテスト能力向上

この効果を直接測定するのは難しいため，品質開発マネージャと，開発エンジニアにインタビューを実施した．その結果，以下の効果があることが判明した．

(1) 品質開発マネージャ

・開発エンジニアが設計・作成したテスト項目は，「同値分割」，「境界値分析」等，既に確立されたテスト技法と大きな違いはない．しかし，多分に経験的であり，系統的でないため，漏れがあった．
・開発エンジニアが設計・作成したテスト項目の内容をチェックし，既に確立された技法を中心に各技法の考え方，目的や効果を説明し，システマチックにバグを摘出するための考え方を説明した．
・開発担当者が設計したテスト項目は，自分がよく理解している機能は細かく設計し，理解していない箇所はテスト項目の密度が粗い．また，シンタックス・チェック等は，機械的にテストを作成できるので作りすぎ，自分がテストを実施する意識が薄い．
・機能的なテストに関しては，大きな漏れはないが，性能やメモリの多寡に関するテストが不足している．性能は後でも上げられると思っているようである．

(2) 開発エンジニア

・テスト項目が不足していると感じた箇所は，ほとんど，指摘を受けた．テスト効率の重要性を再認識した．
・テスト技法によって，摘出が得意なバグと不得意なものがあることが理解できた．
・テスト項目の数が多すぎても，少なすぎても良くないことが理解できた．特に，多すぎる場合は，少ない場合と同様，弊害があることを理解できた．

7.6　テスト要員の削減

第3世代のテスト方式を踏襲したプロジェクトA～Cでは，QAエンジニアが，それぞれ，4人，2人，2人，テスト時に参画した．第4世代のテスト方式を導入したプロジェクトD，Eでは，QA部門から投入した人員は，品質開発マネージャ以外はゼロである．プロジェクトD，Eに，第3世代のテスト方式を適用した場合，QA部門のエンジニアが，2人ずつ必要になると思われる．品質開発マネージャの1人分を差し引くと，D，Eそれぞれのプロジェクトで，1人のQAエンジニアが削減できたことになる．品質開発マネージャによるテスト方式を広く適用した場合，QAの人員を1/3まで削減できると期待できる．

7.7　QA担当者の士気向上

この効果も，数字には表れないので，品質開発マネージャにインタビューを実施した．結果は以下の通り．

・開発側のエンジニアを指揮して，品質を確保するのは，初めての経験だが，品質管理や信頼性の本来の仕事はこうあるべきとの印象を持った．
・自分の考え方や戦略が間違っていると，品質に直接，跳ね返るため，大きな責任を感じる．品質開発マネージャのための作業の標準化，教育セミナー等を充実させる必要があると思う．
・開発エンジニアは，意外に，テスト技法を知らない．
・開発側のテスト項目は，正しく動作するとの思いが入っている．

8　課　題

第4世代のテスト・プロセスは，期待した効果を上げることができた．ただし，この効果は，品質開発マネージャに大きく依存し，以下の能力，技術が必要となる．

・開発，品質管理で5～10年の経験と技術力
・システムの正当性，信頼性，機密性，安全性，可用性等を検証，分析できる能力
・各テスト技法の利点，欠点，限界の知識と経験
・対象プログラムに最適のQA戦略企画力

この能力を備えた品質開発マネージャは，非常に少なく，養成のため，能力の設定，教育コースの構築，品質開発マネージャの作業プロセス定義が必要となる．

9　結　言

「開発部門がデバッグし，独立したQA部門がテストする」という第3世代テスト・プロセスを改善するため，第4世代テスト・プロセスを提案した．第4世代テスト方式は，①QA部門の品質開発マネージャが品質の責任を持つ，②従来，独立に実施したデバッグとテストを1つにまとめる，③デバッグ，テストの実作業は開発部門のエンジニアが担当する，④品質開発マネージャは，テストの戦略を立て，開発部門のエンジニアを指揮して，デバッグやテストを実施する．

第3世代のプロジェクトB～Eの総工数と開発期間（4.75人週/KLOC、2.81週/KLOC）に比べ、第4世代方式のプロジェクト（4.05人週/KLOC、1.51週/KLOC）は、工数と期間共に削減（工数：15%削減、期間:46%削減）できた．1KLOCあたりの実施テスト項目数は，130.5件／KLOCから93.8件／KLOCへ減少（28％減少），また，テスト項目1件当りでバグを摘出できる確率は，0.068から0.09へ上昇した（30％の向上）．

上記以外の効果として，開発側エンジニアが設計したテスト項目を，品質管理のプロである品質開発マネージャがチェックするため，テスト技法が技術移転される．また，QAエンジニアは，士気が向上するという効果もあった．テストは後戻り的なイメージがあったが，品質開発マネージャは品質を開発するという前向きの姿勢があり，創造性と相まって士気が高揚した．

参考文献

[GLASS2004] Glass, R. : Facts and Fallacies in Software Development, Addison Wesley, 2004
[PUTNAM2004] Putnam, L. : Mayers, W., Five Core Metrics, Dorset House, 2004
[YAMAURA1998] Yamaura,T. : How to Design Practical Test Cases,IEEE Software,Vol.15,Ser.6, 1998, pp.30-36
[YAMAURA1999] Yamaura,T. : Three Wishes for Software Quality in the 21st Century, American Programmer, Cutter Consortium, Feb., 1999

論文 Paper

企業横断的収集データに基づくソフトウェア開発プロジェクトの工数見積り


大杉 直樹†


角田 雅照†


門田 暁人†


松村 知子†


松本 健一†


菊地 奈穂美††

プロジェクトの工数見積のための基礎データとして，複数の異なる組織で収集されたソフトウェア開発プロジェクトのデータを利用できれば，蓄積データの少ない組織においても精度の高い見積りができると期待される．

本論文では，1つのケーススタディとして，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・センターが収集した多数のソフトウェア開発プロジェクトのデータを用いて工数見積を行い，その精度を評価した．見積手法として，重回帰分析，対数重回帰分析，ニューラルネット，協調フィルタリングを用いた．評価の結果，協調フィルタリングは他の手法よりも高い精度を示し，実績工数に対する見積値の相対誤差の平均値が0.642，Pred25（相対誤差が0.25以下のプロジェクトの割合）が30.1%であった．

Using Cross-company Data to Estimate the Effort of Software Development Projects

Naoki Ohsugi †, Masateru Tsunoda †, Akito Monden †, Tomoko Matsumura †, Ken-ichi Matsumoto †, and Nahomi Kikuchi ††

To predict software development effort, it would be useful to employ cross-company data collected outside a company, especially for the company who owns few project data. In this paper, as a case study, we conducted an effort prediction experiment using project data collected at the Software Engineering Center, Information-Technology Promotion Agency, Japan. In the experiment we used linear regression, log-linear regression, neural network and collaborative filtering (CF) as prediction methods. As a result, the CF showed the best prediction performance, and its average MRE (Mean Relative Error) was 0.642 and the average Pred25 was 30.1%.

1　はじめに

ソフトウェアプロジェクト管理において，スケジュール管理及び要員配置を適切に行うためには，工数の見積もを高い精度で行うことが重要である[BOEHM1984]．そのために，数多くの工数見積手法が提案されている[CONTE1986]，[OHSUGI2004]，[SHEPPERD1997]，[TAKADA1994]，[WALSTON1977]．現在進行中のプロジェクト（現行プロジェクト）の完遂に要する工数を正確に見積もることができれば，プロジェクト管理者は，人的資源を効率的に配置し，納期の遅れ，コストの超過といったプロジェクトの失敗を未然に防ぐことが可能となる．一般に，現行プロジェクトの工数を見積もる際には，過去に実施されたプロジェクト（過去プロジェクト）のデータを見積りの根拠とする[CONTE1986]．プロジェクトの特性を表す変数（特性変数）として，プロジェクトの完遂に要した工数や日数，要員のスキルや人数，開発に用いた技術や手法，開発された成果物の規模（ファンクションポイントやコード行数等），発見されたバグの数等を，プロジェクトごとに記録し，蓄積する．蓄積した過去プロジェクトのデータから見積モデルを作成し，現行プロジェクトから計測したデータをモデルに当てはめることで見積りを行う．

精度の高い（正確な）見積結果を得るためには，大量のプロジェクトから，同種の特性変数をできるだけ数多く，かつ一貫した計測方法により収集する必要がある．プロジェクトごとに収集されている特性変数の種類や，計測方法が異なる場合，プロジェクト間，並びに特性変数間の因果関係を正確に分析することができないため，見積精度が低下する．そのため，データ収集を始める前に，収集する特性変数，収集の時期（開発のどの段階でデータを収集するのか），収集に用いるツール，収集の手順等について，社内で統一的な取り決めを行い，その取り決めに従ってデータを収集することが重要となる．

しかし，多くの企業において，首尾一貫した方法で大量のデータを収集することは容易ではない．一般に，企業全体で一貫したデータ収集の体制を整えるためには長い年月を要する[PAULK1993]．米Software Engineering Institute（SEI）は，準備を整えるまでに平均43ヶ月を要すると報告している[CMUSE2005][※1]．また，大量のデータを蓄積するためには，長期にわたってデータ収集を継続する必要がある[BRIAND2000]．収集活動に大きな工数を要する一方で，データの蓄積が少ない間は見積りの精度が低く，得られる利益も小さい．早期に大きな利益を生み難い活動に大きな工数を捻出することは，多くの会社にとって容易ではない．

蓄積データが少ない組織において，高い精度で工数見積りを行う1つの方法は，他の組織で収集されたソフトウェア開発プロジェクトのデータを見積りの根拠として利用することである．本論文では，その1つのケーススタディとして，多くの日本のソフトウェア開発企業の協力のもとにSECが収集したデータ（SECデータ）を用いて工数見積りを行い，その精度を評価する．SECデータには，2005年3月現在，日本のソフトウェア開発企業15社から収集されたプロジェクト1,009件分のデータが含まれている．プロジェクトごとに，要した工数や日数等，約490の特性変数が記録され，蓄積されている[IPASEC2005]．

ただし，複数の異なる組織からデータを収集した場合，計測された特性変数が組織ごとに異なり，欠損値（収集されていない値）が多く含まれる可能性が高くなる[MENDES2004]．SECデータにおいても，全体の87.3%が欠損値であった．一般に，30%を超える欠損値が含まれる場合，見積精度が著しく低下するといわれている[KROMREY1994]．SECデータを用いる際には，この問題を解決し，多くの欠損値が含まれるデータを用いた場合でも，精度が低下しにくい見積手法を適用することが望ましい．

現在までに，欠損値の問題を考慮した特性見積りの研究がいくつか行われている[OHSUGI2004]，[STRIKE2001]，[TSUNODA2005]．Strikeら[STRIKE2001]は，各種見積手法を適用する際に欠損値の問題を解決する欠損値処理法（リストワイズ除去法，平均値挿入法等）を紹介し，その有効性を調査した．Strikeらは，代表的な見積手法である重回帰分析に対して欠損値処理法を用い，欠損値が全体の15%以下の場合はリストワイズ除去法，15%を超える場合は平均値挿入法等が有効であることを示した．Ohsugiら[OHSUGI2004]及び角田ら[TSUNODA2005]は，欠損値が多く含まれるデータに適した見積手法として，協調フィルタリングに基づく見積手法を提案した．そして，全体の60%が欠損値であるデータに提案手法を適用し，欠損値処理法を用いた重回帰分析よりも高い精度で見積りを行えることを確認した．

本論文では，見積手法として，重回帰分析，対数重回帰分析[STRIKE2001]，ニューラルネット[TAKADA1994]，協調フィルタリング[OHSUGI2004]を用い，SECデータにそれらを適用した場合の見積精度の評価実験について報告する．協調フィルタリング以外の手法については，欠損値処理法として平均値挿入法を適用し，さらにステップワイズ法により、見積りに用いる特性変数を取捨選択した．一方，協調フィルタリングでは，欠損値を含む特

†　奈良先端科学技術大学院大学情報科学研究科, NAIST
††　IPA/SEC, Software Engineering Center

※1　プロセス改善モデルとして著名なソフトウェア成熟度モデル（SW-CMM）によると，成熟度レベル3を達成することにより組織全体で一貫したデータを収集するための基盤が確立される．SEIは，成熟度レベル1から2への移行に平均23ヵ月，レベル2から3への移行に平均20ヶ月を要すると報告している．

性変数をありのまま用い，また，変数選択を行わず，すべての特性変数を用いた．実験では，相互検証法によって各手法の精度を評価し，精度に統計的な有意差があるかどうか、また、精度のばらつきはどの程度かを調査した．SECデータにおける各手法の見積精度を明らかにすることができれば，今後企業がSECデータを利用する際の指針になると期待される．

以降，2章では，本論文で用いた見積手法について説明する．3章では，精度評価実験の目的，評価指標，実験の手順について説明する．4章では，評価実験の結果と，結果に対する考察を述べる．5章で関連研究を紹介し，6章でまとめと今後の課題を述べる．

2　見積手法

2.1　重回帰分析

重回帰分析は多変量解析に基づく線形見積手法であり，ソフトウェア開発プロジェクトの工数や信頼性を見積もる際に広く利用されている[BRIAND 2000]，[WALSTON 1977]．重回帰分析を適用する場合，特性変数間の依存関係を分析し，次の式　のような回帰式を作成する．

$$\hat{v}_{a,b}=c_1 v_{a,1}+c_2 v_{a,2}+\cdots+c_n v_{a,n}+E$$

ただし，ここでは図1のような表形式のデータを用いて見積りを行うこととする．表中，p_1，p_2，…，p_mは各プロジェクト，m_1，m_2，…，m_nは各特性変数を表す．また，$v\,i\,j'$（i=1, 2,…, m, j=1, 2,…, n）はプロジェクトp_iについて記録された変数m_jの値（特性値）表す．式で，$\hat{v}_{a,b}$はプロジェクトp_aにおける変数m_bの見積値，$v_{a,j}$は見積りに影響する変数（説明変数）の値，c_jは係数，Eは定数項を表す．例えば，プロジェクト完遂に要する工数をm_bとして見積もる場合，ソフトウェアの規模，要員のスキル等が説明変数の候補（説明変数候補）となる．

本論文では，説明変数候補から説明変数を選択する際に，ステップワイズ変数選択を行った．この方法では，次のような説明変数の採用と除去を，回帰式の変化がなくなるまで繰り返す[TANAKA1995]．まず，目的変数と最も相関が高い説明変数候補を1つ採用し，説明変数とする．次に，残る説明変数候補から，式の寄与率を最も大きく増加させるものを採用し，回帰式を作成する．作成された回帰式中の各説明変数の係数が0かどうか検定し，有意水準10%で棄却できる場合に当該変数を除去する．

評価実験を行う際には，平均値挿入法により欠損値の問題に対処した．平均値挿入法は，欠損値を各変数の平均値で埋める方法である[LITTLE1987]．欠損値を埋める方法としては，最も単純かつ一般的な方法である[SALTON1983].

2.2　対数重回帰分析

対数重回帰分析は，重回帰分析を拡張した見積手法であり，プロジェクトの工数や信頼性の見積りに広く利用されている[BOEHM1984]．対数重回帰分析を適用する場合，次の式　のような回帰式を作成する．

$$\hat{v}_{a,b}=e^{(c_1,\log v_{a,1}+c_2,\log v_{a,2}\cdots+c_n\log v_{a,n}+E)}$$

対数重回帰分析を行う際は，特性値全てを底e（＝2.71828182845904）で対数変換して新たなデータを作成し，そのデータに重回帰分析を適用すればよい．評価実験を行う際には，重回帰分析と同様の変数選択法と欠損値処理法を適用した．

	m_1	m_2	…	m_j	…	m_b	…	m_n
p_1	$v_{1,1}$	$v_{1,2}$	…	$v_{1,j}$	…	$v_{1,b}$	…	$v_{1,n}$
p_2	$v_{2,1}$	$v_{2,2}$	…	$v_{2,j}$	…	$v_{2,b}$	…	$v_{2,n}$
…	…	…				…		…
p_i	$v_{i,1}$	$v_{i,2}$	…	$v_{i,j}$	…	$v_{i,b}$	…	$v_{i,n}$
…						…		…
p_a	$v_{a,1}$	$v_{a,2}$	…	$v_{a,j}$	…	$v_{a,b}$	…	$v_{a,n}$
…						…		…
p_m	$v_{m,1}$	$v_{m,2}$	…	$v_{m,j}$	…	$v_{m,b}$	…	$v_{m,n}$

図1　見積りに用いたm行n列の表

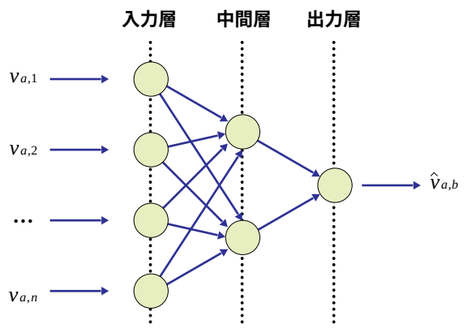


図2　見積りに用いたニューラルネット

2.3　ニューラルネット

ニューラルネットは，人工知能の分野で研究されてきた非線形見積手法である[FUNAHASHI1989]，[ROSENBLATT1958]，[RUMELHART1986]．重回帰分析は，説明変数が互いに独立であり，かつ，それらの線形結合によって目的変数との関係を表現できることを前提としている．しかし，プロジェクトの工数や信頼性の予測においては，説明変数は相互に影響を及ぼしあっている場合がほとんどであり，また，説明変数と目的変数の関係が線形であるという理論的根拠もない．一方，ニューラルネットは，説明変数の独立性は前提とされておらず，かつ説明変数間の非線形な関係を捉えることができるため，重回帰分析よりも高い精度で見積りを行える可能性がある[TAKABAYASHI1999]，[TAKADA1994].

評価実験で使用したニューラルネットは図2に示すように入力層，中間層，出力層からなる．各層の処理ユニットは，隣接する層の処理ユニットと重み付きリンクで接続されている．入力層から説明変数の値が入力され，出力層から見積値が出力される．入力層の各処理ユニットに入力された説明変数の値は，リンクの重みが乗算され，中間層の処理ユニットに送られる．中間層では，送られてきた値が和算された後，シグモイド関数（g（θ）=2/（1+exp（－θ））－1）により非線形変換される．変換された値は，再びリンクの重みが乗算され，出力層に送られる．出力層に送られた値の和がニューラルネットの出力となる[TAKADA1994].

リンクの重みの決定には，様々な学習アルゴリズムが提案されている．本論文では，米Integral Solutions社の統計処理ソフトウェアClementine 8.6に実装されている誤差逆伝播アルゴリズム[RUMELHART1986]を使用した．この方法では，既知のデータを使用し，次のようなリンクの重みの更新を繰り返す．まずニューラルネットの出力と見積値との間の残差平方和を計算する．次に，リンクの重みに対する残差平方和の勾配を計算し，残差平方和を減少させる方向にリンクの重みを変化させる[TAKADA1994].

学習に影響を与えるパラメータとして，中間層に含まれる処理ユニットの個数及び重みを更新する回数（学習回数）を調整する必要がある．処理ユニットの個数を増やすと，より複雑な関係を表現できる．しかし，リンクの重みが多くなるため，学習に使用するデータの量が少なければ，見積精度が低下する[FUNAHASHI1989]．また，処理ユニットの個数の増加に応じて，より多くの学習回数を設定する必要がある．評価実験を行う際には，中間層の処理ユニットの個数を2と3に変動させ，学習回数を10,000と50,000に変動させた．評価実験を行う際には，重回帰分析と同様の変数選択法と欠損値処理法を適用した．

2.4　協調フィルタリング

協調フィルタリングは，情報検索の分野で研究されてきた見積手法である[GOLDBERG1992]，[RESNICK1994]，[SARWAR2001]．協調フィルタリングを用いることで，過去に手がけた類似の案件の実績データをもとに，新規プロジェクトの工数を見積もる「類推見積り」を，系統的に行うことができる．重回帰分析やニューラルネットでは，ただ１つの予測式が作成されるのに対し，協調フィルタリングは予測対象プロジェクトごとに個別に予測式を構築するため，プロジェクトの個別性をより強く反映した予測が行える．また，欠損値の量や分布（欠損値の偏り）が変化しても精度は低下しにくいという特徴がある[KAKIMTO2004]．SECデータには多くの欠損値が含まれるため，重回帰分析やニューラルネットよりも高い精度で見積りを行える可能性がある．

協調フィルタリングは，　特性値の正規化，　プロジェクト間の類似度計算，　類似度に基づく見積値計算という3つの手順で実行される．各手順について，複数のアルゴリズムが提案されており，データの性質に応じて用いるアルゴリズムを変更する必要がある．精度評価実験では，手順　と　で用いるアルゴリズムを1つに固定し，とくに精度への影響が大きい手順　のアルゴリズムを変動させた．各手順で用いたアルゴリズムついて，以降で説明する．

特性値の正規化：データ中の各特性変数は値域が互いに異なる．各変数が見積りに対して与える影響を均一にするため，次式により，値域［0.0, 1.0］に正規化した特性値$v_{i,j}$を求めた．

$$v_{i,j}'=\frac{v_{i,j}-\min(m_j)}{\max(m_j)-\min(m_j)}$$

ただし，ここでは図1のような表形式のデータを用いて見積りを行うこととする．$v_{i,j}$はプロジェクトp_iにつ

いての特性変数m_jの値，max（m_j）は変数m_jの最大値，min（m_j）は変数m_jの最小値を表す．

プロジェクト間の類似度計算：見積り対象のプロジェクトp_aと見積りの根拠として用いる他の各プロジェクトp_iとの類似度sim（p_a,p_i）を求める．実験の際には，代表的なアルゴリズムであるAdjusted Cosine Similarity（式 とCorrelation Coefficient（式 を適用した[RESNICK1994].

$$\mathrm{sim}(p_a,p_i)=\frac{\sum_{j\in Ma\cap Mi}\left\{\left(v_{a,j}{}'-\overline{m_j}'\right)\times\left(v_{i,j}{}'-\overline{m_j}'\right)\right\}}{\sqrt{\sum_{j\in Ma\cap Mi}\left(v_{a,j}{}'-\overline{m_j}'\right)^2}\sqrt{\sum_{j\in Ma\cap Mi}\left(v_{i,j}{}'-\overline{m_j}'\right)^2}}$$

$$\mathrm{sim}(p_a,p_i)=\frac{\sum_{j\in Ma\cap Mi}\left\{\left(v_{a,j}{}'-\overline{p_a}'\right)\times\left(v_{i,j}{}'-\overline{p_i}'\right)\right\}}{\sqrt{\sum_{j\in Ma\cap Mi}\left(v_{a,j}{}'-\overline{p_a}'\right)^2}\sqrt{\sum_{j\in Ma\cap Mi}\left(v_{i,j}{}'-\overline{p_i}'\right)^2}}$$

ただし，M_aとM_iはそれぞれプロジェクトp_aとp_iについて記録された特性変数の集合を表す．また，$\overline{m_j}'$と$\overline{p_i}'$は特性変数$\overline{m_j}'$とプロジェクト$\overline{p_i}'$の正規化された特性値の平均値を表す．類似度sim（p_a, p_i）の値域は［−1.0, 1.0］である．式 ， では欠損していない変数のみを用いるため，欠損値処理法を用いる必要はない．

類似度に基づく見積値計算：手順 で求めた類似度sim（p_a, p_i）を用いて，プロジェクトp_aの特性変数m_bの見積値$\hat{v}_{a,b}$を計算する．実験では，式 に示すAmplified Weighted Sumを適用して見積値計算を行った．

$$\hat{v}_{a,b}=\frac{\sum_{j\in k\text{-nearest Projects}}\left(v_{a,j}{}'\times\mathrm{amplifer}\left(P_a,P_i\right)\times\mathrm{sim}\left(p_a,p_i\right)\right)}{\sum_{j\in k\text{-nearest Projects}}\mathrm{sim}\left(p_a,p_i\right)}$$

$$\mathrm{amplifer}\left(p_a,p_i\right)=\frac{\sum_{j\in Ma\cap Mi}\left(\frac{v_{a,j}}{v_{i,j}}\times\mathrm{correl}\left(m_b,m_j\right)\right)}{\sum_{j\in Ma\cap Mi}\mathrm{correl}\left(m_b,m_j\right)}$$

ただし，k-nearestProjectsは，特性変数m_bの値が収集されており，かつプロジェクトp_aと類似度が高いk個のプロジェクトの集合を表す．類似プロジェクト数kは見積精度に影響を与えるため，評価実験では，kを3から21の間で3ずつ変動させ，最も精度が高くなるk= 15を採用した．また，correl（m_b, m_j）は特性変数m_bとm_jの相関係数を表す．ただし，相関係数が負の値になる場合は0.0として計算した．

3 精度評価実験

3.1 目 的

今後多くの企業がSECデータを利用する際の1つの指針として，SECデータにおける各手法の見積精度を明らかにすることである．本実験では，代表的な見積手法である重回帰分析，対数重回帰分析，ニューラルネット及び近年有望視されている協調フィルタリングを評価対象とした．ただし，ステップワイズ重回帰分析とニューラルネットについては，データに欠損値が含まれる場合に見積りが行えないため，欠損値処理法（平均値挿入法）を適用した上で，ステップワイズ法による変数選択を行った．精度の評価指標は絶対誤差及び相対誤差を用いた．相互検証法（ホールドアウト法）によって10回の試行を行い，各手法の精度及び精度のばらつきを評価，比較した．

3.2 利用したデータ

評価実験ではSECデータを用いて，プロジェクトの完遂に要する工数（人時）の見積りを行った．表1に実験で用いたデータに含まれる特性変数を示す．見積対象の変数はm_0であり，見積りに用いた変数はm_1～m_{97}である．本来，SECデータには約490の変数が含まれている[IPASEC2005].

ここでは，詳細設計完了時に変数m_0を見積る場合を想定し，その時点で収集が終了していると思われる97個の特性変数を選出した．なお，総開発期間（月数）の実績数m_1は，実際には開発が終了しないと確定しないが，ここでは，開発期間（納期）が開発開始時にあらかじめ決定されることを想定し，特性変数に含めている．

選出した変数には，量的変数と質的変数が含まれている．前者は数値の大小に意味があり，加減算が可能な変数である．例えば，工数200人時は150人時に対して50人時の差がある．一方，後者は数値の大小に意味がない変数である．例えば，プロジェクト種別1（新規開発）とプロジェクト種別4（拡張）に3の差があるわけではない．選出した変数のうちm_0～m_{29}は量的変数，m_{30}～m_{97}は質的変数である．ここでは，質的変数をダミー変数化して共分散分析を行うことはせず，量的変数m_1～m_{29}のみを用いて重回帰分析を行った．

見積精度を評価する際，見積もった値（見積値）と実際に収集されたm_0の値（実測値）を比較するため，SECデータに含まれるプロジェクト全1,009件から，変数m_0が収集されているプロジェクト378件のデータを抽出し，実験データとして使用した．実験データには数社のプロジェクトが混在していた．全ての企業から収集された特性変数もあるが，いくつかの企業では収集されなかった変数もある．このため，データには多くの欠損値が含まれていた．

表1に実験データに含まれる欠損値の割合（欠損率）を特性変数ごとに示す．各特性変数の欠損率は，実験データ全378件のプロジェクト中，当該変数を記録しなかったプロジェクトの割合と等しい．実験データ全体の欠損率は約67%と非常に大きく，その分布にも大きな偏りがある．見積りに用いた変数m_1～m_{97}の内，欠損率が20%以下のものは全体の約9%しかない．一方で，欠損率が80%以上の変数は全体の約42%に及ぶ．また，変数m_4, m_{30}等は欠損値を全く含まない．一方で，変数m_2，m_3等は全体の90%以上が欠損値である．

3.3 評価指標

実験では，見積精度を評価する基準として，次の5つの評価指標を用いた．ただし，ここでは図1のような表形式のデータを用いて見積りを行うこととする．$\hat{v}_{a,b}$はプロジェクトp_aにおける特性変数m_bの見積値，$v_{a,b}$はその実測値を表す．また，mは見積もったプロジェクトの数を表す．

絶対誤差平均（MAE）：実測値と見積値の誤差の平均．値域は［0.0, ∞）であり，値が小さいほど見積精度が高いことを示す．

$$\mathrm{MAE}=\frac{1}{m}\sum_{a=1}^{m}\left|\hat{v}_{a,b}-v_{a,b}\right|$$

相対誤差平均（MRE）：誤差が実測値の何倍であるかを表す相対誤差の平均．値域は［0.0, ∞）であり，値が小さいほど見積精度が高いことを示す．

$$\mathrm{MRE}=\frac{1}{m}\sum_{a=0}^{m}\left|\frac{\hat{v}_{a,b}-v_{a,b}}{v_{a,b}}\right|$$

Pred25：見積値全体に占める，相対誤差0.25以下の見積値の割合．値域は［0.0, 1.0］であり，値が大きいほど，見積精度が高いことを示す．

$$\mathrm{Pred25}=\frac{1}{m}\sum_{a=0}^{m}\mathrm{isAccurate}\left(\frac{\hat{v}_{a,b}-v_{a,b}}{v_{a,b}}\right)$$

$$\mathrm{isAccurate}(x)=\begin{cases}1 & x \leqq 0.25\\ 0 & x > 0.25\end{cases}$$

3.4 手 順

実験では，相互検証法（ホールドアウト法）によって各手法の見積精度を評価した．2節で説明した見積手法ごとに，次の手順で評価指標を計算した．

*Es*1. 実験データを無作為に半数ずつに分割し，一方を学習データ，もう一方を試験データとする．

*Es*2. 学習データを用いて見積モデルを構築する．

*Es*3. *Es*2で作成した見積モデルを用いて試験データ中に含まれるプロジェクトの特性値$\hat{v}_{a,b}$（a=0, 1, ..., m）を見積もる．ただし，この際に試験データ中の実測値$v_{a,0}$は見積りに使用しない．

*Es*4. *Es*3で得た各見積値$\hat{v}_{a,b}$と，試験データ中の各実測値$v_{a,0}$を比較し，評価指標を計算する．

*Es*5. 実験結果の信頼性を向上させるため，*Es*1から*Es*4を1回の試行として10回繰り返す．各手法について，10回の試行から得た評価指標の平均値を算出する．

各手法について算出した評価指標の平均値の差の検定を行い，精度に統計的な有意差があるかどうかを調査した．また，10回の試行で得た各評価指標について箱ひげ図（boxplot）を描き，各手法を用いた場合の精度のばらつきを比較した．得られた結果に基づき，その意味と限界について考察を行った．

4 実験の結果と考察

評価実験の結果，10回の試行から得られた各評価指標の平均値を表2に示す．表2では，各列に評価指標を記し，各行にはMREの値が小さい（精度が高い）ものから順に見積手法を記した．行と列が交差するセルには，各評価指標の平均値と共に（ ）内に直下のセルに記した平均値との差の検定によって算出したp値を記した．また，各セルに記した値が，直下のセルに記した値よりも有意水準5%で有意に精度が高い場合，そのセルの値を斜体と下線によって強調した．

表1　評価実験に用いたデータに含まれる特性値と各特性の欠損率

	特性値の種類	欠損率		特性値の種類	欠損率
m_0	実績工数#総計人時#_プロジェクト全体	0.00%	m_{49}	アーキテクチャ	0.00%
m_1	月数#実績#プロジェクト全体	42.86%	m_{50}	開発対象プラットフォーム	33.60%
m_2	利用者数	97.88%	m_{51}	Web技術の利用	44.97%
m_3	利用拠点数	97.62%	m_{52}	オンライントランザクション処理	96.83%
m_4	FP実測値_調整前	0.00%	m_{53}	主開発言語	21.16%
m_5	改修FP値_母体FP	89.15%	m_{54}	DBMSの利用	47.88%
m_6	ILF実績値_FP	60.32%	m_{55}	開発ライフサイクルモデル	0.00%
m_7	EIF実績_FP	61.90%	m_{56}	運用ツール利用	91.27%
m_8	トランザクションファンクション実績値_機能数	74.07%	m_{57}	類似プロジェクト_有無	91.53%
m_9	トランザクションファンクション実績値_FP	65.34%	m_{58}	プロジェクト管理ツール_利用	65.34%
m_{10}	データファンクション実績値_機能数	73.28%	m_{59}	構成管理ツール利用	69.05%
m_{11}	データファンクション実績値_FP	63.23%	m_{60}	設計支援ツール利用	69.05%
m_{12}	設計書文書量要件定義書	81.22%	m_{61}	ドキュメント作成ツール利用	69.05%
m_{13}	設計書文書量基本設計書	81.22%	m_{62}	デバッグ_テストツール利用	69.05%
m_{14}	設計書文書量詳細設計書	80.69%	m_{63}	上流CASEツール利用	97.09%
m_{15}	他規模指標_DBテーブル数	82.80%	m_{64}	統合CASEツール利用	97.09%
m_{16}	他規模指標_画面数	81.22%	m_{65}	コードジェネレータ利用	95.77%
m_{17}	他規模指標_帳票数	83.07%	m_{66}	開発方法論利用	95.24%
m_{18}	月数#実績#要件定義	76.72%	m_{67}	要求仕様_明確度合	52.12%
m_{19}	月数#実績#基本設計	77.25%	m_{68}	ユーザ担当者_要求仕様関与	52.65%
m_{20}	月数#実績#詳細設計	66.93%	m_{69}	ユーザ担当者_システム経験	69.84%
m_{21}	実績工数#開発#システム化計画	83.86%	m_{70}	要求仕様_ユーザ承認有無	97.35%
m_{22}	実績工数#開発#要件定義	37.83%	m_{71}	ユーザ担当者_設計内容理解度	97.35%
m_{23}	実績工数#開発#基本設計	21.43%	m_{72}	設計_ユーザ承認有無	97.35%
m_{24}	実績工数#開発#詳細設計	26.19%	m_{73}	ユーザ担当者_受け入れ試験関与	69.05%
m_{25}	外注実績#工数#要件定義	92.06%	m_{74}	要求レベル_信頼性	67.72%
m_{26}	外注実績#工数#基本設計	83.07%	m_{75}	要求レベル_使用性	97.09%
m_{27}	外注実績#工数#詳細設計	87.30%	m_{76}	要求レベル_性能・効率性	61.64%
m_{28}	レビュー指摘件数基本設計	98.94%	m_{77}	要求レベル_保守性	97.09%
m_{29}	レビュー指摘件数詳細設計	99.47%	m_{78}	要求レベル_移植性	97.35%
m_{30}	開発プロジェクト種別	0.00%	m_{79}	要求レベル_ランニングコスト要求	97.35%
m_{31}	母体システム安定度	70.11%	m_{80}	要求レベル_セキュリティ	67.99%
m_{32}	開発プロジェクト形態	0.00%	m_{81}	法的規制有無	69.58%
m_{33}	受託開発作業場所	64.02%	m_{82}	PMスキル	67.72%
m_{34}	新規顧客	86.77%	m_{83}	要員スキル_業務分野経験	61.90%
m_{35}	新規業種・業務	86.77%	m_{84}	要員スキル_分析・設計経験	67.99%
m_{36}	新規協力会社	88.62%	m_{85}	要員スキル_言語・ツール利用経験	61.90%
m_{37}	新技術利用	87.83%	m_{86}	要員スキル_開発プラットフォーム使用経験	61.90%
m_{38}	役割分担_責任所在	67.46%	m_{87}	主なFP計測手法	4.76%
m_{39}	達成目標_優先度_明確度合	67.99%	m_{88}	FP計測手法実績値の純度	5.03%
m_{40}	作業スペース	67.99%	m_{89}	FP計測_支援技術	47.62%
m_{41}	プロジェクト環境_騒音	67.99%	m_{90}	テスト体制	71.16%
m_{42}	業種	29.63%	m_{91}	定量的出荷品質基準_有無	93.92%
m_{43}	業務種類	57.41%	m_{92}	SLOC実測値_コメント行取り扱い	85.19%
m_{44}	システム用途	95.24	m_{93}	SLOC実測値_空行取り扱い	85.19%
m_{45}	利用形態	0.00%	m_{94}	品質保証体制_基本設計	91.53%
m_{46}	システム種別	0.00%	m_{95}	品質保証体制_詳細設計	90.74%
m_{47}	業務パッケージ_利用有無	29.89%	m_{96}	品質保証体制_結合テスト	89.95%
m_{48}	処理形態	61.64%	m_{97}	品質保証体制_総合テスト（ベンダ確認）	91.01%

表2　10回の試行から得た評価指標の平均値．MAE，MREは値が小さいほど精度が高いことを示す．Pred25は値が大きいほど精度が高いことを示す．(　)内は直下のセルの値との差の検定により算出したp値を示す．

	MAE	MRE	Pred25
協調フィルタリング（Adjusted Cosine Similarity）	*5637.185* (0.017)	*0.642* (0.000)	0.301 (0.069)
協調フィルタリング（Correlation Coefficient）	*5863.709* (0.002)	0.771 (0.417)	*0.286* (0.001)
対数重回帰分析	8102.984 (0.439)	*0.855* (0.014)	0.180 (0.129)
ニューラルネット（3ユニット，50,000回）	9096.571 (0.151)	1.281 (0.222)	0.221 (0.261)
ニューラルネット（2ユニット，50,000回）	*8020.968* (0.010)	1.428 (0.339)	0.218 (0.369)
ニューラルネット（2ユニット，10,000回）	8135.367 (0.640)	1.473 (0.595)	0.209 (0.426)
ニューラルネット（3ユニット，10,000回）	*10279.519* (0.014)	*1.584* (0.002)	*0.201* (0.000)
重回帰分析	12315.063	6.247	0.101

結果として，実験データに対しては協調フィルタリングの見積精度が最も高いことがわかった．対数重回帰分析，ニューラルネットは評価指標によって精度の優劣が異なり，また統計的な有意差もMREを除いて観察できなかった．また，重回帰分析は最も見積精度が低いことが分かった．協調フィルタリングを適用する場合には，Adjusted Cosine Similarityを類似度計算アルゴリズムとして用いた場合の方がCorrelation Coefficientを用いた場合よりも高い精度を示した．ニューラルネットを適用する場合には，中間層のユニット数を3，学習回数を50,000回とした場合が最も高い精度を示した．

評価実験の結果，10回の試行から得られた各評価指標のばらつきを図3，図4，図5にboxplotとして記す．図中では，ニューラルネットをNN，協調フィルタリングをCFと略記した．図では，横軸に各手法を列挙し，縦軸が各評価指標の値を表す．各手法について，最小値，第1四分位数，中央値，第3四分位数，最大値を求め，対応する軸に記した．図によると，協調フィルタリング，とくにAdjusted Cosine Similarityを類似度計算アルゴリズムとして用いた場合に，ばらつきなく高い精度で見積りを行えたことがわかる．さらに，対数重回帰分析はMREのばらつきがニューラルネットよりも小さく，ニューラルネットはMAEやPred25のばらつきが対数重回帰分析よりも小さいことがわかる．また，重回帰分析については，他の手法より精度が低く，ばらつきも大きいことがわかる．

評価実験では，協調フィルタリングが他の見積手法よりも高い見積精度を示した．これは，過去の類似のプロジェクトのデータに基づいて類推見積りを行うという協調フィルタリングのアプローチが有効であったことに加え，実験データに含まれる欠損値が他の手法の精度を低下させたためと考えられる．重回帰分析やニューラルネットを用いた場合，学習データに基づいて見積り対象の特性変数（今回の場合はプロジェクト完遂に要する工数）に大きな影響を与える変数（今回の場合はプロジェクトの月数等）を調べ，数学的な見積モデルを作成する．見積り対象のプロジェクトで，見積値に大きな影響を与える特性変数が欠損している場合，欠損値処理法によって挿入された平均値などに基づいて見積りが行われるため，精度が著しく低下する．一方，協調フィルタリングの場合には，欠損していない変数だけを用いて見積りを行うため，欠損値が多いプロジェクトに対してもある程度高い見積精度を示したものと考えられる．

今回の実験において最も精度が高かった協調フィルタリングでは，MREが0.642，Pred25が30.1%であった．これらの数値は，1つの企業内だけで収集されたデータに基づく工数予測事例と比べても，遜色のない精度である．

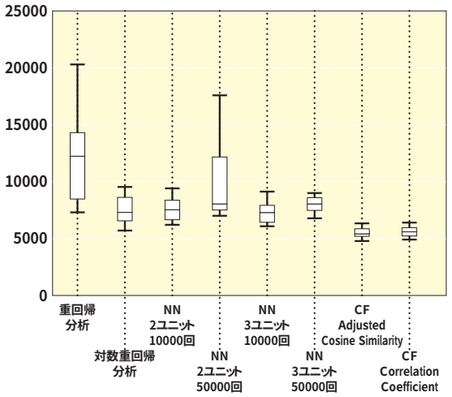


図3　MAEのboxplot

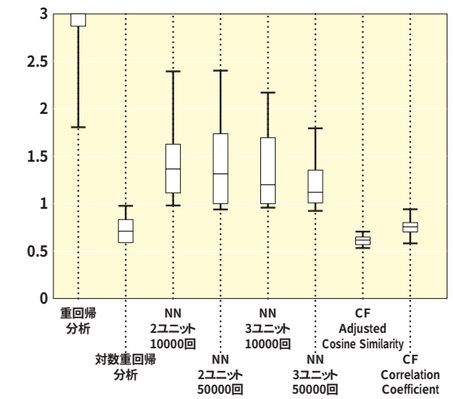


図4　MREのboxplot

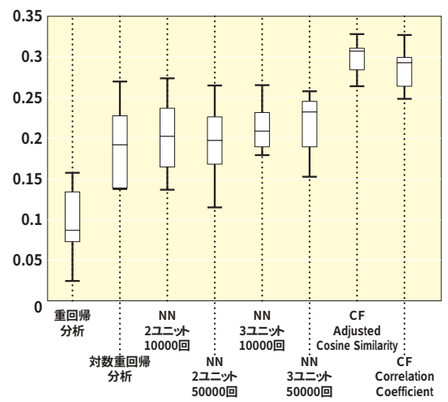


図5　Pred25のboxplot

参考となる事例として，文献[TSUNODA2005]では企業内データを用いて，試験工程開始までに得られるプロジェクト特性値に基づいて試験工数の見積りを行っており，そのMREが0.79，Pred25が37%であった．実験条件は異なるものの，本実験ではより早い工程（詳細設計完了時）での見積りを行ったにもかかわらず，遜色のない精度が得られたといえる．

ただし，MREが0.642であるということは，工数の見積値が平均して64.2%の誤差を含むということであり，現場へ適用する際には注意が必要である．今後，さらなる分析が必要であるが，相対誤差が大きいプロジェクトの多くは，規模が小さなものであった．ある程度以上の規模を持つプロジェクトであれば，より精度の高い見積値が得られる可能性がある．

また，今回の実験は，あくまでも1つのケーススタディであり，SECデータから抽出した378件のプロジェクトの特性値を用いた場合の結果である点に注意する必要がある．SECと同様，多数の異なる企業からプロジェクトのデータを収集している事例として，海外ではISBSGが知られている[ISBSG]．これら他のデータセットを用いて本論文と同様の評価実験を行うことが今後の課題である．

5　関連研究

単一企業から収集したデータを用いた場合と，複数企業から収集したデータを用いた場合の見積精度を比較する研究が行われている．複数企業からデータを収集した方が多くのデータを利用できる反面，データの一貫性が低下しやすく，見積精度が低下する可能性が高い[BRIAND2000]．今回，実験に用いたデータには複数企業のデータが含まれていたが，単一企業のデータのみを用いることで見積精度を改善できる可能性がある．複数企業のデータを用いた場合の見積精度と，単一企業のデータを用いた場合の見積精度を比較調査することは，今後の課題の1つである．

本論文で取り上げた見積手法の他にも，ソフトウェア開発プロジェクトの特性を見積もる手法が提案されている．Amasakiら[AMASAKI2003]は，重回帰分析等の見積手法を用いる場合に問題となる欠損値の問題を解決するため，ベイジアンネットを用いた特性の見積手法を提案した．Amasakiらは実プロジェクトから収集したデータに対して提案手法を適用し，ソフトウェアの品質見積に利用可能であることを確認した．SECデータには非常に多くの欠損値が含まれるため，ベイジアンネットによって高い精度で見積りを行える可能性がある．ベイジアンネットをSECデータに適用した場合の精度を評価することは，今後の課題の1つである．

また，本論文で取り上げた欠損値処理法の他にも，いくつかの欠損値処理法が提案されている．Strikeら[STRIKE 2001]は，データの欠損率を変化させ，平均値挿入法を含む3種類の欠損値処理法を重回帰分析と共に適用した場合の精度を調査した．他の欠損値処理法をSECデータに適用し，見積精度を評価することも，今後の課題の1つである．

6　おわりに

本論文では，企業横断的収集データに基づくソフトウェア開発プロジェクトの工数見積りの1つのケーススタディとして，SECデータを用いて複数の手法により工数の見積りを行い，その精度を評価した．その結果，協調フィルタリングは他の3つの手法（重回帰分析，対数重回帰分析，ニューラルネット）よりも高い精度を示し，実績工数に対する見積り工数の相対誤差の平均値が0.642，Pred25が30.1%となった．

今後は，ISBSGなど他のデータセットを用いた場合との比較，単一企業のデータのみを用いた場合との比較，ベイジアンネット等の他の見積手法の評価等を行っていきたい．

謝辞

本研究の一部は，SECとの共同研究の成果並びに文部科学省「e-Society 基盤ソフトウェアの総合開発」の委託により行われた研究成果に基づく．本論文の執筆に際し，見積手法や評価実験に関するご意見をいただいた奈良先端科学技術大学院大学情報科学研究科 柿元健氏に深く感謝する．また，統計手法全般にわたってご助言をいただいた同志社大学文化情報学部 宿久洋博士に深く感謝する．

参考文献

[AMASAKI2003] Amasaki, S., Mizuno, O., Kikuno, T., and Takagi, Y.：A Bayesian Belief Network for Predicting Residual Faults in Software Products, In Proc. of the 14th Int'l Symposium on Soft. Reliability Eng., pp.215-226, 2003
[BOEHM1984] Boehm, B.W.：Software Engineering Economics, IEEE Trans. on Soft. Eng., vol.10, no.1, 4-21, 1984
[BRIAND2000] Briand, L.C., Langley, T., Wieczorek. I.：A Replicated Assessment of Common Software Cost Estimation Techniques. In Proc. of the 22nd Int'l Conf. on Soft. Eng., pp.377-386, 2000
[CMUSEI2005] Carnegie Mellon University and Software Engineering Institute：Process Maturity Profile: Software CMM 2004 Year End Update, 2005
[CONTE1986] Conte, S.D., Dunsmore, H.E., and Shen, V.Y.：Software Engineering Metrics and Models. The Benjamin/Cummings Publishing Company, Inc., Menlo Park, California, 1986
[FUNAHASHI1989] Funahashi, K.：On the Approximate Realization of Continuous Mappings by Neural Networks, Neural Networks, vol.2, no.3, pp.183-192, 1989
[GOLBERG1992] Goldberg, D., Nichols, D., Oki, B.M., and Terry, D.：Using Collaborative Filtering to Weave an Information Tapestry, Comm. of the ACM, vol.35, no.12, pp.61-70, 1992
[IPASEC2005] 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：ソフトウェア開発データ白書2005 ～IT企業1000プロジェクトの定量データを徹底分析～, 日経BP社, 2005
[ISBSG] Software Benchmarking Standards ISBSG Homepage, http://www.isbsg.org/
[KAKIMTO2004] 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のロバスト性評価, 第11回ソフトウェア工学の基礎ワークショップ, 2004
[KROMREY1994] Kromrey, J., and Hines, C.：Nonrandomly Missing Data in Multiple Regression: An Empirical Comparison of Common Missing-Data Treatments, Educational and Psychological Measurement, vol.54, no.3, pp.573-593, 1994
[LITTLE1987] Little, R., and Rubin, D.：Statistical Analysis with Missing Data, John Wiley & Sons, Inc., 1987
[MENDES2004] Mendes, E., and Kichenham, B. A.：Further Comparison of Cross-Company and Within Company Effort Estimation Models for Web Applications, In Proc. of the 10th Int'l Soft. Metrics Symposium, pp.348-357, 2004
[OHSUGI2004] Ohsugi, N., Tsunoda, M., Monden, A., and Matsumoto, K.：Applying Collaborative Filtering for Effort Estimation with Process Metrics, In Proc. of the 5th Int'l Conf. on Product Focused Soft. Process Improvement, Springer, Berlin Heidelberg, pp.274-286, 2004
[PAULK1993] Paulk, M., Curtis, B., Chrissis, M., and Weber, C.：Capability Maturity Model for Software （Version 1.1）, CMU/SEI-93-TR-024, 1993
[RESNICK1994] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J.：GroupLens: An Open Architecture for Collaborative Filtering of Netnews, In Proc. of the ACM Conf. on Computer Supported Cooperative Work, 175-186, 1994
[ROSENBLATT1958] Rosenblatt, F.：The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Psychological Review, vol.65, No.6, pp.386-408, 1958
[RUMELHART1986] Rumelhart, D.E., Hinton, G.E., and Williams, R.J.：Learning Representations by Back-propagating Errors, Nature, vol.323, pp.533-536, 1986
[SALTON1983] Salton, G., and McGill, M.：Introduction to Modern Information Retrieval, p.448, McGraw-Hill, New York, 1983
[SARWAR2001] Sarwar, B.M., Karypis, G., Konstan, J.A., and Riedl, J.：Item-Based Collaborative Filtering Recommendation Algorithms, In Proc. of the 10th Int'l World Wide Web Conf., pp.285-295, 2001
[SHEPPERD1997] Shepperd, M., and Schofield, C.：Estimating Software Project Effort Using Analogies, IEEE Trans. on Soft. Eng., vol.23, no.12, pp.76-743, 1997
[STRIKE2001] trike, K., El Eman, K., and Madhavji, N.：Software Cost Estimation with Incomplete Data. IEEE Trans. on Soft. Eng., vol.27, no.10, pp.890-908, 2001
[TAKABAYASHI1999] Takabayashi, S., Monden, A., Sato, S., Matsumoto, K., Inoue, K., and Torii, K.：The Detection of Fault-Prone Program Using a Neural Network, In Proc. of the Int'l Symposium on Future Software Technology'99, pp.81-86, 1999
[TAKADA1994] 高田義広, 松本健一, 鳥居宏次：ニューラルネットを用いたソフトウェア信頼性予測モデル, 電子情報通信学会論文誌D-I, vol.J77-D-1, no.6, pp.454-461, 1994
[TANAKA1995] 田中豊, 垂水共之（編）：Windows版 統計解析ハンドブック 多変量解析, p.240, 共立出版, 東京, 1995
[TSUNODA2005] 角田雅照, 大杉直樹, 門田暁人, 松本健一, 佐藤慎一：協調フィルタリングを用いたソフトウェア開発工数予測方法, 情報処理学会論文誌, vol.46, no.5, pp.1155-1164, 2005
[WALSTON1977] Walston, C., and Felix, C.：A Method of Programming Measurement and Estimation, IBM Systems Journal, vol.1, pp.54-73, 1977

論文 Paper

通信ソフトウェア開発におけるプロセス改善のためのフィールド品質に注目した主要な活動要因の抽出




菊地 奈穂美†,††,‡　安藤 津芳†　水野 修††　菊野 亨††

本論文では，高信頼性が要求される通信ソフトウェアの開発プロジェクトに対し，プロダクトのフィールド品質を改善するのに有用な開発活動に関する要因を特定する試みについて述べる．ここでは，プロダクトのフィールド品質をそのプロダクトをリリース後の6ケ月間にユーザから報告された問題（クレーム）の数で判断する．提案するアプローチでは，まずフィールド品質の良しあしでプロジェクトをGoodとFairの2つのグループに分類し，それぞれのグループの代表的なプロジェクトへのインタビューを行った．その結果明らかになったフィールド品質と関わりのある4つの問題について，それを計測するメトリクスを決定した．引き続いて，メトリクスデータを収集し，フィールド品質との関連をロジスティック回帰モデルで記述した．次に，24個のプロジェクトを対象として評価実験を行い，品質に強い影響を与えると思われる主要な要因として，レビュー活動，母体の品質の改善，及び仕様に基づく活動の3点を確認した．さらに，モデルによるフィールド品質予測の精度を評価する実験を行ったところ，精度が約96%になることを確認した．

Key Factors in Process Management for Improving the Field Quality of Telecommunications Software Development

Nahomi Kikuchi†,††,‡,　Tsuyoshi Ando†,　Osamu Mizuno††, and Tohru Kikuno††

This paper presents an approach for identifying and quantitatively evaluating the key factors of projects that improve the field quality of a software product, for high reliable telecommunications software projects. For this purpose, we define field quality as the number of problem reports received within six months of the release of a software product. The approach involves classification of projects based on field quality and interviews with project managers. Since the problems related to the field quality were identified through the interview results, we determined the metrics that corresponded to the identified problems. We then collected the metrics data, conducted a logistic regression analysis, and constructed a logistic regression model related to the field quality. Finally, we examined the model using the data obtained from 24 projects. As a result of this analysis, the metrics concerning the performance of review activity, the origin's quality, and the specification related activity were confirmed to be effective. The accuracy of the constructed model with regard to the quality prediction for the data obtained from all the 24 projects was approximately 96%.

Key Words & Phrases：　Process improvement, Software metrics, and Quality

†　沖電気工業株式会社, Oki Electric Industry Co., Ltd.
††　大阪大学大学院情報科学研究科, Graduate School of Information Science and Technology, Osaka University
†††　IPA/SEC, Software Engineering Center
‡　大阪大学大学院基礎工学研究科, Graduate School of Engineering Science, Osaka University

1　はじめに

通信システムは企業におけるビジネス活動に必要なネットワークを担うものであり，広く社会インフラとして利用されることから，通信システムのソフトウェアには高い品質が求められ，特に高い信頼性と機能性が要求される．近年，通信サービスの多様化，ビジネススピードの加速化に伴って，納期遅延はサービスプロバイダ企業の通信サービスの開始時期の遅れを意味するため，通信ソフトウェア開発プロジェクトを納期どおりに実現することは必須であると言っても過言ではない．

一方，ソフトウェア製品の品質は，顧客企業やエンドユーザの視点から見て，リリース後の問題数で評価されることが多い（文献[QUEST2003] p.44 問題報告数を参照）．品質は開発プロセスの様々な活動を実行した結果を反映している．実際，ソフトウェア製品の品質に関する研究[ARTHUR1993, EBENAU1993]では「十分なレビューとテストは良い品質につながる」ことが指摘されている．また，統計的モデル[MUSA1987, MIZUNO2000]でプロジェクトのリスク分析が試みられている．

本研究の目的は，通信ソフトウェアの開発において，フィールド品質の良否に影響するプロジェクトの活動を特定することである．高品質のソフトウェア製品の開発を目指したプロセス改善において，品質の向上に有用である活動特性を定量的に把握することは非常に意義が大きい．そのような要因を見つけることができれば，企業としても組織横断的に開発プロセスの改善に取り組むことができる．さらに，各プロジェクトマネージャ（PM）としては，抽出された要因をリスクと考えて，そのプロジェクトを改善するための計画，管理，制御を行うことが可能になる．

ここで提案するアプローチでは，最初に，フィールド品質との関連で考慮しておくべき開発プロセスの特徴を述べる．次に，フィールド品質の良しあしでプロジェクトをGoodとFairの2つのグループに分類し，各グループの代表的なプロジェクトについて現場のプロジェクトマネージャへのインタビューを行う．インタビューの結果に基づいてフィールド品質に影響を与える問題を明らかにし，その問題を評価するためのメトリクスを決定する．引き続いて，メトリクスデータを収集し，ロジスティック回帰分析によってフィールド品質への影響の大きい要因を特定する．

本論文の残りは次のような構成となっている．2節では通信ソフトウェア開発の特徴と本研究の目的を述べる．3節で関連研究について議論する．4節は，フィールド品質に関する要因分析のアプローチの概要を示す．5節において，プロジェクトの分類とインタビューによる特徴の分析方法を説明する．続いて6節に，要因候補をもとに新しいメトリクスを設計し，メトリクスデータ収集の過程を示す．7節でロジスティック回帰分析モデルによる分析を行う．最後に8節でまとめを述べる．

2　準　備

この節では対象のソフトウェア及び開発プロセスの特徴を述べる．

2.1　通信ソフトウェア

一般的に通信ソフトウェアはサービスレイヤ，共通プラットフォームレイヤから成る．サービスレイヤは通常，複数個の機能ブロック部とアプリケーション共通部をもつ．多種にわたる通信サービスはこれらの機能ブロックの組合せによって実現されている．

あるシステムの開発において新しいソフトウェアが開発されれば，その後の他システムや後継バージョンシステムで再利用されることが多い．サービスレイヤや共通プラットフォームレイヤ内の既存ソフトウェア部分で，当該システムの開発時点において開発のベースとして使用される部分を母体と呼ぶ[PARK1992].

本論文で対象としている通信ソフトウェアプロジェクトでは，機能ブロック，アプリケーション共通部，共通プラットフォームレイヤの開発を行っている．開発するソフトウェアに焦点を当ててみると，新規開発，既存部の改造による開発（流用開発，再利用という）から構成

されている．

プロジェクトの開発作業の特徴についてまとめると次のようになる．

(1) 新規開発： 新規に設計され，プログラムコードが作られ，テストも十分に行われる．

(2) 母体利用の開発：さらに次のように分類される．

(a) 改造： 既存の設計やコードを削除や追加により変更する．変更に関わる部分を含めて全体をレビューし，テストする．

(b) 流用： 既存の設計やコードへの変更はごく限られた範囲で行う．変更に関わる部分を限定的にレビューし，テストも品質確保のためにある程度行われる．

(c) 再利用： システム全体として統合されて動くことになるが，この部分について注目した設計の見直しやテストはほとんど行わない．

母体利用の開発が特に通信ソフトウェア開発では多くなりがちであり，こうした部分に着目した分析を行うことが必要とされている．

2.2 開発プロセス

対象とする組織の開発プロセスはウォーターフォール型である．まず，方式検討工程でソフトウェアの主な要件を決定し，ソフトウェア開発計画を作成する．続いて，概要設計，機能設計，詳細設計，プログラミング及び単体テスト，結合テスト，システムテストの工程が続いている．各工程では，成果物（設計書等のドキュメント，プログラム，テスト仕様書）が作成されており，それらに対してレビューが実施される．プロジェクトマネージャは，各工程で進捗を評価し，報告書を作成する．すべてのプロジェクトでは，各工程において工数，期間，問題数等のプロセスデータ，及びソフトウェア規模，リソース使用量，性能数値等のプロダクトデータが収集される．さらに，リリース後にはフィールドからの問題報告が収集される．本論文では，このフィールド問題の数の大きさ（5.1節でフィールド品質と定義する）を改善するのに有効と思われる要因を明らかにすることを目指す．

通信ソフトウェア開発プロジェクトは，関連するハードウェアやファームウェア，またはインタフェースをもつ別のソフトウェアシステムの開発と並行して行われることが多い．そのため，関連システムからは，開発対象のソフトウェア部分に対してインタフェース仕様の変更要求が開発の途中の段階でしばしば発生する．また，実際のネットワーク等の環境がない開発初期工程においては，性能に関する仕様の実現性を確認することが困難である．

2.3 研究の目的

本研究の目的は，高信頼性が要求される通信ソフトウェアの開発プロセスを対象に，フィールド品質を改善するのに有用な要因を特定することである．ソフトウェアの品質は開発プロセスの実行結果であるので，ソフトウェア開発プロセスのマネジメント及び技術の2つの面の活動を対象とするべきだと考えている．そのために，実際の開発プロジェクトを分析し，フィールド品質に密接な関連のある重要な要因を明らかにする．それらは通信ソフトウェアに特有の要因だけでなく，他のソフトウェアにも共通の要因も含むと考えている．

次に，実用面への配慮から，限られた時間内に重要な要因を特定できることも重要である．具体的には，開発現場の専門家の参加は2時間以下程度の短時間に抑える必要がある．

最後に，ここでは，ある企業に固有の要因を見つけることを当面の目的とする．つまり，あらゆる種類のソフトウェア開発に適用できるような一般的な要因を求めることは目指していない．それは我々の将来の研究課題の1つである．

3 関連研究

ソフトウェアプロジェクトやプロセス改善に有用な要因を発見する手法が既に幾つか提案されている．以下にそのうちの主要なものを示す．

Khoshgoftaarら[HUDEPOHL1996]はリリース後の問題を予測することに焦点を当てている．具体的には，ソースコードから得られるメトリクスを分析して，既存部分で問題を起こす可能性の高いモジュールを予測している．このアプローチはソースコードを利用できる場合には適切であり，母体部分が大きい場合には有用と思われる．しかし，システムの再利用率が低い場合には適用の効果は限定的である．また，通信ソフトウェアのようなリアルタイム性の高いシステムでは動作時の複合的なタイミングによる問題の対応が重要であるが，ソースコードのスタティックな解析のみではそれらは検出が困難である．

COQUALMO[BOEHM2000]はCOCOMO[BOEHM2000]の要因を利用することでソフトウェア製品中の残存不具合数を予測する見積りモデルである．このモデルでは不具合除去の手法の種類や人員，計画，製品，プラットフォームの特徴などを変化させてその影響を見ることが可能である．残念ながら，この手法は品質改善のための活動要因を特定していない．

CoBRA[BRIAND1998]は生産性のための見積りモデルを開発する手法である．このモデルでは品質の見積りについては行っていない．さらに，コストオーバーヘッドの見積りの為に主観的要因群を利用しているが，そのための専門家へのグループインタビューは大がかりになりがちで，例えば10名近い専門家が参加して合計で約2日もかかる．一方，我々のアプローチではPMO（プロジェクト管理オフィス）スタッフの知識とPMOが保有するデータを利用することにより，各専門家は2時間程度の協力で可能となる．

高木ら[TAKAGI2005]やWohlinら[WOHLIN2003]の研究ではリスク質問表を利用して集めた主観的な回答データを用いてプロジェクトの混乱・成功を特徴づけている．これらの研究ではリリース後の品質を直接的には考慮していないため，我々とは目的が異なる．

CMM[PAULK1995]は多数の企業における知見に基づいており，組織改善のための規範的プラクティスを与えている．しかし，CMMを用いた正式なアセスメントは，結果を得るまでに約1週間を要する．また，品質に関わる重要な要因を明確には特定できない．従って，これも我々の目的とは異なる．

4 フィールド品質の要因分析の概要

フィールド品質に関する属性を特定するアプローチの概要を図1（a）に示す．図1（b）には，それらの詳細な説明を描いている．上段には活動内容を，下段にはその実施者を書いている．

Step1 プロジェクトの分類: PMOスタッフがプロジェクトの分類を行う．フィールドで報告される問題の数に基づいて，品質がよいもの（Good）とそうでないもの（Fair）に分類する．

Step2 インタビューによる分析: GoodとFairのそれぞれから代表的なプロジェクトを選んで，それらの特徴を把握するために，PMOスタッフがプロジェクトマネージャへインタビュー（1〜2時間程度）を行う．インタビューで確認する質問表は事前に作成しておく．質問項目には，PMBOK [PMI2000]を参考に管理面の観点及びJIS X 0160 [日本工業標準調査会審議1996]の開発プロセスの内容を観点として取り入れる．その後でGoodとFairプロジェクトのインタビュー結果を分析することで，Fairプロジェクトの特徴を定性的に導く．

Step3 メトリクスの設計: Step2で確認した定性的な特性を反映（あるいは表現）するようにメトリクスを設計する．各メトリクスはさらに幾つかの要因に分解されている．直感的には，各要因がプロジェクトの開発や管理の活動に対応している．

Step4 メトリクスデータの収集: PMOから規模やプロセスメトリクスデータを収集する．メトリクスの値の計算（実際には各要因に対応する質問表の評価）をプロジェクトマネージャに記入してもらう．記入された評価内容についてはPMOスタッフが妥当であるかを確認する．

Step5 ロジスティック回帰分析による分析: 収集データに対してロジスティック回帰分析を行い，主要な特徴を分析する．

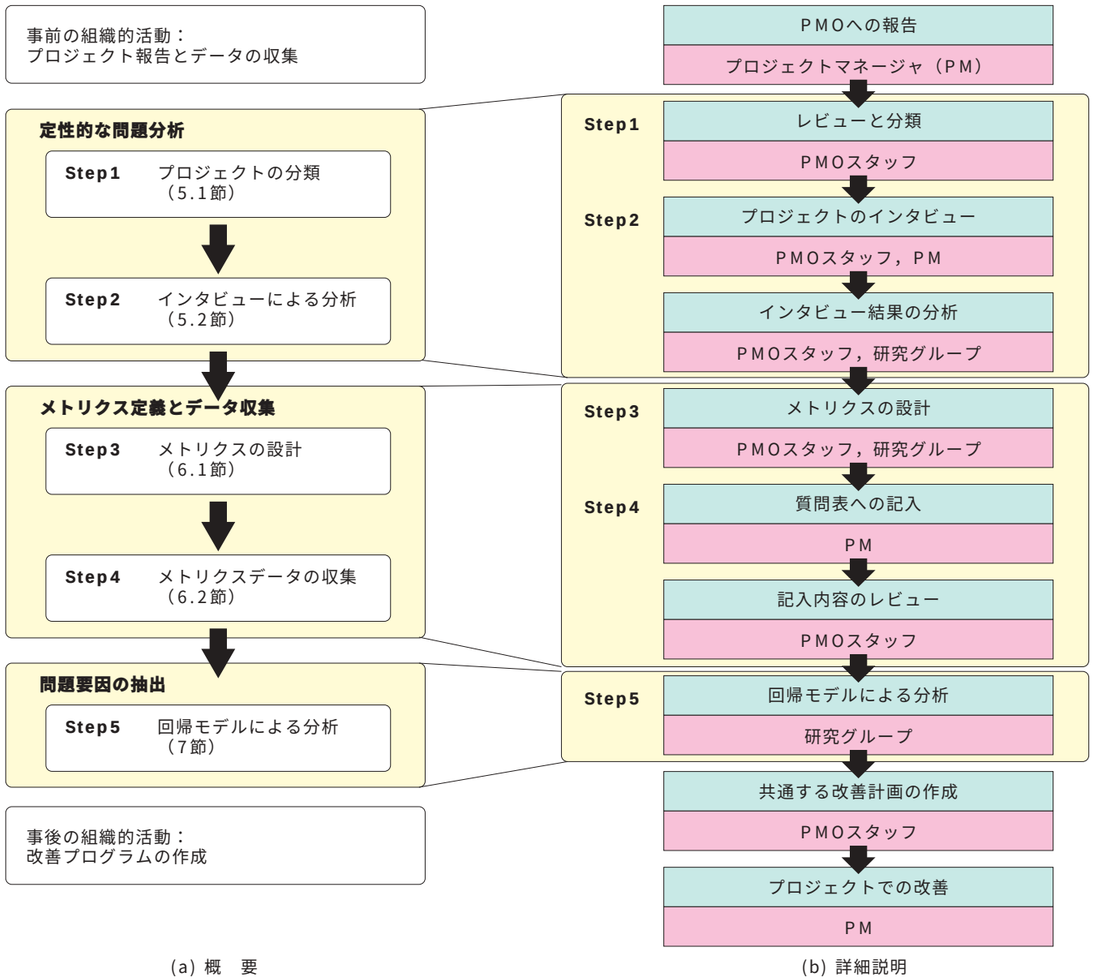
5　定性的な問題分析

5.1　プロジェクトの分類

ここではリリース後にユーザから出てくる報告の内で，ソースコードの調査と修正を必要とするものを「フィールド問題」と呼ぶ．そして，フィールド問題の数を「フィールド品質」と呼び，その大小に基づいてプロジェクトをGoodとFairに分類する．そのために，リリースして6ヶ月の間に発生したフィールド問題の数を表すメトリックとしてN_6を採用する．

N_6：フィールド問題の数（リリースから6ヶ月間）

N_6に基づくプロジェクトの分類を決定するために，PMOの品質専門スタッフが2つの基準値c_1及びc_2を決定した（その数値は社外秘により，記述することができない）．2つのパラメータN_6とc_1を用いて，プロジェクトをGoodとFairに分類するルールを表1に示す．表1では3つ目のパラメータc_2を導入して，FairをさらにAverageとPoorに分類するルールも合わせて示している．なお，この組織では，品質の管理をGood，Average，Poorの3段階で行っているが，Averageに含まれるプロジェクトの数が非常に少なかったため，本論文ではGoodとFairの2つに分類する．



(a) 概　要　　(b) 詳細説明

図1　アプローチの概要

フィールド問題の数を使用するねらいは，顧客の視点から見た時に，問題の絶対数が開発側及び顧客の双方に共有できる指標であるためである[QUEST2003]．

5.2　インタビューによる分析

Good及びFairプロジェクトの違いを調べるために，幾つかの分析を行った．GoodグループとFairグループから，それぞれ3つのプロジェクトを選択した．そして，両方のグループのプロジェクトについてプロジェクトマネージャにインタビューを行った[KIKUCHI2000]．このインタビューの実施にあたっては，あらかじめ準備したインタビュー用質問項目を使った．このインタビュー用質問項目で洗い出す観点を含めておき，定性的に，品質のよいプロジェクトと悪いプロジェクトでの差を検出できるようにした．

その結果，Fairグループのプロジェクトを特徴づけることができる，次のような4つの主要な問題を抽出することができた．

(1) 問題1（母体の品質確認と改善の不備）

母体自体でフィールド問題が出ている．例えば母体の品質を改善するための設計やテストが十分に行われていない．また，母体の改造を行う場合に変更影響を確認するための開発活動が十分ではない．とくに，母体の比率が高くかつ品質が悪い場合，母体の変更部分のみの設計とテストを行うことでは不十分である．

(2) 問題2（テストとレビュー活動の不足）

通信ソフトウェアではとくに重要な異常系動作やタイミングの複合条件などの条件を考慮したレビューやテストが十分ではない．これらは運用時の信頼性で重要となる．さらに，ソースコードレビューと単体テストが少なく，プロジェクト全体での実施の徹底も低い．また，現実のネットワーク環境と同じテスト環境構築が困難なために，シミュレーションによるテストが行われるが，その場合にはシミュレータの仕様の正確性が重要な鍵となる．

(3) 問題3（プロジェクト管理と計画の不備）

開発計画や報告書の作成が遅れ気味になる．また，仮に計画が作成されても，十分に詳細ではないことが多い．一方，開発作業に重点が置かれるために，プロジェクト管理が手薄になる傾向にある．プロジェクト管理に遅延が起きても，プロジェクトの外からそれを見ることは難しく，適切な対応策をPMOが検討するのが困難となる．実際にあるプロジェクトでは工数の実績値が計画値と比較して15%以上超過するのがみられた．そこでは開発するソースコード規模は見積られていたが，母体の変更規模見積りの正確さは低かった．

(4) 問題4（仕様変更の多さ）

設計仕様の変更がプログラミングやテスト工程でも起きることがあった．仕様変更の許可が厳密なルールの下で行われない時期があったと考えられる．その結果として，遅い時期での変更を招いた．他の原因としては設計段階の検討で運用時の動作環境や条件の想定漏れ，共通プラットフォームへのインタフェースの解釈違いなどがあった．これらの影響が後のフェーズでの仕様変更として現れる．

表1　プロジェクトの分類ルール

グループ		条　件
Good		$N_6 < C_1$
Fair	Average	$C_1 \leqq N_6 < C_2$
	Poor	$C_2 \leqq N_6$

表2　問題とメトリクスの対応

5.2の問題	メトリクス
問題1	N, N_6, N_0, M_O
問題2	M_r, M_t
問題3	D_E, M_m
問題4	M_s

6　メトリクス定義とデータ収集

6.1　メトリクスの設計

この節では5.2節で示した4つの問題に対応してメトリクスを導入する．表2に4つの問題と導入した9個のメトリクスの対応関係を示す．

各メトリクスは複数個の要因から構成される．直感的には，この要因がソフトウェア開発の各工程での開発者の品質に関わる活動の良しあしを確認するものとなっている．また，基本的に各要因はリッカート尺度[FENTON 1997]で定義する．各要因は，例1と例2で説明するように，詳細な意味の定義を与えており，評価をなるべく客観的に行えるようにしている．具体的には，ポイント0からポイント3までで評価される．ポイント0が最良の状態で，ポイントが大きくなるに従い悪くなることを示す．

（1）まず，フィールド品質とその母体の品質に関する4つのメトリクスを示す．

N ：フィールド問題の総数

N_6：フィールド問題の数（リリース後6ヶ月間の問題報告の数[※1]）

N_o：母体のフィールド問題の数（具体的には，母体ソフトウェアのN_6を母体設計時の新規開発と改造の規模の和で割った値，と定める）．

Mo：母体の品質を改善する活動内容の評価

o_1：母体品質のよさの度合い

o_2：母体品質を設計段階で調査し考慮する度合い

o_3：母体品質を改善する設計・テスト活動の度合い

o_4：仕様変更による母体への影響の検討の十分さ

例1　メトリクスM_oの要因o_2の評価基準を次に示す．

－ポイント0：母体への影響等を十分に評価

－ポイント1：母体への影響等の評価は60%程度

－ポイント2：母体への影響等の評価は30%程度

－ポイント3：母体への影響等の評価はしていない

（2）次に，レビューとテストに関する2つのメトリクスを示す．

M_r：レビューの活動内容の評価

r_1：アーキテクチャレビューの実施時期の適切さ

r_2：設計レビューの実施時期の適切さ

r_3：レビューの計画的実施と結果の記録と管理の適切さ

r_4：設計レビューでの異常系，準正常のケースの考慮の十分さ

r_5：レビュー手法の使用の度合い

M_t：テストの活動内容の評価

t_1：異常ルートや運用レベルでの条件の十分さ

t_2：テスト仕様書の作成の度合い

t_3：ソースコードレビュー及び単体テスト実施の度合い

t_4：テスト仕様のレビューの十分さ

t_5：テスト実施の管理の適切さ

t_6：静的解析ツールの活用の度合い

t_7：動的解析ツールの活用の度合い

例2　メトリクスM_tの要因t_6の評価基準を次に示す．

－ポイント0：全ソースコードを対象に解析

－ポイント1：新規作成及び変更部分のみ解析

－ポイント2：新規作成部分の限定部分のみ解析

－ポイント3：ツールによる解析なし

（3）次に，プロジェクトの管理に関する2つのメトリクスを示す．

D_E：工数の計画値と実績値の差

M_m：プロジェクト管理に関する活動内容の評価

m_1：計画の作成時期，更新の適切さ

m_2：作成する計画書の情報（帳票種類）の十分さ

m_3：品質と性能の計画の度合い

m_4：工数見積の方法の適切さ

m_5：進捗評価の頻度の度合い

m_6：PMOへの報告の頻度の度合い

m_7：構成管理の方法と頻度の度合い

m_8：変更管理の方法の適切さ

（4）最後に，仕様の変更に関するメトリクスを示す．

M_s：仕様・設計の活動の内容の評価

s_1：仕様が確定される時期の適切さ

s_2：仕様の明確さの度合い

s_3：要求仕様に対するアーキテクチャの検討及び実現可能性検討の十分さ

メトリクス M_s，M_r，M_t，M_o，M_m の値は各メトリクスのもつ要因のポイントの合計値とする[MOLLER1993].

6.2　メトリクスデータの収集

我々は，次の3つの条件を満たす24プロジェクトを選択し，メトリクスデータの収集を行った．

納期どおりに完了している[※2]．

ほぼ同じ開発プロセスに従っている．

ほぼ同じ設計方法と開発言語（C，C++）を使用している．

実際のメトリクスデータの計測は，プロジェクトの開発計画書及び報告書から得られるデータを参考にして，プロジェクトマネージャが算定して行った．その算定結果についてはPMOスタッフがレビューしている．

なお，24プロジェクト中Goodとされたものは15件，Fairとされたものは9件であった．

次節の分析では，「プロジェクトのフィールド品質がGoodかFairか」という2値の目的変数Yを使用する．この目的変数Yの値がGoodになるか，それともFairになるのかは，ロジスティック回帰モデルで確率を計算することで求められる．説明変数の候補として7つのメトリクス N_O，M_s，M_r，M_t，M_o，M_m，D_E を考える．

7　問題要因の抽出

7.1　ロジスティック回帰モデル

多変量ロジスティック回帰モデル[MIZUNO2000, BASILI 1996] は一般に次の式で与えられる：

$$E(Y \mid x_1, \cdots, x_n) = \frac{e^{b_0+b_1x_1+\cdots+b_nx_n}}{1+e^{b_0+b_1x_1+\cdots+b_nx_n}}$$

ここで，目的変数 Y は今回の場合，プロジェクトの状態がGood か Fairであったかを表す．また，説明変数 x_i は6節で挙げたメトリクスに対応させる．このとき，条件付き確率$E(Y|x_1,\cdots,x_n)$ を求めることができる．

24個のプロジェクトから収集されたデータを利用して，上式の係数 b_i をステップワイズ法によって推定し，次式に示すロジスティック回帰モデルを作成した．作成されたモデルは統計的に有意なものであった．

$$E(Y \mid N_O, D_E, M_s, M_r, M_m) = \frac{e^{-6.93+1.73N_O+4.18D_E+0.71M_s+0.20M_r-0.14M_m}}{1+e^{-6.93+1.73N_O+4.18D_E+0.71M_s+0.20M_r-0.14M_m}}$$

ステップワイズ法によって選択された変数とその係数を表3にまとめる．

表3からは，2つのメトリクスN_OとD_Eが他のメトリクスに比べて高い係数をもつことが読み取れる．つまり，メトリクスN_OとD_Eがプロジェクトの状態がGoodかFairかを判別するのに有効であることを示している．

表3　推定された係数

メトリクス	係　数
M_s	0.71
M_r	0.20
M_m	－0.14
N_O	**1.73**
D_E	**4.18**

表4　精度の評価結果

	品質予測結果	
実際の品質	Good	Fair
Good	15	0
Fair	1	8

※1 本論文で分析対象としているソフトウェアは，リリース後の実運用開始後に即使用されるものが殆どであり，6ヶ月間で十分フィールド品質を確認できると考えている．なお，当該企業ではリリース後毎月の品質を追跡しており，長期運用システムでは1年以上経過後も追跡している．

※2 納期どおりに完了とは，プロジェクト側の理由（開発サイドの開発遅れ等）で，予定していた完了日（納期）を延ばすような変更をすることなく，納期までに提供できるように開発作業を行ったことを表す．

であるのかを決定するのに大きく影響を与えていると判断できる．また，メトリクスM_sとM_rも意味のある変数として選択されたことがわかる．一方，メトリクスM_mは負の係数を持って選択されている．しかし，係数の値は－0.14と比較的小さいため，モデル全体に与えている影響は小さいと判断できる．

この結果，4つのメトリクスN_O，D_E，M_s，M_rがソフトウェア品質に影響を与える主要な要因として抽出できる．ここで，これらのメトリクスが特に通信系のソフトウェア開発において抽出されたことの意味については次節で考察する．

引き続いて，作成されたモデルと収集された24プロジェクトのデータを用いて，このモデルの品質予測の精度を評価する．ここでは評価手法としてモデル作成時に利用したデータへの自己適用を行った．表4に示す結果は，モデルによって算出された確率$E(Y|N_O, D_E, M_s, M_r, M_m)$を基準値0.5を用いて分類したものである．すなわち，あるプロジェクトについての評価値Eが$E > 0.5$を満たせば品質をFairと判定し，そうでなければ品質をGoodと判定する．評価の結果，24プロジェクト中の23プロジェクトで正しく予測された．つまり，精度は約96％と計算でき，極めて高い精度をもつモデルであると考えられる．抽出した要因によるモデルでプロジェクトの分類を精度よく説明できることからも，抽出した要因は適切であると考える．もちろん，モデルによる予測で全プロジェクトの予測が正しく行えたわけではないが，本手法による品質予測が実用的な観点からも有効であることが確認できた．

7.2 考　察

7.1節の結果より，以下に示すような幾つかのことが確認できた．

母体品質（メトリクスN_O）がフィールド品質に強い影響を与えているという仮説を，7.1節の分析と評価の前に立てていた．実験の結果はこの仮説を強く支持するもので，母体品質に注目することがフィールド品質の改善に有用であるとのPMOの主張の正当性を確認した．

また，工数の予測と実績の差（メトリクスD_E）が別の強い要因として抽出された．このことから，「品質が安定しないプロジェクトではテスト工程で追加のテストを行うという対策を取ることによって大幅に工数が膨らむ」という現象が，ある程度は説明されたと考える．

通信ソフトウェアの複雑な条件の考慮不足や仕様確定の遅れによる品質の安定の遅れ（メトリクスM_rとM_s）が選択されたことは，「テスト工程での品質安定にしわ寄せがきて，それが最終的にはフィールド品質に影響する」というPMOの見解とも一致した．とくに，M_sではs_3において通信系での性能など（例えば，パケットの処理での通信特有部分の仕様について）を含めた実現可能性を聞いている．また，M_rでは通信ソフトで品質の鍵となるr_4（設計レビューでの異常系，準正常のケースの考慮の十分さ）が考慮されている．また，そのレビュー実施の時期のr_1，r_2も間接的ではあるが通信ソフトウェアの特徴と考えられる．こうした特徴は，例えばMISシステム（業務アプリケーション）の開発においてはあまり問題にならない種類のものと考えられる．

一方，今回選択されなかったプロジェクト管理（メトリクスM_m）に関しては技術的な側面（WBSの作成やスケジュールの管理）だけではなく人的側面も重要になる[KREZNER1998]．本研究では人的側面や管理的な側面（例えば，契約の管理など）は含んでいない．その主な理由は，今回利用したプロジェクトではそうした違いがほとんど無かったからである．しかし，今後別の組織への適用を行う際には，こうした側面についても考慮する必要があると考える．

また，今回抽出したメトリクスに基づくプロセス改善につながる具体的な施策について，次のようにPMO中心に検討した．改善の実行策を検討する際に，抽出したメトリクスの係数を参考にして改善施策実行の優先度を考慮した．

N_O: 母体を改善する調査やテストの活動としては，母体の品質確認と改善のためのコード解析実施の徹底

D_E 及び M_s: 開発プロセス実施状況の第三者レビューの実施．

M_r: 仕様の検討及びレビューでの重要な観点についてチェックリスト採用（通信系のもの）の組織内展開

こうした施策はすべて，開発現場においてすでに現実に着手している．

8　むすび

本研究では，通信ソフトウェアのプロジェクトを対象に，フィールド品質に影響を与える主要な要因を特定する試みを行った．

実際のプロジェクトデータを用いた評価実験を通じて，4つのメトリクスがフィールド品質に強く影響を与えていることを確認した．この結果は，プロジェクトマネージャがプロジェクト管理上の決断を行う上で非常に重要な基準を与えるものと期待される．また，開発組織においては，開発プロセス改善のための足がかりにすることが可能である．実際に，本研究のデータ収集に協力した組織では，この分析の結果を受けて，抽出された要因に対する組織横断的な改善活動を開始している．

我々のアプローチは，少数の的を絞ったメトリクスは使用するが基本的には大量のメトリクス収集を必要としない．そのため，これまでに大量の種類のデータを蓄積していない組織においても容易に適用可能であると考えている．また，示した方法は，それぞれの組織の特徴に合わせたメトリクスの選定と活用の指針として有用であると考える．

謝辞

本研究は，沖電気工業株式会社と大阪大学の共同研究契約によって一部サポートされており，協力頂いた皆様に感謝いたします．有益なご指摘をくださいました査読者の方々に深謝いたします．

参考文献

[ARTHUR1993] Arthur, L. J.：Improving Software Quality －An Insider's Guide to TQM－, John Wiley & Sons, 1993
[BASILI1996] Basili, V. R., Briand, L. C., and Melo, W. L.：A Validation of Object-Oriented Metrics as Quality Indicators, IEEE Trans. on Software Eng., vol. 22, no.10, pp.751-761, 1996
[BOEHM2000] Boehm, B. W. et al.：Software Cost Estimation with COCOMO II, Prentice Hall PTR, 2000
[BRIAND1998] Briand, L. C., Emam, K. E. and Bomarius, F.：COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking and Risk Assessment, In Proc. of 20th International Conference on Software Engineering (ICSE 1998), pp.390-399, 1998
[EBENAU1993] Ebenau, R. G. and Strauss, S. H.：Software Inspection Process, McGraw-Hill, 1993
[FENTON1997] Fenton, N. E. and Pfleeger, S. L.：Software Metrics: A Rigorous & Practical Approach, 2nd Edition, PWS publishing, 1997
[HUDEPOHL1996] Hudepohl, J. P., Aud, S. J., Khoshgoftaar, T. M., Allen, E. B.,and Mayrand, J.：Integrating Metrics and Models for Software Risk Assessment, In Proc. of 7th International Symposium on Software Reliability Engineering (ISSRE 96), pp.93-97, 1996
[KERZNER1998] Kerzner, H.：Project Management, 6th edition, John Wiley & Sons, 1998
[KIKUCHI2000] Kikuchi, N., Mizuno, O., and Kikuno, T.：Identifying Key Attributes of Projects that Affect the Field Quality of Communication Software, In Proc. of 24th Annual International Computer Software and Applications Conference (COMPSAC 2000), pp.176-178, 2000
[MIZUNO2000] Mizuno, O., Kikuno, T., Takagi, Y., and Sakamoto, K.：Characterization of Risky Projects based on Project Managers' Evaluation, In Proc. of 22nd International Conference on Software Engineering (ICSE 2000), pp.387-395, 2000
[MOLLER1993] Moller, K. H. and Paulish, D. J.：Software Metrics - A Practitioner's Guide to Improved Product Development, IEEE Press (Chapman & Hall), 1993
[MUSA1987] Musa, J. D., Iannino, A., and Okumoto, K.：Software Reliability: Measurement, Prediction, Application, McGraw-Hill, 1987
[PARK1992] Park, R. E.：Software Size Measurement: A Framework for Counting Source Statements, CMU/SEI-92-TR-20, 1992
[PAULK1995] Paulk, M. C., Weber, C. V., Curtis, B., and Chrissis, M. B.：The Capability Maturity Model: Guidelines for Improving the Software Process, Addison-Wesley, 1995
[PMI2000] Project Management Institute著, PMI東京支部監訳：プロジェクト・マネジメントの基礎知識体系ガイド(PMBOK(R)2000), Project Management Institute, 2000
[QUEST2003] QuEST Forum著, TL9000翻訳委員会訳：TL9000 品質マネジメントシステムハンドブック リリース3.5 －電気通信産業分野における測定法－, 日本規格協会, 2003
[TAKAGI2005] Takagi, Y., Mizuno, O., and Kikuno, T.：An empirical Approach to Characterizing Risky Software Projects on Logistic Regression Analysis, Empirical Software Engineering, Vol.10, No.4, pp.495-515, 2005
[WOHLIN2003] Wohlin, C. and Andrews, A. A.：Prioritizing and assessing Software Project Success Factors and Project Characteristics using Subjective Data, Empirical Software Engineering, Vol.8, No. 4, pp.285-303, 2003
[日本工業標準調査会審議1996] 日本工業標準調査会審議：ソフトウェアライフサイクルプロセスJIS X 0160-1996, 日本規格協会, 1996

ソフトウェア機能規模測定法の最新動向

ISO/IECやJISの観点からファンクションポイント法を概観する

NTTアドバンステクノロジ株式会社　コアネットワーク事業本部システム統括部　担当部長
日本ファンクションポイントユーザ会（JFPUG）会長
西山　茂

機能規模測定法は、ファンクションポイント法に対応するISO/IEC（又はJIS）の概念である。機能規模はソフトウェアの規模の一表現であり、その規模は、ソフトウェアを測定し、分析をするための基礎となる重要なメトリクス（メジャー）である。このため、機能規模測定は、ソフトウェア開発の重要な位置を占める。本稿では、機能規模測定法の技術と最近の動向を概観する。

1. はじめに

今日、ソフトウェアは産業の中で極めて重要な位置を占めており、ソフトウェアの工学（技術）的取り扱いは不可避な課題である。工学的に扱うためには、対象を認識する必要があり、認識するためには、測定が必要である。しかし、測定は、ソフトウェア関連の技術の中で最も遅れている分野であり、残念ながら工学的扱いもあまり進んでいない。

それでも、ソフトウェアを測定しようという動きは活発に続いている。ソフトウェアの様々な測定対象の中でも、規模は、他の測定結果の分析をするための基礎となる重要なメトリクス（メジャー）である。このため、ISO/IECやJISでもこれを規格化している。

本稿では、ソフトウェアの機能規模測定について、ISO/IECやJISの観点を含めてその技術内容及び手法の動向について概観する。

2. ファンクションポイント法

本節では、ファンクションポイントの定義と様々なファンクションポイント法について概観する。

2.1 ファンクションポイント法の定義

ファンクションポイント法※1とは、ソフトウェアの機能（ファンクション）に注目し、これを数値化することにより、ソフトウェアの規模を獲得する技術である。

この手法は、米国IBMのA.J.Albrecht（オルブレクト）が1979年に、アメリカ・IBMのユーザ団体であるGuide/Shareで報告したのが最初の公開であるとされている［ALBRECHT1979］。その後、この手法はアメリカの団体International Function Point Users Group（IFPUG）［IFPUG］に引き継がれて、維持管理され、今日に至っている。これからわかるように、この手法は、極めて歴史のある手法である。

この後、ファンクションポイント法は、様々なバリエーションが考案されたが、以下の3種類に分類できる。

① 基本的にAlbrecht（あるいはIFPUG）の手法と同一の手法
② Albrecht（あるいはIFPUG）の手法を引き継ぎつつ変更を加えた手法
③ Albrecht（あるいはIFPUG）の手法に刺激され新たに考案した手法

これらについては、後述する。

IFPUGの最新マニュアル［FP4.1.1］によれば、ファンクションポイント法とは次のことである。

○ファンクションポイント：アプリケーションソフトウェアの機能規模を表す尺度
○ファンクションポイント法：顧客の視点からソフトウェア開発や保守を測定するための標準的手法

一方、公式、非公式にかかわらず様々なファンクションポイント法が考案・利用されてきたため、ファンクションポイント法の標準化を行う動きが生まれ、1994年に国際標準化団体であるISO/IECのソフトウェア技術の標準化を担っているJTC1/SC7の中に作業部会ISO/IEC JTC1/SC7/WG12が設置された。この作業部会が今日に至るまで多くの国際規格を開発してきた。これについては後述する。

ISO/IEC JTC1/SC7/WG12が開発した国際規格では、ファンクションポイント及びファンクションポイント法は、より一般的に、機能規模及び機能規模測定法と呼ばれている。本稿でもこれ以降、IFPUGが維持管理している手法を指すとき以外は、機能規模あるいは機能規模測定法という用語を用いる。

ISO/IEC 14143－1：1998によれば、機能規模、機能規模測定、機能規模測定法は次の様に定義されている。

○機能規模（Functional Size）　利用者機能要件を定量化して得られるソフトウェアの規模。
○機能規模測定（Functional Size Measurement, FSM）機能規模を測定するプロセス。
○利用者機能要件（Functional User Requirements）利用者要件の部分集合。利用者機能要件は、利用者の要求を満足するためにソフトウェアが実現すべき利用者の業務及び手順を表す。品質要件及び技術要件は除く。
○FSM手法（FSM Method）本規格の要求事項に適合し、規則の集合によって定義されるFSMの特定な実現。

ファンクションポイントあるいは機能規模は、その手法が定める規則をソフトウェアドキュメントに対して適用して取得するため、本稿ではこの行為を「測定」と称することにする（これは広く了解されている）。

機能規模がソフトウェアの規模かどうかは難しい問題である。しかし、ソフトウェアの機能に着目し、それを数量化していること、これまでの研究［NISHIYAMA 1994］,［ALBRECHT1983］,［KEMERER1987］により機能規模と開発工数に正の相関が認められることを考慮すると、機能規模は、ソフトウェアの“機能”に着目した規模であると言って問題ないと考える。

2.2 機能規模測定法の種類

Albrechtの発表以来、様々な機能規模測定法が生まれたのは、次の理由による。

a）規則の明確化あるいは利用者に合わせたカスタマイズ
b）規則の簡略化
c）測定対象に合わせた測定規則の大幅な変更あるいは新規の考案

a）はファンクションポイント法の規則が自然言語で記述されているため、曖昧であったり、オリジナル版の開発時には想定外であったため規則がない等から、もとの規則に変更を加えたようなものが多い。

b）は、次の動機によるものが多い。

b1）ファンクションポイントを求めるための規則を適用する工数が多いため、規則を簡略化し、測定の工数を削減したい。

b2）ソフトウェア開発で情報が少ない段階、すなわち通常機能規模を測定する段階よりさらに上の段階（工程）で、機能規模を測定・推定したい。

c）は、次の動機によるものが知られている。

c1）測定値（機能規模）の利用法に適合するよう、測定対象ソフトウェアの構造の捉え方（ソフトウェアモデル）を変える。

c2）ファンクションポイント法あるいは類似の手法が不得意とする分野のソフトウェアの機能規模を測定する。

最も多く使われている機能規模測定法のオリジナルはAlbrechtが考案し、IFPUGが維持管理しているIFPUG法（ファンクションポイント法）であると言われている。しかし、これをそのまま利用している例はなく、上の分類でいうa）に属するものが多い。

b）に属するものは、アメリカ・SPR社が開発したSPR法（b1）、オランダ・NESMA［NESMA］※2のEstimated Function Point Counts（NESMAのサイトでは「概算法」と訳されている。以下ではこの訳を用いる）（b1）、NESMAのIndicative Function　Point Counts（概算法と同様の理由から「試算法」の訳を用いる）（b2）がある。また、オーストリア・ISBSG［ISBSG］※3が出版している“Software Project Estimation”にある、IFPUG法の測定要素（単位）の統計的な分布（割合）に着目して、ファンクションポイントを求める方法もある（b2）。

この他にも多数あるとみられるが、多くは機能規模測定を適用している企業内で工夫して使用している例が多く、あまり公表されていないようである（筆者もかつて実験的に（b2）に属するものを開発したことがある）［NISHIYAMA1995］。

c）に属するものには、C. R. Symonsが考案し［SYMONS1991］、現在UKSMA［UKSMA］※4が維持管理しているMkII　FPA（c1）、A.　Abran等を中心とするCOSMIC［COSMIC］※5グループが普及を推進しているCOSMIC - FFP（c2）がある。

以上をまとめて表1に示す。

※1「ファンクションポイント法」は、“Function Point Analysis”に対して先達が与えた訳である。この訳では英語の“Analysis”（分析）を「法」としているが、当を得た訳であると言えるだろう。

※2　NESMA：Netherlands Software Metric Association
※3　ISBSG：International Software Benchmarking Standards Group
※4　UKSMA：United Kingdom Software Metric Association
※5　COSMIC：Common Software Metric International Consortium

3. 各種のFSMの概略

本節では、2.2節で述べた代表的な機能規模測定法について詳述する。

3.1 機能規模測定法

機能規模測定法は、次を備えている。

① 測定対象の機能的なソフトウェアモデル（測定対象のソフトウェア構造）。

② ソフトウェアモデルの最小要素（ISO/IECの規格ではBFC（Base Functional Component）と呼ぶ）に数値を与え、それを統合化する手順。

また、機能規模測定は、次のように行われる。

i ）測定対象の利用者要件書、機能仕様書等からBFCを抽出する。

ii ）手法が決めた手順に従ってBFCに数値を割り当てる。

iii）割り当てられた数値を統合化する。

以下の代表的な機能規模測定法の説明では、①ソフトウェアモデル、②数値化の手順を述べる。

3.2 IFPUG法（ファンクションポイント法）

広範に利用されている機能規模測定法である。単にファンクションポイント法といえば、IFPUG法を指すことが多い。

（1）ソフトウェアモデル

IFPUG法では、次の5つのBFCを規定している。

① EI（外部入力）：いわゆる入力

② EO（外部出力）：いわゆる出力

③ EQ（外部紹介）：簡単な入出力の組

④ ILF（内部論理ファイル）：測定対象ソフトウェア（IFPUG法ではアプリケーションと呼ぶ）が維持管理するファイル（DB等も）

⑤ EIF（外部インタフェースファイル）：測定対象ソフトウェア以外のソフトウェアが維持管理するファイル（DB等も）

IFPUG法のソフトウェアモデルを図1に示す。IFPUG法は、いわゆる事務処理計算（アメリカではMIS（Management Information System）と呼ぶようである）のソフトウェアを念頭においているのでIFPUG法の最適な測定対象ソフトウェアは、事務処理計算系であるといわれている。

（2）数値化手順

抽出した、EI、EO、EQ、ILF、EIFのそれぞれに表2のようなテーブルを用いて数値を割り当てる（評価する）。この割り当ては、扱うデータ項目数やファイル数等によって行う。表2はEI用であるが、他の4つのBFC全てに対して表が用意されている。表の説明は省略する。

次に、抽出された全てのBFCの要素に割り当てられた数値を合計する。この値を未調整ファンクションポイント（UFP）と呼ぶ。

さらに、次の式を用いて、最終的なファンクションポイントを求める。

UFP×（0.01×一般システム特性の評価値の合計＋0.65）

ここで、一般システム特性は、システムの開発に影響を与える14の要因であり、その各々を0～5点で評価する。

これで、IFPUG法の測定は終了である。極めて単純であるが、実際には、仕様書等からBFCの抽出と評価に労力の大半がつぎ込まれる。

筆者は、一般システム特性は、機能規模を求めることと、それを使って工数予測を行うことが（手法開発時の状況から確信犯的に）混同された結果であり、本来の機能規模はUFPであると考えている。この考えは、ISO/IECの国際規格でも支持されている。

表1　機能規模の種類と実手法

分類		手法
オリジナル		IFPUG法、NESMA法
規則の明確化あるいはカスタマイズ		一般的なIFPUG法
規則の簡略化	工数削減	SPR法、NESM概算法
	上流工程適用	NESMA試算法、ISBSGの要素割合法（筆者による命名）
測定規則の大幅な変更または新規の考案	ソフトウェアモデルの変更	MkII　FPA
	対象ソフトウェアの変更・拡大	COSMIC-FFP

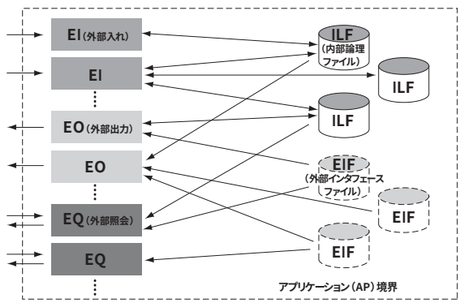


図1　IFPUG法のソフトウェアモデル

3.3 COSMIC-FFP法

計算機の利用が進み、いわゆる埋込み（組込み）系のソフトウェアの利用が広がるにつれ、この分野のソフトウェアの機能規模測定にも適した機能規模測定法が求められるようになった。これに応えて開発された手法が、COSMIC-FFPである。COSMIC-FFPのマニュアルは、ケベック大学のサイト［LRGL］や東京電機大学のサイト［ITLAB］から入手可能である（どちらからも日本語マニュアルが入手できる）。

COSMIC-FFPの詳細な説明がプロジェクトマネジメント学会誌の最新号に掲載された［NAGANO2005］のでそちらも参照されたい。

（1）ソフトウェアモデル

① ソフトウェアは階層されて構成される。

② 各階層のソフトウェアは、機能プロセスからなる。

③ 機能プロセスは、次の機能サブプロセスからなる。

a）エントリ（E）：主として他階層からの入力

b）エグジット（X）：主として他階層への出力

c）リード（R）：（自階層の）メモリからの読出し

d）ライト（W）：（自階層の）メモリへの書出し

COSMIC-FFPのソフトウェアモデルを図2に示す。COSMIC-FFPは、いわゆる従来概念のファイル（データベース）を意識しないモデルになっていること、メモリへの読み書きを導入していること等から、より組込み（埋込み）型ソフトウェアに適した手法であると言われている。

（2） 数値化手順

COSMIC-FFPの数値化手順は極めて単純である。測定対象ソフトウェアを上記のソフトウェアモデルによってモデル化し、得られた機能サブプロセスの数の合計をCOSMIC-FFPの値とする。例えば、Eが10個、Xが10個、Rが20個、Wが30個得られたときは、70　CFSUとなる。CFSUは、COSMIC-FFPの単位である。

3.4 NESMA法

NESMA法は、IFPUG法とNESMA概算法（以下、概算法）及びNESMA試算法（以下、試算法）から構成される。ここでは、概算法と試算法について述べる。なお、両手法のソフトウェアモデルは、IFPUG法と同じと考えてよい。NESMA法のマニュアルは、NESMAから入手できる（有料）。ただし、概算法、試算法についてはNESMAのサイトで紹介されている。

（1）概算法

IFPUG法では、抽出されたBFCを専用の表を使って評価し、数値を割り当てる。しかし、概算法では、この数値をBFCの種別ごとに1つの値に固定し、数値化手順を簡略化している。NESMA法では、この手法で得られる数値を予測値（Estimated）と呼んでおり、これにより得られる値は、IFPUG法で得られる値を開発のより早期の段階で予測するものであるとしている。しかし、概算法は機能規模測定の概念にも適合していること、他にも類似の手法があること等から、筆者は、IFPUG法によるファ

データ項目数 ＼ 参照ファイル数	0～1	2	3以上
1～4	低		
5～15		中	
16以上			高

低→3
中→4
高→6

表2　EIの評価表

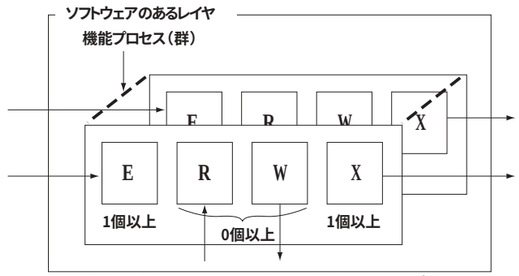


図2　COSMIC-FFPのソフトウェアモデル

ンクションポイントを求める工数を削減する、1つの独立した機能規模測定法であると考えている。

(2) 試算法

試算法では、次式により大まかなFPを求める。

試算FP＝35×ILFの数＋15×EIFの数

この手法は、ソフトウェア開発ではファイル（あるいはDB）が開発のかなり初期の段階で決定されることに着目したものである。工程が進んで、IFPUG法が測定できるようになる以前に、大まかに推定するときに用いることができる。

3.5 MkII FPA

開発者のC.Symonsによれば、Albrechtの手法をもとに開発されたもので、工数や期間と相関の高い値が得られる手法である［SYMONS1988］。日本での利用例はあまり聞かないが、英国で普及しているようである。

(1) ソフトウェアモデル

MkII FPAのソフトウェアモデルは単純で明快である。つまり、ソフトウェアは、入力、処理、出力から構成される論理トランザクションからなる。図3にこれを示す。この手法が最適なソフトウェアも事務処理系と考えられる。このソフトウェアモデルが、COSMIC-FFPのソフトウェアモデルのもとになっていると筆者は考えている。

(2) 数値化手順

数値化は、情報処理規模と技術的複雑さによる調整からなる。

情報処理規模（UFP）：

① 各論理トランザクションを識別し、入力データ要素数、出力データ要素数及び処理で使用するエンティティの数を識別する。

② 全論理トランザクションに対して得られた、入力データ要素数、出力データ要素数及び処理で使用するエンティティの数をそれぞれで合計する。

③ 次の式を用いて、情報処理規模（UFPと呼ぶ）を求める

UFP＝0.56×（全入力データ要素数）
＋0.26×（全出力データ要素数）
＋1.66×（全エンティティ数）

技術的複雑さの調整：

19個の既定の項目とアプリケーションごとに追加する項目を0～5点評価する。

MkII　FFPのファンクションポイント：

次の式で求める。

FP＝UFP×(0.65＋0.005×技術的複雑さの調整の合計)

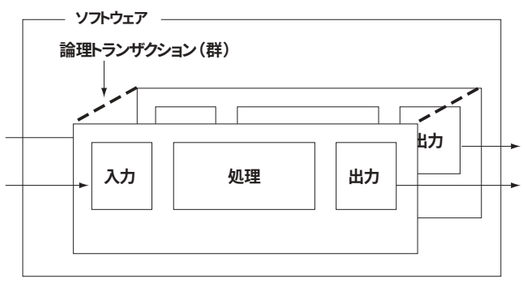


図3　MkII FFPのソフトウェアモデル

3.6 その他の手法

世界中で様々な手法が考案され、利用されているが、ここでは、ファンクションポイント法の権威であるC. Jonesが設立したSPR社の手法について述べる。SPR社の手法は、文献［JONES1993］のpp.78－81（SPRファンクションポイント法）、pp.91－105（SPRフィーチャポイント法）で紹介されている。SPR社の手法はSPR社の財産的情報とされており、手法の詳細は同社に問い合わせる必要がある［SPR］。

A. SPRファンクションポイント法

SPRファンクションポイント法は、IFPUG法の簡略法といえる。

(1) ソフトウェアモデル

IFPUG法と同じ。

(2) 数値化手順

① 全てのBFCの値をIFPUG法の3段階のうちの中間の値とする。

② 問題の複雑さ、プログラムの複雑さ、データの複雑さを1～5点で評価し、これから表引きにより、複雑さの補正係数を求める。

③ 次の式によりファンクションポイントを求める。

FP＝BFC評価の合計値×複雑さの補正係数

B. SPRフィーチャポイント法

事務処理系のソフトウェア以外への適用を拡大するために、（プログラム）のアルゴリズムを評価に入れた手法である。アルゴリズムの評価は難しく、あまり成功していないのではないかと考えられる。また、ソフトウェアに何をさせたいかではなく、どのようにさせたいかを評価するため、ISO/IECで規定する機能規模測定法とは異なった概念であると考えられる。

3.7 各種手法の比較の整理

各手法が得意とするソフトウェア分野と測定の詳細度を表3に示す。

3.8 手法間の互換性

機能規模測定法の手法間の互換性は、いろいろ試されているが、ここでは次の2件を紹介する。

①MkII FPAの開発者であるC. SymonsがIFPUG法とMkII FPAを比較して、あまり互換性がないと述べている［SYMONS1991 pp.34 - 45］。

②IFPUGのWGがIFPUG法とCOSMIC-FFPとの間の互換性について調査を開始したが、途中で打ち切っている。

筆者は、互換性について、次のように考えている。なお、ISO/IEC 14143－1（JIS X 0135－1）では、互換性をConvertibility（変換性）と呼んでいる。

①ソフトウェアモデルが類似のもの：統計的手法により互換性を示すことが可能であり、そこに意味がある。

②ソフトウェアモデルが異なるもの：測定対象ソフトウェアの構造の捉え方が異なるため、互換性の考えはあまり意味がない。異なるものを測定しているので、測定値間に関連はない。

表3　各手法の比較

	詳細測定	簡略測定	予測
事務処理系	•IFPUG法 •NESMA法 •MkII FPA	•NESMA概算法 •SPR FP法※	•NESMA試算法
組込み系	•COSMIC-FFP	－	－

※ SPRファンクションポイント法

4. 機能規模の利用法

機能規模測定のベースとなる利用者機能要件には品質要件、技術要件を含まない。したがって、機能規模自体をいくら評価しても、品質、技術に関する情報は得られない。しかし、機能規模は利用者がソフトウェアにさせたいこと（機能）を定量化したもの（と期待されるもの）であり、同じ位置に並べるため、開発関連諸量を正規化する手段として利用できる。あるいは、関数のパラメータ等により開発関連諸量と関連付けて示すことに利用できる。以下に、例を示す。

(1) 開発注者との価格、納期交渉の基礎値

次のような利用が考えられる。

“○○FPのソフトだから、費用及び開発期間の見積り式から開発費△△円、開発期間□□月である”と提案する。

“●●FPの仕様増加だから、費用及び開発期間の見積式から▲▲円、開発期間■■月の増加となる”として交渉する。

(2) 開発管理

開発費、開発期間、所要工数見積りのパラメータ、進捗管理のパラメータ、試験密度やバグ密度の分母（正規化手段）等で利用して、管理する。

(3) 資産管理（ポートフォーリオ）

“全現所有ソフトは●●FP、今年度は□□FP開発するから、来年度の全資産は△△（●●＋□□）FP”などとして資産の多さを示す。

5. ファンクションポイント法推進団体

本節では、国内外の機能規模測定法の推進団体について述べる。

(1) 日本国内の団体

国内でファンクションポイント法の普及を行っている団体が1993年に設立された日本ファンクションポイントユーザ会（JFPUG）［JFPUG］である。任意の団体であるが、300名を超える会員を擁し、年4回の会合の他、測定法の問題検討、ファンクションポイントをメトリクスと

して使うときの課題検討、IFPUG法の講習会、マニュアル類の翻訳等、活発に活動を行っている。

また、現在のIFPUG法の普及団体から、国内のソフトウェアメトリクスの中心的団体に変身すべく、設立10年目の昨年VISION2010を策定し、目標に向けて活動を行っている。

IFPUGには、ファンクションポイント法の測定能力を認定するCFPS（Certified Function Point Specialist）という資格認定制度がある。2005年度までこの試験を国内でも行っていたが、受験は英語であった。2006年度からは、日本語で行えるように作業を行っている。これも、JFPUG活動の大きな成果である。

(2) 外国の団体

機能規模測定に関する団体で世界一は、世界最初であることも含めIFPUGである。ただし、メンバ数を明示せず、メンバ資格も複雑であり、正確な数字は不明であるが、600メンバくらいであろうと思われる。

IFPUGを除くと、JFPUGが世界最大の機能規模測定団体であろうと筆者は推測している。

機能規模あるいはメトリクス関連の団体は、世界中にある。これを表4に示す（ただし、IFPUGとJFPUGは除く）。多くの団体が名称の一部に“FPUG”（Function Point Users Group）を付ける代わりに“SMA”（Software Metrics Association）を付けるようになってきた。機能規模測定法を足がかりに、広くソフトウェアメトリクスの利用法を促進しようとする傾向がうかがえる。

近年、ブラジルと韓国でファンクションポイント法の利用が進んでいると言われている（ブラジルはCFPSの人数が世界一、韓国は国家のプロジェクトではIFPUG法の利用を指定）。

これらの中で異色な団体がISBSGである。オーストラリアに本拠を置くNPOで、世界中からソフトウェア開発に関するデータを収集し、これを分析した結果を公表（有料）している団体である。また、データを送ると、ベンチマークして返却してくれる（無償）。現在、収集したプロジェクト数は2000以上に上るとしている。

6. 標準化の動向

本節では、ISO/IEC及びJISの動向について述べる。

(1) ISO/IECの動向［ISO］

ISO/IECでは、次の6つの機能規模の定義に関連する規格の開発を進めてきた（英文表記は内容がわかる程度に省略）。

ISO/IEC 14143－1 Definition of Concepts
ISO/IEC 14143－2 Compliance Assessment
ISO/IEC TR 14143－3 Verification
ISO/IEC TR 14143－4 Reference Model
ISO/IEC TR 14143－5 Functional Domain
ISO/IEC 14143－6 FSM IS use Guide

以上のうち、ISO/IEC 14143－6を除き、全て出版済みである。ISO/IEC 14143－6は、日本が主体（提案及びEditor）になって進め、現在、規格化の最終段階にある。

具体的な手法に関して、次の国際規格が出版されている（英文表記は内容がわかる程度に省略）。

ISO/IEC 19761 COSMIC－FFP
ISO/IEC 20926 IFPUG Function Point Analysis
ISO/IEC 20968 UKSMA Function Point Analysis
ISO/IEC 24750 NESMA Function Point Analysis

これらの手法は、前述した。

(2) JISの動向［JISC］

機能規模測定の規格のJIS化は、国内事情を考慮しながら、ISO/IECの規格に対応して進められている。表5に対応関係とJIS化の状況を示す。なお、表中に“JIS化予定”とあるのは、日本規格協会情報技術標準化研究センター（INSTAC）のソフトウェア生産管理委員会（機能規模測定）の見通しであり、公式予定でない点に留意されたい。

7. おわりに

ソフトウェアの規模及び規模測定は、ソフトウェアを工学（技術）的に扱う各種手法の中で、それほど大きな位置を占めるものではないが、ソフトウェア開発に関する諸属性を分析する基礎となる重要なデータである。規模をきちんと把握することなくして、ソフトウェアの工学的扱いの次のステップに進めないと考える。また、その規模は計り易いものであり、メトリクスプログラムを開始する良い動機になると考える。

本稿が、機能規模測定の概念と状況の理解の助けとなれば幸いである。最後に、次の言葉を紹介したい。

“You cannot control what you cannot measure.”
―― 英Lord Kelvin、米Tom DeMarco

表4 各国の機能規模関連団体

団体略称	国	コンタクト情報
ASMA	オーストラリア	www.asma-sqa-nsw.org.au
BFPUG	ブラジル	—
CSPIU	中国（本土）	www.cspi.com.cn
FISMA	フィンランド	www.fisma.fi
DASMA	ドイツ	www.dasma.de
NASSCOM	インド	www.nasscom.org
GUFPI - ISMA	イタリア	www.gufpi.org
KFPUG	韓国	www.kfpug.co.kr
NESMA	オランダ	www.nesma.nl/sectie/home/
AEMES	スペイン	www.aemes.org
SASMA	南アフリカ	—
SWISMA	スイス	www.swisma.ch
UKSMA	イギリス	www.uksma.co.uk
ISBSG	オーストラリア	www.isbsg.org/

表5 ISO/IECの規格とJISの対応及びJIS化の予定

ISO/IECの規格番号	JISの番号と名称	JIS化の状況
ISO/IEC 14143-1	JIS X 0135-1 ソフトウェア測定－機能規模測定－第1部:概念の定義	出版済み
ISO/IEC 14143-2	JIS X 0135-2 ソフトウェア測定－機能規模測定 －第2部:ソフトウェア規模測定手法のJIS X 0135-1:1999への適合性評価	出版済み
ISO/IEC TR 14143-3	なし	未定
ISO/IEC TR 14143-4	TR X 0073-4 ソフトウェア測定－機能規模測定－第4部:参照モデル	出版済み
ISO/IEC TR 14143-5	なし	未定
ISO/IEC TR 14143-6	なし	今後JIS化予定
ISO/IEC 19761	作成中	作業中。2006年度出版予定
ISO/IEC 20926	作成中	作業中。2006年度出版予定
ISO/IEC 20968	なし	未定
ISO/IEC 24750	なし	試算法、概算法のみ今後JIS化予定

参考文献

［ALBRECHT1979］ Albrecht, A.J.：Measuring Application Development Productivity, Proc.Joint SHARE/GUIDE/IBM Application Development Symposium, pp83－92, 1979
［ALBRECHT1983］Albrecht, A.J.and Gaffney, J.Jr.：Software Function, Source Lines of Code and Development Effort Prediction： A Software Science Validation, IEEE Tr. on SE Vol. 9, No. 6, pp. 639－648,1983
［COSMIC］http://www.cosmicon.com/
［FP4.1.1］日本ファンクションポイントユーザ会監訳：ファンクションポイント測定マニュアル リリース 4.1.1
［IFPUG］http://www.ifpug.org/
［ISBSG］http://www.isbsg.org/
［ISO］http://isotc.iso.org/
［ITLAB］http://www.itlab.k.dendai.ac.jp/2000/form.htm
［JFPUG］http://www.jfpug.gr.jp/
［JISC］http://www.jisc.go.jp/
［JONES1993］C.Jones著,鶴保他監訳：ソフトウェアの定量化手法,共立出版,1993
［KEMERER1987］ Kemerer, C.F.：An Empirical Validation of Software Cost Estimation Models, CACM, Vol. 30, No. 5, pp. 416－429,1987
［LRGL］http://www.lrgl.uqam.ca/cosmic－ffp/manual.html
［NAGANO2005］長野伸一：機能規模測定法COSMIC－FFPの組込ソフトウェアにおける適用性,プロジェクトマネジメント学会誌,Vol.7,No.5,p3－8,2005
［NESMA］http://www.nesma.nl/
［NISHIYAMA1994］西山 茂, 古山 恒夫：ファンクションポイント法の有効性と適用性,第14回ソフトウェア生産における品質管理シンポジウム,B－5,1994
［NISHIYAMA1995］ 西山他：ファンクションポイント方の効率的適用に関する一考察,第15回ソフトウェア生産における品質管理シンポジウム,B－7,1995
［SPR］http://www.spr.com/
［SYMONS1988］Symons,C.R.：Function Point Analysis：Difficulties and Improvements, IEEE Trans. Software Eng., Vol.14 No.1, Jan. 1988
［SYMONS1991］C.R. Symons：Software Sizing and Estimating, John Wiley＆ Sons, 1991
［UKSMA］http://www.uksma.co.uk/

先進ソフトウェア開発プロジェクト Part Ⅱ

SEC所長補佐
松浦　清

SECエンタプライズ系プロジェクト　研究員
神谷芳樹

SECプロジェクト統括グループ　研究員
樋口　登

既報（プロジェクトレポート　先進ソフトウェア開発プロジェクト　樋口登）[HIGUCHI2005]の通り、SECの活動領域の1つである先進ソフトウェア開発プロジェクトでは、実際のソフトウェア開発を担当するCOSE（COnsortium for Software Engineering）（ソフトウェアエンジニアリング技術研究組合）とソフトウェアエンジニアリング手法の適用を担当するSECが緊密に連携し、プロジェクトを推進してきた。

本プロジェクトで、COSEは、自動車プローブ情報システムプラットフォームの開発を行っている。一方SECは、EASE（Empirical Approach to Software Engineering）プロジェクト[MITANI2005]の協力を得て、定量データ自動収集ツールEPM（Empirical Project Monitor）の導入と運用を中心に、プロジェクトに対しソフトウェアエンジニアリング的側面からの支援を行っている。

本稿では、SECとEASEが行った活動を中心に述べる。

① COSEによる開発の問題と対策

COSEは、自動車プローブ情報システムプラットフォームの第1次開発を終え、結合試験、総合試験を完了した段階である。今後、実証試験を行い、評価を行った後、第2次開発へと継続していく予定である。

ソフトウェアの開発では、COSEのプロジェクトマネージャのもとで、ベンダ5社が個別に開発し、それぞれ単体試験、社内結合試験を終えた後、各社分を合わせた社間結合試験、総合試験を実施した。この開発は、各社間がそれぞれブラックボックスの状態で行ったので、各社間はもとより、COSEのプロジェクトマネージャでも、ソースコード、開発工数、開発体制（外部委託先、要員数等を含む）に関しては知りえない状況であった。COSEプロジェクトマネージャが主催する進捗会議では、各社のプロジェクトマネージャから、各工程の進捗割合と予定日数とのずれが報告されるのみであった。COSEプロジェクトマネージャは、これらの報告に基づき、質疑を行い、プロジェクトの進捗を管理した。

このようなブラインドマネジメントを行わなければならなくなったCOSEプロジェクトマネージャを支援するため、定量データ自動収集ツールEPMが導入された。SECに毎週収集されたデータは、SEC/EASEの研究員によって分析され、それは適宜、COSEプロジェクトマネージャ、ベンダ各社にフィードバックされた。これまでも、EASEのEPMまたは同種の自動収集ツールは、実際のプロジェクトに何度も適用されてきた。しかし、データは刻一刻、自動で収集されていても、単に蓄積されるのみであり、その分析自体は、プロジェクト終了後に行われ、あれこれと後付けの説明がなされることが多かった。また、後追いのため、ある時点での異常値が見つかっても、その前後の事情（コンテキスト）がわからず、迷宮入りとなる事例も多かった。そこで今回は、未だ進行中の開発プロジェクトに対して、データ収集後、可及的速やかに分析結果のフィードバックを行った。また、異常値もその都度、必要なコンテキストデータを得て、説明が可能となるようにした。

② EPMによるプロジェクト計測

EPMは、Linuxおよびオープンソースのミドルウェア上で動作する自動収集ツールである。その構成は、バージョン管理ツールCVS、バグ管理ツールGNATS及びメールサーバからなっている。EPMのベータバージョンは、オープンソースソフトウェアとしてインターネット上に公開されている。今回は、個別に共同研究契約を結んで提供されている上位バージョンのアルファ版を用いた。

今回の先進プロジェクトではこのEPMに、さらに、まだ製品化されていない拡張分析機能を、またさらに、EASEプロジェクトで開発されているツールやSECの活動で収集されているベンチマーク・データベースをも活用して、多次元にプロジェクトを計測・分析し、フィードバックする試みを実践した。

次に、その概要を示し、全体の構成概要を図1に示す。具体的には、下記に示す6つの計測技術と方法を採用した。

2.1　EPMによる計測と分析

開発プロセスとプロダクトの基本情報は、EPMを用いて取得した。EPMは、バージョン管理ツールのCVS（Concurrent Versions System）[NONGNU]、バグ管理ツールのGNATS（GNats: A Tracking System）[GNU]、メーリングリスト管理から自動的に情報を取得する。そしてこれらからの情報をXML形式の標準データ形式に変換したのち、分析に供するためにRDBに格納する。

収集したデータからEPMのアナライザ機能は、次のような基本的な情報をビジュアルに表示する。それらは、プログラム行数の推移、累積障害件数、残留障害数の推移、障害平均残留時間の推移、社間メール件数の推移、これらとチェックイン、チェックアウトのタイミングとの関係であり、さらに、チェックアウト数の関係、障害件数とSRGM曲線の関係である。

上記EPMの分析結果から
製造工程の進捗実態
ソースコードの管理体制
障害解決力
等が見えてくる。

例えば、プロジェクトの進捗会議では、週次等で進捗状況の報告が行われるが、実際にCVSに登録されているソースコード量から、その報告が適切であるかが判断できる。また、CVSに登録されているソースコード量が日々増加している場合には、開発者が毎日コーディングした部分をCVSに登録していることが考えられ、CVSの

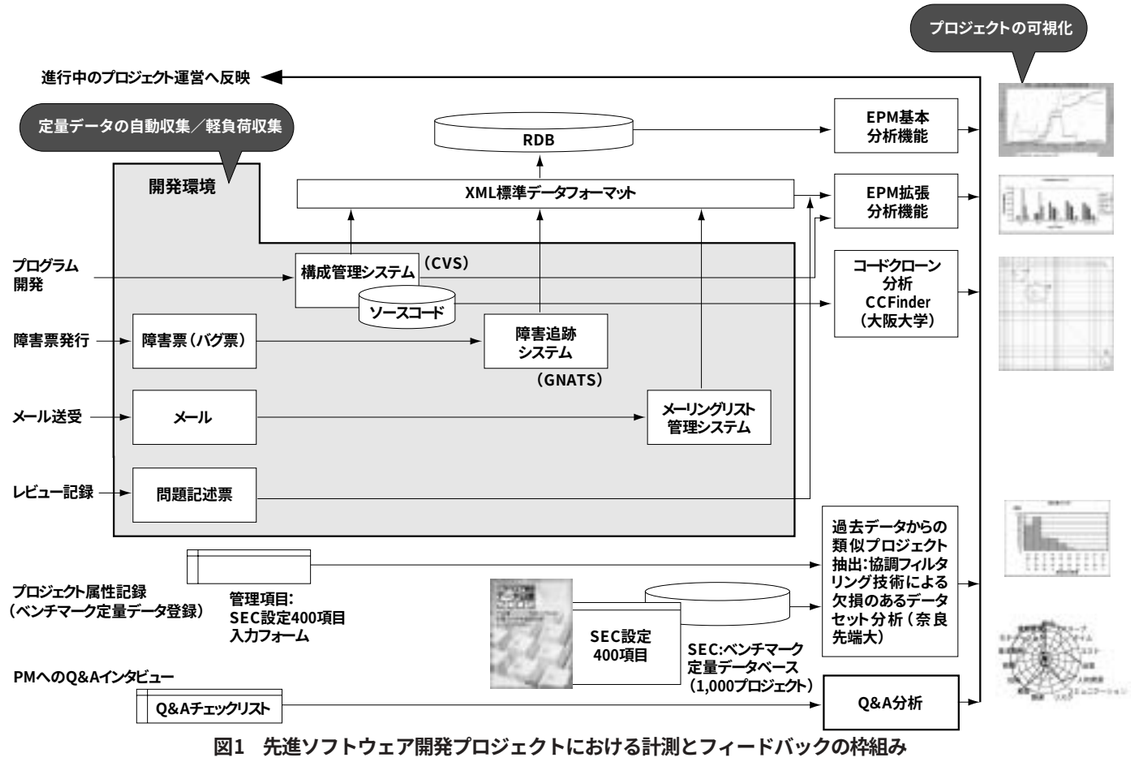


図1　先進ソフトウェア開発プロジェクトにおける計測とフィードバックの枠組み

運用が徹底されていると判断できる。この場合、レビュー前のソースコードが登録されていることになるため、登録されたソースコードが後で修正される可能性が高いと予測できる。もし、ソースコード量が毎日ではなく、週に一度というように、ある程度の間隔をあけて増加している場合には、モジュールやファイル単位等まとまった単位でCVSに登録していること、またレビュー後のソースコードをCVSに登録している可能性が高いことが考えられる。

障害に関する情報からは、新規の障害が登録されていないにもかかわらず残留障害数が減らないこと、あるいは障害平均残留時間が長くなる場合には、障害解決が進んでいないことが考えられる。したがって、その原因の究明及び、それを除去するための対策が必要であろう。

2.2 レビュー記録の収集と拡張EPMによる計測と分析

レビュー記録については、専用の電子フォームを用いて、基本設計と詳細設計に対し情報収集した。

レビュー記録の分析、ファイル更新に関する様々な分析等はEPMの新しい分析機能を用いて、CVSレポジトリの分析を試みた。

上記の分析結果から
障害解決コスト
各ソースファイル間の関係
開発担当者間の関係
障害の影響度
等が見えてくる。

例えば、ソースコードのCVSへの登録が常に同じ一人のメンバで行われている場合には、CVS登録専任者が置かれており、実際にコーディングを行っている開発者とCVS登録者が異なっていることが考えられる。

この場合、実際の開発者がソースコードを修正してからCVSに登録されるまでの間は、CVSに登録されているソースコードと開発者の手元にあるソースコードに差異が生じる。そのため、修正されたソースコードの登録漏れや登録の遅れ等が問題になり、連絡方法や管理体制に注意 する必要がある。

2.3 CCFinderによるコードクローン分析

コードクローンとは、ソースコード中の類似するコード片のことである。このクローンの分布状況、含有率等から、プロダクトの性質を推し量ることができる。今回ソースプログラムの提供を受け、EASEプロジェクトに参加する大阪大学のCCFinderというツール[CCFINDER]を用いて、ソースコードからコードクローンの含有状況、含有率の分析を実施した。ソースコードへのクローンの含有状況は、各種のクローン関係メトリックスと合わせて、スキャタープロットと呼ばれるダイヤグラムによって、全体を鳥瞰できるビジュアルな形で表示された。

上記の分析結果から
ソースコードの作成経緯
障害検出時のリスク
リファクタリングの必要性
等が見えてくる。

例えば、2つのファイル間のコードクローン率が非常に高い場合には、どちらかのファイルをコピーし、一部を修正して他方のファイルを作成したことが考えられる。あるコード部分が複数のファイルにまたがってコードクローンになっている場合には、その処理部分に障害が含まれていた場合の影響範囲が分散しているため、修正コストが大きくなり、修正漏れの可能性も高くなる。このような場合には、共通化等の対策を検討する必要がある。

2.4 ベンチマーク・データベースと協調フィルタリングツールの応用による分析

ベンチマークのデータは、SECが収集してきたデータ白書の400項目に沿った専用の入力フォームで収集した[IPA2005]。今回のプロジェクトでは、基本設計終了時に、全体のプロジェクト計画値と基本設計終了時までの実績値を集めた。プロジェクト終了時には残りの実績値を収集する。

集められたベンチマークデータは、EASEプロジェクトにおける奈良先端科学技術大学院大学（以下奈良先端大）の、欠損のあるデータセットから類似のデータをグルーピングする協調フィルタリングの技術を応用したツール[OHSUGI2006]を用いて分析した。これとSECが保有している約1,000プロジェクトの過去データを参照して、類似のプロジェクトグループを探し出し、プロジェクト途中でその将来を予測する試みを行った。

上記の分析結果から
製造工程以降の計画値の妥当性
プロジェクトの特性
等が見えてくる。

例えば、SECが保有するデータ内で類似度が高いと判断されたデータの製造工程以降の工数が予測値よりも大幅に多い場合には、予測値の妥当性について検討する必要がある。そのプロジェクト特有の事情によるのなら問題ない可能性もあるが、その特有の事情については把握しておく必要がある。

2.5 インタビュー質問用チェックリストの開発とリスク情報の収集

プロジェクトの管理体制等のリスク情報を収集するため、COSEの各企業を訪問した。そこで、企業内のプロジェクトマネージャや開発リーダに対し、SECの作成した80項目程度のチェックリストによるインタビュー調査を実施した。このインタビューに先立ち、30項目のチェックリストによる自己診断も行い、インタビュー結果と比較した。このチェックリストは、PMBOKの知識領域を参考にSECで作成した[IZAWA2005]。このインタビュー調査を通して、EPMのような自動収集ツールでは収集できないプロジェクトコンテキストに関する多くの情報を得ることができた。

このチェックリストは、SECのなかで、「プロジェクト見える化部会」の活動としてブラッシュアップがすすめられている。

2.6 会議継続参加によるコンテキスト情報の収集と分析への反映

さらに各種のコンテキストデータを得るために、計測グループのメンバの一部が、プロジェクトを通してプロジェクト会議に参加し、自動収集では得られない情報の取得に努力した。

③ プロジェクト開始後の問題点

今回の開発では、EPMの導入は事前に決まっており、ベンダ各社とその外部委託先の数が限られていたので、導入自体はスムーズに行われた。既に自社ツールを使われているところは、そのツールによる出力フォーマットから、EPMフォーマットへのコンバートを依頼した。コンバートの際にも、項目の追加や定義の微妙な変更があったが、極力他社と同じになるようにした。

(1) データ精度

しかし、事前に入力する項目や、自動収集の項目の定義については、決定に時間を要し、またそれを実際の開発要員に周知させることがなかなかできなかった。とくに、出来高など生産性に直結するデータは、その数字をインセンティブ（支払い金額と関連付ける等）とすると、必ず人為的操作、調整が入り、正確なデータ収集の妨げになることから、発注部門の閲覧を禁じた。これにより人為的操作はなくなったものの、逆に、データ収集へのインセンティブが失われ、漫然と入力が行われ、データの精度が問われる可能性があった。例えば、LOC（コードライン数）を行単位で入力すべきところを1,000行単位と間違えて入力してしまったケースがあった。

(2) 入力項目の定義

これまでは企業のプロジェクトにEPMを導入した場合、導入企業が入力項目の種類や定義を決め、所与のデータをいただくしかなかった。今回はほぼ強制的にEPMの適用を依頼したので、SEC/EASE側からは、どのようなデータも原則としては収集できることになった。

しかし、個別の入力項目については、入力データの必要性、すなわちデータの意味合い、つまりデータに対して、どのような仮説を持ち、分析し、結果を想定しているかについての検討が必要である。そのため、個別の項目の検討をCOSE、SEC、EASEで行った。とくに、収集コストの高い（収集に手間がかかる）項目は、収集の目的が厳しく問われた。

その結果、収集に手間がかかり、収集目的が明確でな

いものは次々と落とされていった。また、目的や分析例から、逆に項目の定義を修正したこともあった。意味合いは、漠然とわかるものの、定義となると各社各様となるもの（例えば、バグ率）は、厳密に何を何で割るのかという、分母子を明確に定義する必要があった。逆に、漠然とした意味合いのまま、プロジェクトマネージャの主観的評価（だいたい40%進捗した等）に頼らざるを得ないデータもあった。これらの収集項目の調整には、予想外の工数を必要とした。

④ ソフトウェア開発時の問題点

EPMデータの入力については、事前に周知したにもかかわらず、入力違い、EPMの操作違い等が起こり、ヘルプデスクの必要性を感じた。また、EPMは、自動収集は行うものの分析は手作業であり、基本的なグラフの表示と印刷だけでも時間がかかった。毎週の収集締め切り日と分析サイクルを定義し、フィードバックの日程は（祝祭日の回避、お盆休みに配慮等）事前に押さえたが、なかなか思うように実施できなかった。COSEのプロジェクトマネージャへの支援を考え、各社別のフィードバックより、COSEプロジェクトマネージャへのフィードバックの日程を優先したが、完全なリアルタイムフィードバックとはなり得なかった。むしろ、経験豊かなプロジェクトマネージャの勘を追証することになるケースも多かった。

フィードバック自体も、資料を用意し、プレゼンテーションをCOSEのプロジェクトリーダと各社の代表に個別に行う必要から、SEC/EASE側の作業時間をとられた。一方、1週間前のデータの分析では既に鮮度が落ち、フィードバックミーティングの対象（各社のプロジェクトリーダとするか、外部委託先のリーダとするか等）とタイミングの調整に苦労した。フィードバックミーティングで議論されたことも、文書化し整理して初めて役に立つが、分析とフィードバックに追われなかなかまとめる時間がなかった。

フィードバックの内容については、コードの進捗やバグの推移のような比較的地味な収集と分析より、協調フィルタリングやコードクローンのような新しい分析が好評であったように思えた。

⑤ 今後の課題

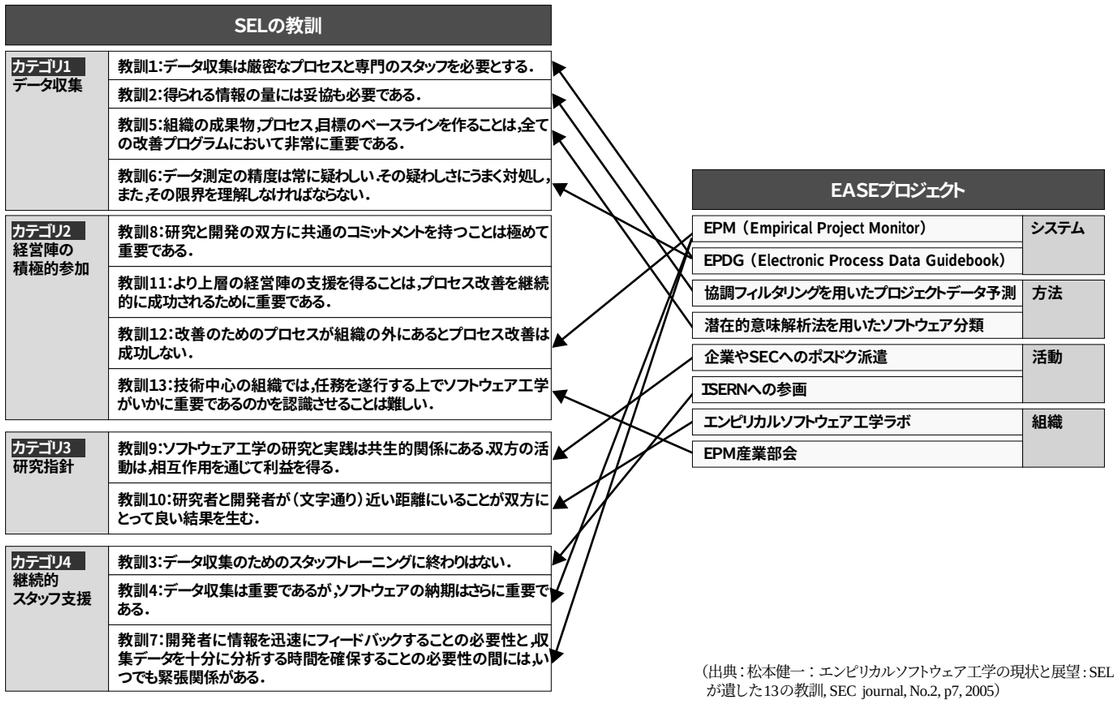
EPMの導入により、収集は自動化された。しかしその後の分析、フィードバックはマニュアル（手作業）であり、リアルタイムフィードバックは、標榜通りには実現できなかった。今後は、定番の分析の自動化と分析結果の着眼点、簡単な読み方の整理と事前配布の必要があると思われる。ともすれば遅れがちな開発の進捗とフィードバックのミーティングに時間を割くことは対立関係にあり、自動データ収集だけでなく、自動で簡単な診断書まで出てくるような体制があれば、ミーティング時間の短縮、もしくは省略が可能となろう。

今回は外部委託先を含めても、プロジェクト参加企業は十数社であり、EPMのインストールは簡単に行われた。しかし今後、大規模プロジェクトに適用したり、業界で広く利用されるためには、インストーラ等が必要となる。また、インストールサポートや初期運用サポート等のサービスも必要となろう。

入力項目の（定義の）見直しは、データの蓄積や蓄積データ間の比較のためには極力避けたいが、項目の種類と定義を標準的なものに収束させるためには、適宜実施していく必要がある。その際、各社間の定義の共通化が課題となる。例えば、A社では、今回のプロジェクトでは、重大バグがほとんど報告されなかった。その背景には、原子力発電所、金融機関の勘定系等を重要システムと位置づけており、また、重要システムのバグは、それ自体は軽微なものでも重大バグとなるよう、開発対象のソフトウェアの重要度によって、バグの重要度が変わる管理をしていたという事情があった。COSEのソフトウェアの中では、重大バグであっても、A社にとっては、重大バグとは認識されないことがあった。

今回のプロジェクトのように、プロジェクトマネージャや開発各社の間に多数のブラックボックスが存在する開発形態は例外的なものと思われた。しかし、外部委託先の細分化や、インド、中国等でのオフショア開発管理を考えると、今後は、このような開発形態が増えていく傾向にあると思われる。そのため、EPMの普及とそれによって得られたデータの蓄積は、ますます重要になってくるであろう。

本文を書くにあたり、上述の樋口のレポートを見るために『SEC journal』No.2を再見した。その中に、「エンピリカルソフトウェア工学の現状と展望：SELが遺した13の教訓」という奈良先端科学技術大学院大学の松本健一教授の招待論文が掲載されている。図2は同論文における＜SELの教訓とEASEプロジェクト＞の図である[MATSUMOTO2005]。今こうしてプロジェクトを経験して、再度この論文を見ると、なかなか含蓄が深いものがある。例えば、“ 教訓6：データ測定の精度は常に疑わしい.その疑わしさにうまく対処し,また,その限界を理解しなければならない. ” ということである。



（出典：松本健一：エンピリカルソフトウェア工学の現状と展望：SELが遺した13の教訓, SEC journal, No.2, p7, 2005）

図2　SELの教訓とEASEプロジェクト

謝辞

今回は、実際に動いているプロジェクトに対して、エンピリカルな分析のフィードバックができた。分析とフィードバックをご担当いただいたEASEの松村知子研究員、大阪大学の肥後芳樹氏、吉田則裕氏他の皆様、忙しい開発の合間を縫ってフィードバックに参加してくださったベンダ各社、そして、貴重な助言を賜ったプロジェクトマネージャの勝又敏次氏（NTTデータ）に御礼を申し上げる。

※本記事のPart Iは『SEC jounal No.2』（先進ソフトウェア開発プロジェクト）に掲載されているので、合わせてご覧いただきたい。

参考文献

[CCFINDER] http://www.ccfinder.net/index-j.html
[GNU] http://www.gnu.org/software/gnats/
[HIGUCHI2005] 樋口登：先進ソフトウェア開発プロジェクト, SEC journal,No2,2005
[IPA2005] IPA/SEC：ソフトウェア開発データ白書2005,日経BP社,2005
[IZAWA2005] 井沢澄雄：PMBOKの概要,SEC journal ,No.2,2005
[MATSUMOTO2005] 松本健一：エンピリカルソフトウェア工学の現状と展望：SELが遺した13の教訓 , SEC journal,No.2,2005
[MITANI2005] 神谷芳樹：組織紹介,EASEプロジェクト,SEC journal,No.1,2005
[NONGNU] http://www.nongnu.org/cvs/
[OHSUGI2006]大杉直樹：企業横断的収集データに基づくソフトウェア開発プロジェクトの工数見積り, SEC journal,No.5,2006

岐阜県の情報産業振興の取り組み

The efforts of Gifu Prefectural Government to promote regional information industry

http://www.pref.gifu.lg.jp/

岐阜県 産業労働部参事兼情報産業課長
宇野　秀雄

岐阜県では、県南部地域にIT産業の集積を目指す「スイートバレー構想」の下、まだITという言葉が一般的ではなかった頃から「ソフトピアジャパン」や「テクノプラザ」等の中核拠点を整備し、IT産業振興を強力に推進してきた。その結果、現在では大手企業からベンチャー企業まで多くの企業が集積する一大情報拠点に成長した。

また、21世紀の情報社会の最大の社会資本は人材であり、人の育成こそが地方における産業振興の鍵であると考え、主にITエンジニアを対象とした重点的な研修を実施している。

本稿では、こうした岐阜県の取り組みを紹介したい。

1 県内情報サービス業の状況

特定サービス産業実態調査（経済産業省）によると、平成16年の岐阜県内における情報サービス業は、事業所数200事業所、従業者数2,170名となっている。過去のデータの推移を見ると、全国平均と比べて岐阜県の事業所数の伸びはとくに大きいが、年間売上高に直結していない。これは、ベンチャー等売上規模の小さな企業が調査母数の多くを占めていることによると考えられる。その一方で、県内IT企業の売上上位を占める企業については、堅調に売上げを伸ばしており、ここ10年程でIT産業が本県の地場に根付いてきたことを示している。

2 スイートバレー構想

2.1 考え方

岐阜県では、交通、人材、通信インフラ等、ビジネスの活動拠点としての優位性をさらに活用するため、情報通信・ロボット分野の研究開発拠点、ハイテク産業、教育機関等が集積している県南部の木曽三川流域を「スイートバレー」と位置づけ、その資源を結集して、IT関連企業、コンテンツビジネス、ロボット産業等、世界有数の先端技術産業集積地の形成に取り組んでいる。

この「スイートバレー構想」を推進することにより、高度なIT関連産業のスイートバレーへの集積が一層加速され、県内産業の競争力・成長力の向上による経済活性化、雇用の確保が期待できると考えている。

具体的には、次のような意義が考えられる。

① ソフトピアジャパン・プロジェクトを中心とした、ITを重視した産業政策を推進する中で、地方における情報産業の育成のモデルを作り上げることができる。

② 本地域の伝統的な強みとなっているモノづくり産業の振興とIT施策を進める中で、日本のモノづくり産業の新しい姿、可能性を創造することができる。

③ ITの産業面への活用のみならず、地域づくり全般にもIT化を進めることにより、21世紀の地域社会づくりを進めることができる。

今後は、人材資源立地による産業の育成・誘致を目指すと共に、製造業との連携、地域経済の活性化、雇用の確保、そしてすべての県民がITにより豊かな生活を実感できる社会の実現を目指していきたいと考えている。

2.2 スイートバレー構想推進のための拠点及び体制

県内のIT産業振興の中核拠点として、平成8年にIT戦略拠点「ソフトピアジャパン」（大垣市）を、平成10年にはITとモノづくりの融合拠点「テクノプラザ」（各務原市）を整備し、現在225社の企業が集積している。県ではこの2つの拠点を中心に、情報産業の育成と産業の情報化を推進している。

スイートバレー構想における具体的な取り組み内容は以下のとおりである。

（1）ソフトピアジャパン・プロジェクト（大垣市）

IT関連産業の集積促進と、産業競争力の強化を目的に、国際的ITリサーチパーク「ソフトピアジャパン」を核としたインフラ整備及び各種事業展開を図る、大型の産学官連携プロジェクトである。

（2）テクノプラザ・プロジェクト（各務原市）

「テクノプラザ」は、岐阜県科学技術振興センターを中心に、ロボット技術やVR技術等、科学技術に関する各種研究開発機能が集積する研究開発拠点である。岐阜県では「テクノプラザ」を、21世紀型モノづくりの拠点と位置づけ、「ITとモノづくりの融合」による産業の高度化・情報化及び新産業の創出を推進している。

「テクノプラザ」では上記目的達成のため、①研究開発拠点づくり、②交流基盤づくり、③人材育成基盤づくり、④ニュービジネス創出拠点づくりの4つのコンセプトを柱に、各種施策を積極的に展開している。

平成17年12月時点で、61社、580名がテクノプラザ内に勤務している。61社の内訳としては大手・県外企業等9社、地元企業13社、ベンチャー企業24社、県及びベンチャー支援機関等15社。また平成17年10月にはテクノプラザに隣接している「テクノプラザ2期分譲地」を整備し、分譲地への進出も受け付けている。

（3）岐阜県情報関連業務戦略的アウトソーシング事業

県の情報システムの合理化と高度化及び県内情報関連産業の振興を図ることを目的に、平成13年度から平成19年度末までの7年間の予定で、情報関連業務のアウトソーシングを行っている。委託先は、総合評価一般競争入札方式（入札参加事業者からの提案について、内容点、価格点の両面から総合的に評価し、事業者を決定する方法）によりNTTコミュニケーションズ株式会社に決定。契約金額は約129億4千万円で、その内訳は情報システム関係分82.6%、情報産業振興関係分17.4%となっている。

3 具体的施策

3.1 人材育成

地方における情報産業の育成を推進するにあたっては、

岐阜県紹介

岐阜県は国土の中央部に位置し、日本の人口重心（一人ひとりが同じ重さを持つとしたときに日本全体の人口を一点で支える点）を県内に有している。面積は約1万596平方キロメートル（全国第7位）、人口は約211万人（全国第18位）で、7つの県に囲まれた数少ない内陸県の1つである。また、新幹線で東京まで2時間、大阪まで1時間で行くことができる等、鉄道や高速道路等の交通インフラの整備も進んでおり、平成11年には首都機能移転先候補地の1つとして選ばれている。

岐阜県北部は飛騨地域と呼ばれ、御嶽山、乗鞍岳、奥穂高岳等、標高3,000mを超える山々が連なっている。一方、南部の美濃地域は濃尾平野に木曽三川（木曽川、長良川、揖斐川）が流れ、特に長良川中流域は「日本の名水百選」に選ばれるほど美しい清流である。このように、自然に恵まれている岐阜県は、古くから「飛騨の山、美濃の水」という意味で「飛山濃水」の地と呼ばれてきた。

また、岐阜県では、飛騨の匠や関の刃物に代表されるように古くからモノづくりが盛んで、現在においても工業は岐阜県の中心的な産業となっている。全産業の事業所のうち工業の割合は約17%（平成16年事業所・企業統計調査）と、全国で最も高くなっており、ファッション、陶磁器、家具・木工、刃物、紙、プラスチック、食品等の特色ある地場産業がある。

首都圏のような大きな市場を見込めないことから、「人材資源による立地」を進める必要がある。優れたIT技術者を育成・確保することにより、質の高い仕事を提供できる企業を育て、首都圏からの大きな仕事も取れるような体力を地元企業に備えてもらおうということである。

具体的な人材育成のための研修は、IT業界における人材ロードマップを念頭に、スキルアップできるメニューを用意している。まずは未就職者を即戦力に育てる雇用直結型研修、次に就職後のスキルアップをサポートする現有戦力強化研修（全国マルチメディア専門研修センター研修、アネックス・テクノ2研修）が二本柱であり、その他、未来戦力育成研修（高校生向け研修）や近年、企業内での必要性が急速に高まってきた情報セキュリティ人材育成研修がある。以下にそれぞれの概要を記す。

（1）雇用直結型IT人材養成事業（表1、表2）

現在中部圏は、主産業である製造業が活気づき、IT技術者の人材需要も増えている。また、最近の企業動向をみると、首都圏における人材・通信・電力環境等の飽和状態を懸念し、また、危機管理上の理由から拠点の一極集中を解消して地方拠点を求める展開がみられるようになった。コールセンター業界はいち早く動き、昨年、岐阜市にも東京海上日動保険等2社が進出を決めている。

県ではこうした企業の人材需要に対応し、企業誘致と既存産業の強化を図るため、昨年2月より雇用直結型IT人材養成事業を始めた。フリーター等を対象に無料の短期集中研修を行い、SEやコールオペレーター等として地元企業への就職を実現しており、現時点で多くの就職者を輩出している。

また今後の展開としては、IT技術者の職種の中では高度な部類に属するSEや、エンベデッド技術者（プログラマー）等の既設コースに加え、さまざまなIT職種に対応した新コースを幅広く設置することにより、育成する人材のすそ野が広がり、産業振興と雇用創出の両輪を充実させることができると考えている。

表１　雇用直結型IT人材養成事業の概要

主　催	岐阜県、財団法人ソフトピアジャパン
場　所	ソフトピアジャパン
目　的	IT企業の成長の要であるIT人材を中期的視点に立って育成しIT関連産業の振興を図ると共に雇用創出及び人材需給ミスマッチの効果的な解消を図るため、雇用に直結する無料IT研修を実施する
目　標	IT技術者450名を育成（平成18～20年度）
効　果	①地元IT関連産業の振興（県税収入の増） ②雇用の創出・人材需給ミスマッチの解消（フリーター、リストラ等への対応） ③若い人が住んで働くことのできる地域づくり（少子化対策）

表２　　雇用直結型IT人材養成事業　現在実施中の研修コース

SE養成コース　平成16年2月～	
目　的	SE（システムエンジニア）の育成
期　間	標準6ヶ月
実　績	就職40名（内定者含む） （応募519名　受講48名　修了40名）
スペシャリスト養成コース（エンベデッド）平成17年2月～	
目　的	ITスペシャリスト（プログラマー、オペレーター等）の育成を目指すコースで、人材需要に応じた柔軟なカリキュラムを用意する。 現在はエンベデッド（組込み）技術者向けを開講
期　間	標準6ヶ月
実　績	就職13名（内定者含む） （応募65名　受講29名　修了14名）
コールセンター専門コース　平成17年2月～	
目　的	コールオペレーターの育成
期　間	2週間
実　績	就職68名 （受講289名　修了203名）

（2）現有戦力強化研修

地元IT企業の競争力強化には、企業の第一線で活躍している技術者のスキルアップが必須であるため、企業のさまざまな研修ニーズに応じた多様なカリキュラムを有料で提供している。

○全国マルチメディア専門研修センター
　①CIO（情報担当役員）スクール
　②IT企業向け研修
　③一般企業向け研修
　④県民向け研修
　⑤IAMAS公開講座

○アネックス・テクノ2研修事業＜主にテクノプラザで開催＞

モノづくりの高度な企業技術者の育成、地域産業の高度化に資することを目的とした2次元・3次元CAD研修を開催し、基礎から応用レベルまでの企業ニーズに応じた研修を実施している。

①2D／3D　CAD研修
②実技研修（試作加工等）

（3）情報通信セキュリティ人材育成センター研修事業

セキュリティ事故が多発している現在、総務省の推計によると、IT分野の人材は約42万人が不足しており、中でも企業が望む高度な知識と経験を有する情報セキュリティ分野に関する専門的な人材不足は、約12万人であるという。

このような中、平成16年12月に発行された中部経済連合会の『情報セキュリティに関する提言』の中で、「ソフトピアジャパンを活用した情報セキュリティ分野での人材育成ができるセンター創設の必要性」が言及され、これを受けて、岐阜県大垣市の財団法人ソフトピアジャパンでは、本年度総務省の補助を得て、情報通信セキュリティ人材育成センターを、平成18年4月のオープンを目途にセンタービル内に整備している。

本センターは、中部地区初の実践的・専門的研修施設で、企業内ネットワーク（イントラネット）とインターネット接続環境を仮想的に構築することで、典型的な企業ネットワークシステムを実現する。そこでは、従来型の予防、抑止等の対策に加え、実際にインターネットやイントラネット上で発生しうるいくつかのインシデントを発生させることで、受講者がインシデントに対応するための業務面・技術面双方におけるレスポンス手順を習得し、組織内での体系的セキュリティマネジメントの実践対応能力の向上を目指す。

また、企業がセキュリティ被害にあわないための対策だけでなく、被害にあったことを想定したIT防災訓練をどのように実践するかを研修する。企業内外に存在する脅威と脆弱性を分析する能力や、セキュリティ戦略を策定し、運用（実践）する手法を研修することで、個々の業務だけでなく企業システム全体をより幅広い視点で見渡し、インシデントの発生に対し、迅速に行動できる人材を育成する。

これにより、セキュリティ関連被害の防止及び迅速な対応が可能となり、個人情報等が守られ、誰もが安心・安

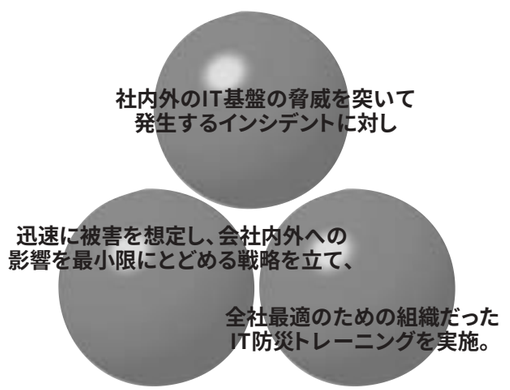


図1　情報通信セキュリティ人材育成センター研修のねらい

全･快適にネットワークを利用できる社会を実現すると共に、中部圏の製造業･IT関連企業のさらなる競争力向上及び高付加価値化へ貢献する。

（4）高校生向け研修事業

今日、ITに関する技能は、情報収集、決済、通信等を行うために必要不可欠な技能となりつつあり、読み書きと同様、早い段階で情報リテラシーを身に付けることが重要である。そこで、IT系企業への就職を想定した、高校生向けのITスキルアップ研修を無料で実施している。

○高校生を対象としたJava研修

インターネットが爆発的に普及し、サーバサイドのプログラミング言語であるJava言語のニーズが増加しているが、Javaの技術者は慢性的に不足している。そこで、県立実業高校生の求職活動を支援し、今後のITカリキュラムを検討するうえでの実践・実験的なモデルケースとして、インターネット社会の標準言語であるJavaプログラミング技術を学習する当研修を実施することとした。

・期　間：講義・演習10日間、事前Eラーニング10日間程度
・受講者：平成17年度受講者　35名（県下実業系高校生）
・特　徴：履修者に対してサンマイクロシステムズから認定証を交付

○高校生向けインターネットマスター養成研修

情報化が急速に進展する今日、インターネットに関する技能は、情報収集、決済、通信等を行うために必要不可欠な技能となりつつある。そこで、資格取得に向けた学習をすることにより、インターネットに関する知識を習得するだけでなく、自分の実力を客観的に把握することを目的として、当研修を実施することとした。

・研修方法：Eラーニング（基礎学習、検定用学習）
・受講者：平成17年度受講者　50名（県下高校生）
・特　徴：検定合格者に「ドットコムマスター」認定証を交付

○高校生向けCAD研修

工業高校において、設計・製図等で2次元CADを利用することはあったが、企業で実際に利用されている3次元CADが十分に整備されておらず、現場の高度な技術に触れる機会がなかった。

そこで、モノづくりの次代を担う工業系の高校生を対象に3次元CADに関する研修を実施することにより、企業が求める専門技術を有した、即戦力となりうる人材を育成し、県内製造関連企業への就業機会の向上と、県内産業の高度化及び育成振興を図るため当研修を実施することとした。

・期　間：講義・演習2日×5校
・受講者：平成17年度受講者　41名（県下工業系高校生）

3.2　企業誘致

前述の戦略的アウトソーシング事業で行っている企業誘致活動のうち、特にめざましい成果を上げた事例がヤフー株式会社、株式会社ブロードバンドタワーの誘致と株式会社サービスウェア・コーポレーションの事例である。以下にその誘致による効果の概要を記す。

■事例①：ヤフー株式会社、株式会社ブロードバンドタワーのソフトピアへの誘致

◆誘致効果1：税収増・投資増

ヤフーがソフトピアジャパン地区に事務所を構えることにより、税収増（年間約10億円）が期待できる。なお、総投資額は、300億円以上。

◆誘致効果2：地域連携ビジネスの進展

インターネット利活用により、ヤフー株式会社、株式会社ブロードバンドタワーと、地方自治体や地元のさまざまな業種の企業とが組んだ地域連携ビジネスが、今後著しく成長し、地域生活を大きく変えていくことが期待できる。

■事例②：株式会社サービスウェア・コーポレーションのソフトピアへの誘致

◆誘致効果1：雇用増

同社の進出により、コールセンターオペレーター等の雇用増が期待できる。また、未就職者の雇用増だけでなく、これまで金融業等で勤務し高い接客スキルを持つ主婦層等に働く場を提供するという意義も大きい。なお、新規雇用創出効果は約140名であった。

◆誘致効果2：人材の育成

コールセンターという職場があるため、優秀なオペレーターが育ち、さらに優秀な人材を求めてコールセンターが進出してくるという好循環が期待できる。

4 今後の取り組み

戦略的アウトソーシング事業を中核として、今後も引き続き、県情報システムの再開発を円滑に推進すると共に、産業振興分野においては、契約期間（平成19年度末まで）の満了時までに、「県内情報産業の売上100億円増、新規雇用創出1000人増」を目標に、とくにIT人材育成、製造業の競争力向上とIT産業の活性化等を重点項目として事業を推進する。

また、ソフトピアジャパンやテクノプラザを中部圏のIT人材育成拠点と位置づけ、中部圏の産業界に優れた人材を輩出、または再教育し、中部産業界の底上げに貢献するための施策を推進する予定である。

表3　ソフトピアジャパン・テクノプラザ集積状況

岐阜県情報産業課

○ソフトピアジャパン（就労者数　2,089人＋WS24フラッツ※1 87人＋メゾン東大垣※2 211人＋昼間交流人口※3（1日当たり）546人）

・進出企業等数　（平成17年7月1日調べ）

				進出形態				
				分譲地	技術開発室		インキュベートルーム	その他
					通常	アーリーステージ		
大手･県外企業	33	大手	11	4	7			
		海外系	1		1			0
		県外	21		21			
				4	29	0	0	0
地元企業	33			14	19	0	0	0
ベンチャー企業	83	海外系	2				2	
		県外	16			10	6	
		県内	65			30	35	
				0	0	40	43	0
ベンチャー支援機関	10	海外系	1					1
		県外	1		1			
		県内	8		1	3		4
				0	2	3	0	5
支援施設	5（すべて県内）							5
その他	0（すべて県内）							
合　計	164			18	50	43	43	10

○テクノプラザ（就労者数　580人＋昼間交流人口※3（1日当たり）120人）

・進出企業等数　（平成17年12月1日調べ）

				進出形態				
				分譲地	技術開発室	インキュベートルーム		その他
						アーリーステージ	通常	
大手･県外企業	9	大手	0					
		海外系	0					
		県外	9	5	4			
				5	4	0	0	0
地元企業	13			2	11	0	0	0
ベンチャー企業	24	海外系	0					
		県外	7			3	4	
		県内	17			2	15	
				0	0	5	19	0
ベンチャー支援機関	14	海外系	0					
		県外	1		1			
		県内	13		5			8
				0	6	0	0	0
支援施設	1（すべて県内）							1
合　計	61			7	21	5	19	9

※1、2　ソフトピアジャパン内の居住施設　　※3　1日あたりの施設利用者

国立情報学研究所　トップエスイー・プロジェクト
スーパーアーキテクト養成のための「サイエンスによる知的ものづくり教育」

http://www.topse.jp/wp/wpb.da3

国立情報学研究所　知能システム研究系　教授・研究主幹／
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
本位田　真一

ソフトウェアシステムに対する最先端の様々な開発支援ツールについて、実問題への適用ノウハウ、新しい問題ならびにツールに対応するための応用力を身に付けた、いわばスーパーアーキテクトを育成することが望まれている。そこで国立情報学研究所では、平成16年度から文部科学省/科学技術振興調整費により、スーパーアーキテクト養成のための「サイエンスによる知的ものづくり教育」をスタートさせている。

1　人材養成の必要性

近年ソフトウェア開発現場において、開発効率や品質向上を目的として、様々な開発支援ツールが導入されるようになってきている。しかし、同様な機能を持つ類似のツールが多数存在し、その取捨選択が容易ではない。また適切なツールを選択しても、その適用箇所の特定、そして実問題への適用も容易ではなく、導入の際の大きな障壁になっている。さらにソフトウェアシステムの開発では、開発対象はすぐに変化し、それに伴い、ソフトウェアサイエンスの成果として新たなツールが次々に登場する。これも障壁をより高くしている要因となっている。したがって、開発現場において様々な最先端のツールが活用されているとは言いがたい。

こうした課題の解決のためには、最先端の様々なツールの実問題への適用ノウハウや、新しい問題ならびにツールに対応するための応用力を身に付けた、いわばスーパーアーキテクトを育成することが望まれる。しかし、企業においてそのようなエンジニアの育成を行うのは容易ではなく、独立した教育機関の重要性が高いものと考えられる。

2　トップエスイー・プロジェクトの取り組み体制

このような社会からの要請にこたえるため、国立情報学研究所では、平成16年度から文部科学省/科学技術振興調整費により、スーパーアーキテクトの育成を行う「サイエンスによる知的ものづくり教育」をスタートさせた。

本プロジェクトでは、まず国立情報学研究所、産業技術総合研究所、企業、および大学との協力により、平成16年度から計15件の教材開発を行っている。現時点の参加企業は、NTTデータ、東芝、日本電気、日立製作所、富士通研究所、PFU、東芝情報システム、CSK、およびFeliCa Networksである。特に、教材の題材として、企業からは数年先に顕在化する実問題を持ち込んでもらっている。なお、個々の教材ごとにWGを構築し、各WGでは、ほぼ１年をかけて、最先端のツールの実問題への適用実験を繰り返しながら、教材を開発している。

3　トップエスイー・プロジェクトの活動

本プロジェクトでは受講生として、平成17年9月には1期生を、18年9月には2期生を、19年9月には3期生を採用し、最終的には平成20年度までに計50名のスーパーアーキテクトを輩出することを目指している。平成17年度は、企業の若手エンジニアを中心に、計19名を採用した。

本プロジェクトの特徴を図1に示す。

特徴の1つに、テーマとして「情報家電」を選定した点がある。このテーマにより、組込み系とエンタープライズ系両方の知識を習得できる。また、もともとわが国が強みとしている家電に対し、わが国の弱点であるネットワーク機能を活用したソフトウェアの技術を強化することにより、国際競争力の向上への寄与が期待できる。

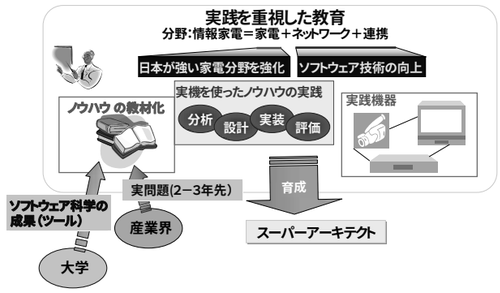


図1　トップエスイー・プロジェクトの特徴

本プロジェクトの講座では、各種開発支援ツールを実問題に適用する際のノウハウを習得させる。しかも本プロジェクトで取り上げる手法・ツールは、科学的基礎に基づいた、高度かつ先進的なものを採用した。また講座毎に、極力複数のツールを取り上げている。これにより、個別のツールの使い方だけでなく、手法・ツールの原理からの理解に基づいた、新たなツールへの応用力も身に付けさせる。

平成17年度に開講した講座は、「要求分析」「形式仕様記述（基本編）」「設計モデル検証」「コンポーネントベース開発」「アジャイル開発」の5講座である。平成18年度、19年度に、前年度に開発するそれぞれ5教材に対応する講座を開講し、最終的には計15講座から構成することとなる。1教材当たり2、3のツール・手法を習得する構成になっているので、最終的には30〜40のツール・手法を理解できることになる。例として、名称だけであるが、一部を列挙すると、SPIN、SMV、LTSA、KAOS、i*、UPPAAL、Java PathFinder、FDD、Catalysis、KobrA、B4free、Click'n'Prove、VDM-SLToolbox、ESC/Java等である。ちなみに、上記ツール・手法のうち、たとえば３種類以上を使いこなしているソフトウェア技術者は多くはない。

カリキュラムの構成は、今後の情報家電の課題を整理して「セキュリティ・安全性」「信頼性・性能」「柔軟性」を重視したものとした（表1）。上流工程の要求分析から下流工程の実装まで幅広い構成になっているが、複雑化するソフトウェア開発を克服するため、上流工程の強化を狙ったものとなっている。また、現状のソフトウェア

表1　カリキュラム

課題	セキュリティ・安全性	信頼性・性能	柔軟性
工程			
要求分析		要求分析:基本	
	安全要求分析		
	安全要求分析:検証	形式仕様記述:基本	
設計		コンポーネントベース開発	
		設計モデル検証	アジャイル開発
	形式仕様記述:セキュリティ	性能モデル検証	ソフトウェアパターン
		形式仕様記述:応用	
実装		プログラム分析	アスペクト指向設計・プログラミング
		テスティング	
		実装モデル検証	

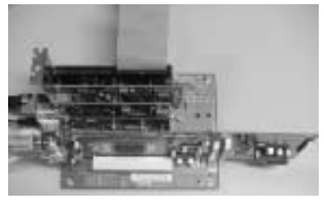


写真1　講座で使用する実機（評価ボード）

開発の実態に即してUMLやJava等オブジェクト指向をベースに、コンポーネントやアスペクト指向等、最近のトピックス等も取り入れている。

講座の流れは、全12回で、座学を最小限にとどめ、演習中心に、問題への取り組みにおいてツールを使い、グループ討議をしながら実機（写真1）を使って実践していくところがポイントである（図2）。最後に、最低８講座を取得した受講生を対象に、習得したツール全般を、各自毎の小規模の実問題に適用する、３ヶ月間の修了制作期間を用意している。

4　まとめ

本プロジェクトの講座を修了した技術者には、まさにスーパーアーキテクトとして、開発現場において技術面でのリーダシップを発揮し、わが国のソフトウェア開発を牽引することを期待している。

本プロジェクトで開発した実問題をベースとした教材は、全国の大学や企業に広く普及させる予定である。なお、教材の一部であるレクチャーノートは、今春から近代科学社より「サイエンスによる知的ものづくり」シリーズとして、15巻を順次刊行予定である。

なお、SECとは、教育プログラムの客観的な評価手法の構築に関する共同研究をスタートさせている。本プロジェクトに関する客観的評価もその一環として実施していく予定である。

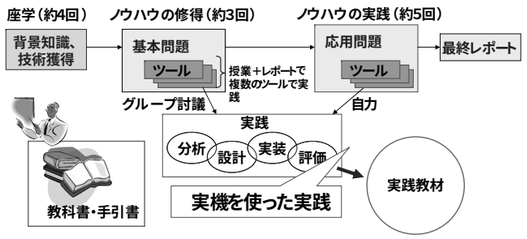


図2　講座の流れ

オブジェクト・マネジメント・グループ

http://www.omg.org/
http://www.otij.org/

オブジェクト・マネジメント・グループ（OMG）／オブジェクトテクノロジー研究所（OMG日本代表）
鎌田　博樹

1989年に設立された非営利法人OMGは、ミドルウェアやモデリング及びモデル駆動開発に関する標準を提供してきた。工学的なソフトウェア開発・管理に必要とされる標準を開発することでSECの活動とは密接な関係にある。日本における活動を含めて、OMGの標準とその目指すものを簡単に紹介する。

1　OMGの概要

OMGは1989年、技術的・空間的に分散するアプリケーションの協調運用のための標準を開発することを目的に設立されたソフトウェア標準化コンソーシアム（NPO）である。米国東海岸のボストン近郊のニーダムに本部があり、日･英・独・伯の４か国に代表を置いている。会員は全世界に600余りで、民間の標準化団体としては最大級。ソフトウェアベンダの他に、ユーザ、政府機関、大学・研究機関が参加している（日本からも20余りの企業・団体が参加）。

これまで分散ミドルウェアの標準プロトコルであるCORBA※1（1991～）、モデリングの標準表記法UML※2（1997～）をはじめ、それらを基盤とする標準体系（OMA※3-MDA※4）から、100以上の仕様を送り出してきた。一部はISOの国際標準としても採用されている。

OMGの最大の特徴は、設立当初から、有力なユーザ企業、政府機関がメンバとして参加していることだろう。金融、製造、医療、政府、航空宇宙等産業界のニーズを反映することで、OMGの活動は当初から、特定分野の技術標準の開発・普及というよりは「ユーザの問題解決」のために、アプローチを変えながら発展してきた。1990年代には主としてCORBAで知られてきたOMGが、最近ではUMLやMDA等で登場することが多いのは、そうしたユニークな性格によるといえよう。以下、16年間の標準化の流れを簡単に整理してみよう。

2　OMGの標準化の流れ

OMGが世に知られるようになったのは、CORBA 1.0（1991年）が最初である。これは異質な実装環境の間でのアプリケーション間通信を実現する、分散ミドルウェアのアーキテクチャとインタフェースを規定したもので、その拡張性の高さから、コンピュータばかりでなく、交換機や組込みシステムに応用され、大きな成功を収めた。大小多数のベンダから実装製品が提供されたが、それらはターゲットとする市場に対応した独自の特徴を持ちながら相互運用性を損なっていないという点で、標準化の成功例といえるものである。

1997年、OMGはUMLを採択したが、これはオブジェクト指向分析／設計を前提にしたシステム表記法であり、それまでの分散ミドルウェアから独立した初めての標準であった。UMLは、それまでの表記法の乱立を統一したもので、開発環境における相互運用性、モデルからコードへの変換といった、ソフトウェア工学の年来の課題の解決への歩みを加速した。

2000年、モデル表記法とミドルウェア標準という2系列の標準を統合し、それまでのミドルウェア・ベースの体系を転換することになるMDA（モデル駆動アーキテクチャ）が提案され、最終的に2001年3月に採択された。これはUML以外の非オブジェクト系のモデルも統合しうるMOFを基底的な仕様とし、特定プラットフォームに依存しないモデル（PIM）を、ミドルウェアも含めてプラットフォームを前提としたモデル（PSM）に変換し、最終的にコードに変換する、モデル駆動開発を前提としたアーキテクチャである。

UMLを使ったモデル駆動というアプローチは、金融決済（ISO20022）やeコマース（ebXML, UN/CEFACT）、地理情報システム（ISO19100）等様々な分野でも使われている。OMGはISO/IECをはじめとする公的な標準化機関、W3CやOASIS、UN/CEFACT、TMF、HL7等の標準化コンソーシアム、主要なオープンソースプロジェクトのほとんどと協調関係にあり、仕様の共通化や重複の回避、変換性の確保等に務めている。

3　OMGの組織形態と機構

OMGの標準技術はすべてメンバ企業によって提案、合意、仕様化されている。

OMGの機構は、Board of Directors（BoD、理事会）の下に3つの組織、Architecture Board（AB、アーキテクチャ評議会）、Platform Technology Committee（PTC、プラットフォーム技術委員会）、Domain Technology Committee（DTC、ドメイン技術委員会）が置かれ、それぞれ提案／仕様の整合性審査、プラットフォーム技術標準の策定、ドメイン技術の策定を担当している。これらの組織の下で、Task Force（TF、タスクフォース）、Special Interest Group（SIG）、Sub Committee（SC、小委員会）が、対象技術の性格によって具体的な技術を担当する。

PTCでは、分析設計、ミドルウェア、リアルタイム／組込み、レガシー移行等のTFが、DTCではビジネスモデリング、C4I（軍事）、金融、医療、製造、宇宙航空、生命科学、ソフトウェア無線、運輸、ロボット、電子政府、規制遵守等に関するTFやSIGが活動している。

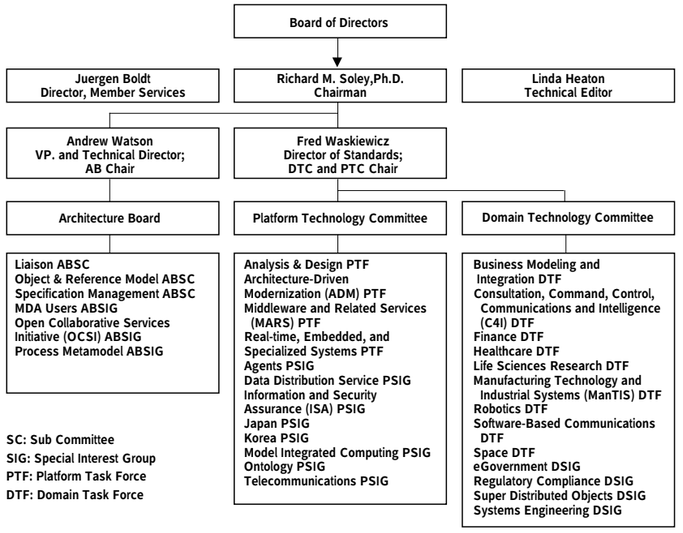


図1　組織図

4　日本におけるOMGの活動

日本におけるOMGの活動は古く、日本で開催されたTC Meeting（技術委員会総会）も、1990年以来5回を数える。1991年以降は日本代表を設置し、OMG技術の日本での普及、日本のOMGメンバへの支援、日本からのOMG活動への参加促進を行ってきた。現在は、2003年にOMGの日本法人（OMGジャパン、2000～2002年）から独立したオブジェクトテクノロジー研究所（代表・鎌田博樹、本社東京）が、OMG日本代表の責を負っている。

活動の1つはテクニカルカンファレンスの開催で、1992～96年に開催されたObject World Expoや、2001年以降開催しているUML Forum/Tokyo等の国際的イベントは、CORBAやUML、MDA等の技術や製品を、最初に日本に紹介する場としても機能してきた。2002年以来、年4回開催されるMDA Technology Forumシリーズは、ハイレベルのアーキテクトを対象に、新技術紹介と情報交換の場となっている。また2003年からは、日本が主導する形でUML技術者の国際的資格認定プログラム（OCUP）を発足させたが、これまで全世界で5,000名を超えるUML技術者を認定した。

OMGの標準化活動の重点は、すでに成熟期に入ったCORBAやUMLよりは、産業・用途別（ドメイン）の標準に移行している。OMGメンバの半数以上は、IT産業以外のユーザである。OMGは、ソフトウェアはITだけのものではないという認識を当初から持っており、過去10年にわたって、分野別標準に力を入れてきた。日本でも、ようやくそうしたOMGの役割が認知され始めており、バイオ、ロボット等での活動では、日本のメンバが大きな役割を果たしている。

標準技術を日本に普及させるだけでなく、日本のリーダーシップによる世界標準を送り出すことが、OMG日本代表としてのテーマであると考えている。

※1　CORBA：Common Object Request Broken Architecture
※2　UML：Untified Modeling Language
※3　OMA：Object Management Architecture
※4　MDA：Model-Driver Architecture

財団法人 日本科学技術連盟 SPC研究委員会

http://www.juse.or.jp/software/

財団法人 日本科学技術連盟　応用システム課
高橋　輝江

「SPC（ソフトウェア生産管理）研究委員会」は1980年にソフトウェアの品質の向上、効果的開発の方法論の確立と発展を目指して、財団法人 日本科学技術連盟に設置された委員会で、ソフトウェア品質に関する調査、研究、教育等、様々な活動を実施している。

1　SPC研究委員会とは

SPC研究委員会とは産官学の人材が集まり、1980年に財団法人 日本科学技術連盟内に設置された委員会である。現在、本研究委員会は、東京大学 教授 飯塚　悦功氏を委員長とし、下記の諸氏にご協力いただいている。

委員一覧　（敬称略・順不同）（2006年1月現在）

委員長	飯塚　悦功（東京大学）
副委員長	北島　義弘（株式会社CRCソリューションズ）
	榊原　彰（日本アイ・ビー・エム株式会社）
委　員	大野　晋（株式会社SKサポートサービス）
	大西　建児（株式会社豆蔵）
	岡崎　靖子（日本アイ・ビー・エム株式会社）
	小笠原　秀人（株式会社東芝）
	小井土　亨（株式会社OSK）
	鈴木　三紀夫（TIS株式会社）
	鷲見　研作（マイクロソフト株式会社）
	高橋　寿一（ソニー株式会社）
	西　康晴（電気通信大学）
	野中　誠（東洋大学）
	向井　清（住商情報システム株式会社）
	安井　昌男（清水建設株式会社）
	湯本　剛（株式会社豆蔵）

本委員会はこれまでに海外調査団（詳細後述）等の活動を通じて、日本のソフトウェア品質のレベルの高さを世界各国へ向けて発信してきている。

とくに、ASQ（米国）、EOQ（欧州）といった品質管理の推進団体と共に、ソフトウェア品質に関して世界各国が一堂に会する国際会議「世界ソフトウェア品質会議」（第1回（1995年）アメリカ、第2回:横浜、第3回:ドイツ）の開催団体の1つでもある。

SPCの特徴の1つとしては、日本のソフトウェア品質について、実学の切り口から技術の教育などを通して技術者の支援を行っていることが挙げられる。また、出版等を通じたソフトウェア開発技術の情報発信も行ってきた。

2　財団法人 日本科学技術連盟（日科技連）

財団法人 日本科学技術連盟は、1946年に設立され、1962年に科学技術庁（現文部科学省）所管の財団法人となり、今日に至っている。

創立以来、広い意味での科学技術の進歩と発展を図るために必要な諸活動を通じて、産業界に寄与することを基本方針として、これを実現するために調査、研究、開発、大会やシンポジウムの開催、教育訓練や国際交流の実施等の活動を通じて科学的な経営管理技術の普及、進歩、発展に努めている。とくに「品質」を中核とする経営管理技術に関する各種の事業は、国内はもとより世界各国から注目を集め、高い評価を得ている。

3　活動内容

長年にわたって蓄積してきたノウハウや人脈をベースに、人「財」育成と高品質なソフトウェアや情報サービス提供のためのマネジメント、エンジニアリング向上に資すべく調査、研究、教育を行っている。主な活動としては、セミナー、研究会、シンポジウム等の開催、海外調査団の派遣等を挙げることができる。

セミナーは、ソフトウェアを開発する際に必要となる基本的な開発技術や管理技術を、体系的、実践的に習得できる「技術者コース（6日間コース）」をはじめ、デザインレビューやテスト、プロジェクトマネジメント、PS

研究会－熱心な分科会風景

（Partner Satisfaction）といったテーマのセミナーを開催している。また、トピック的な内容のミニセミナーも不定期に開催している。

研究会は、ソフトウェア開発に関連した教育と事例研究を行うことを目的として、定例会を年8回開催している。定例会の午前は品質管理の基本的な考え方や諸手法、これからのソフトウェアに関する重要なテーマ、関心が高いテーマについて、各分野の専門家による特別講義を行っている（2005年度の特別講義の内容例：SWEBOK概要、要求分析、ファンクションポイント、設計パターン、形式検証、品質要求定義）。午後は、プロセス改善、プロジェクトマネジメント、テスト、ソフトウェア要求工学、ソフトウェア品質保証の基礎等、7つの分科会のいずれかで研究活動を行っている。年度の最終例会までに、分科会ごとに研究成果を論文としてまとめ、成果を発表し合っている（http://www.juse.or.jp/software/study_data2004.html）。

シンポジウムは、毎年9月に2日間の会期で開催している。2005年は、「ビジネスを成功に導くソフトウェア品質～ユーザ、ベンダ、組込み、エンタープライズの枠を越えて～」をテーマに開催した。4会場に分かれてのセッションでは、ソフトウェア品質に関わるホットなテーマ（人材育成、組込みソフトウェア、アジャイル開発、プロジェクトマネジメント、オフショア開発等の計16テーマ）に関して、一般からの投稿による論文発表、招待発表、各分野の専門家によるチュートリアル講演の計32

『クオリティ ワン』の表紙

件の講演や発表が行われた。

海外調査団は、原則年に1回の頻度で各国へ派遣している。2005年度は、9月に第3回世界ソフトウェア品質会議（The 3rd World Congress for Software Quality：3WCSQ）が開催されたドイツ・ミュンヘンへ派遣し、SECと共同研究を行っているドイツ・フラウンホーファー協会IESE（Institute of Experimentalles Software Engineering）を訪問した。また、11月には中国の上海、杭州へ調査団を派遣し、IT企業を視察したほか、杭州では日中ソフトウェア品質シンポジウムに参加し飯塚委員長が「ソフトウェア産業の競争優位要因について考える」をテーマに記念講演を行った。

以上の活動のほか、2005年には、SPC研究委員会初の試みとなるソフトウェア品質のための総合情報誌『クオリティ ワン』を発刊した。同誌では、経済産業省と飯塚委員長とのインタビュー「ソフトウェア産業の将来像は？」やIBMカナダのブラン・セリック氏（モデル駆動型開発はいかにしてソフトウェア品質を向上しうるのか）や豆蔵の羽生田栄一氏（ソフトウェア工学は何を目指すか）等、諸氏から寄稿をいただいている。

4　今後の活動

SPC研究委員会では、ソフトウェアエンジニアリングに関する資格制度や知識体系の整備等、ソフトウェアの品質向上の一助を担うべく、さらに幅広く活動を行っていくつもりである。

BOOK REVIEW

見える化　強い企業をつくる「見える」仕組み

遠藤　功 著

ISBN：4-492-53201-3　東洋経済新報社刊
四六判・200頁・定価1,680円（税込）　2005年10月刊

突きつけられる、ソフト開発現場の対極の姿

書店でこの目立つ表題が平積みになっていて、つい手を伸ばした。エンピリカルソフトウェア工学に親しんでいる者にとって「見える化」は今では旧来のテーマだ。どうせトヨタ系の人の自慢話だろう、と思って読み始めると、そうではない。「見える化」とは相当に奥深く、著者はこれを真剣に掘り下げている。前半に、「見える化」とは何か、「見える化」の体系、と続く。読み進んで、この本をマネすれば、「ソフト開発の見える化」という本が書けるかな、と思い始めるが、1/3ぐらいのところで、それはできない夢とわかる。本書では実に34もの実例が紹介されている。SECや連携するEASEプロジェクトが進めてきた大掛かりな枠組みの成果でも、広く紹介できる事例は数例に過ぎない。

著者は経営コンサルタントでありビジネススクールの教授だが、その基礎はトヨタでの施策のようである。この世界最大を窺う自動車メーカの現場は、日本のソフトウェア産業の現場の対極にあるように思える。締めくくりの章は『「良い見える化」を実現するために』というタイトルで、シンプルだが核心をついた指摘ばかりである。いつか、SEC/EASE発『ソフト開発の見える化』という本が書店に平積みになり、他産業の人がこれを手にする、そういう日を迎えたい。

（神谷芳樹）

ペアプログラミング　エンジニアとしての指南書

Laurie Williams・Robert Kessler 共著　長瀬嘉秀 監訳・今野　睦・テクノロジックアート 訳

ISBN：4-89471-699-2　ピアソン・エデュケーション刊
A5判・279頁・定価3,150円（税込）　2003年3月刊

OJTをPJT（Pair Job Training）に名称変更!?

ペアプログラミング（ペアプロ）は名前のとおり、2人でプログラミングする作業方法であり、XP（Extreme Programming）におけるプラクティスの1つとして有名である。本書では、そのペアプロの効能、方法、注意事項などを解説している。

2人で意見を出し合いながらの作業なら、ケアレスミスも見つけやすく、パートナーの技を盗むことも簡単である。特に品質向上の効果が期待でき、さらに組織としての人材育成が可能となるなどの特徴ももつ。このような作業はOJT（On the Job Training）を連想させ、新人とエキスパートのペアを想像してしまうが、新人同士の組合せ、さらに性格の違いによる組合せ方法などについても解説されている。これらを参考にすれば、ペアプロ作業をより効果的に実施できる。私にもペアプロの経験があるが、この書籍を読んでおけば良かったと後悔している。

ペアで作業する方法はプログラミング以外にも利用できる。属人性が高く、文書化や定型化ができない作業においては有効な方法であろう。ペアモデリング、ペアデバッグなどが有効で、一部では実践していると聞いている。1人より2人。当たり前であるが、工数や環境などから実施していない作業方法である。あえて名前に「ペア」を付けて作業すると説得もしやすく、効果が上がるかもしれない。OJTも名前をPJT（Pair Job Training）と変えることで、効果的な後進育成の方法に変わるかもしれない。（渡辺　登）

ソフトウェア・エンジニアリング関連イベントカレンダー

作成：SEC journal編集委員会

開催時期	開催日	イベント名	主　催	開催場所	URL
2006年1月	30日（月）～31日（火）	JaSST '06 in Tokyo	ソフトウェアテストシンポジウム（JaSST）実行委員会	東京都千代田区・都市センターホテル	http://www.jasst.jp/jasst06e/attribute.html
2月	9日（木）～10日（金）	Developers Summit 2006（デブサミ2006）	翔泳社	東京都目黒区・目黒雅叙園	http://www.seshop.com/event/dev/
3月	6日（月）	組込みソフトウェアフォーラムin福岡	社団法人　日本システムハウス協会（JASA）九州支部	福岡県福岡市・福岡システムLSI総合開発センター	http://www.jasa.or.jp/top/
	7日（火）	日本のコンピュータ生誕50周年記念シンポジウム	社団法人 情報処理学会	東京都新宿区・工学院大学新宿キャンパス	http://www.ipsj.or.jp/
	7日（火）～10日（金）	第68回全国大会（学会創立45周年記念大会）	社団法人 情報処理学会	東京都新宿区・工学院大学新宿キャンパス	http://www.ipsj.or.jp/
5月	10日（水）～11日（木）	Embedded Technology West 2006／組込み総合技術展 関西	社団法人 日本システムハウス協会（JASA　社団法人 組込みシステム技術協会）	大阪府大阪市・マイドームおおさか	http://www.jasa.or.jp/etwest/
	17日（水）～19日（金）	IPAX2006（ビジネスショーと共同開催）	IPA	東京都江東区・東京ビッグサイト	http://www.ipa.go.jp/
6月	12日（月）～13日（火）	SEC Forum 2006	IPA/SEC	東京都千代田区・大手町サンケイプラザ	http://sec.ipa.go.jp/
	28日（水）～30日（金）	SODEC	リード・エグジビション・ジャパン	東京都江東区・東京ビッグサイト	http://www.sodec.jp/
	28日（水）～30日（金）	ESEC	リード・エグジビション・ジャパン	東京都江東区・東京ビッグサイト	http://www.esec.jp/
7月	7日（金）	ITスキル標準プロフェッショナルコミュニティ（IPCF2005）	IPA/ITスキル標準センター	東京都港区・明治記念館	http://www.ipa.go.jp/
10月	2日（月）	情報化月間記念特別行事	経済産業省	東京都港区・全日空ホテル	http://www.ipa.go.jp/
	5日（木）～7日（土）	ネットワーク・セキュリティ・ワークショップ in 越後湯沢 2006	NPO新潟情報セキュリティ協会（ANISec）	新潟県湯沢町・湯沢町公民館／イナモト旅館	http://www.yuzawaonsen.gr.jp/conf/
	11日（水）～13日（金）	SEPG Japan 2006（仮称）	日本SPIコンソーシアム（JASPIC）	茨城県つくば市・つくば国際会議場	http://www.jaspic.jp/
	18日（水）～20日（金）	Security Solution2006	日経BP社	東京都江東区・東京ビッグサイト	http://expo.nikkeibp.co.jp/secu-ex/
	中・下旬	IPA Forum 2006	IPA	調整中	http://www.ipa.go.jp/
11月	15日（水）～17日（金）	Embedded Technology 2006／組込み総合技術展	社団法人 日本システムハウス協会（JASA　社団法人 組込みシステム技術協会）	神奈川県横浜市・パシフィコ横浜	http://www.jasa.or.jp/et/

＊JASA「社団法人　日本システムハウス協会」は、2006年4月1日「社団法人 組込みシステム技術協会」に名称変更予定です。

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いたします。

編集後記

SECが発足し2回目の新年を迎え、「SEC journal」も表紙イメージを一新しました。今年一年お付き合いするイメージのため、最後までなかなか決まりませんでしたが、如何でしょうか。

今年のカレンダーは正月休みも短く、皆様早々にエンジンをかけられたことと思います。SECでも、成果物（報告書・出版物）の作成とその公開（各種展示会・セミナー・Web）準備に追われる日々が始まりました。

なかでも、主催する「SEC Forum 2006（6月12日、13日）」には力が入っており、2005年度の成果をわかりやすく発表するために、SEC研究員が日夜頑張っております。

さて、「SEC journal」前号は、SEC journal創刊記念論文の特集号で、創刊の記念として特別に募集した論文の中から、査読委員会、審査委員会を経て採録された論文を掲載いたしました。今回掲載した論文は、昨年から募集を開始した論文で、新規に発足した「SEC journal論文編集委員会」により査読され、採録された3件です。（論文の投稿を頂けるか非常に不安でしたが、一安心）

「SEC journal」では、継続して論文を募集していますので、投稿をお待ちしております。採録された論文は、IPA Forum 2006（昨年同様開催予定）でのSEC journal論文賞（副賞付き）の対象となります。

SECでは、活動の1つとして地域活性化をテーマとしており、地域密着の支援をしております。その成果として、新規に「地域からの発信」という新シリーズを掲載開始いたしました。どの地域から紹介を始めるか、非常に多くの候補から、今回「岐阜県の情報産業振興の取り組み」を選択いたしました。これら地域からの発信は、「SEC journal」だけでは掲載しきれないため、メールマガジンでの配布も検討開始いたしました。是非SEC Webサイトのトップページから利用者登録を行い、SECメルマガもご購読ください（現在、SECメルマガは約1,500名にご登録いただいております。）

また、昨年は「東京以外でもイベントを」というご意見をいただきました。社団法人 日本システムハウス協会（4月より「社団法人 組込みシステム技術協会」に名称変更予定）が「Embedded Technology West 2006」を大阪市・マイドームおおさかで開催予定（5月10日、11日）です。SECスペシャルセッションも予定しておりますので、ご期待ください。

最後に、技術解説、組織紹介で執筆いただいた方々に、短納期での執筆依頼にお応え頂き深く感謝いたします。

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

SEC journal 第2巻第1号（通巻5号） 2006年1月31日発行

独立行政法人 情報処理推進機構 2006

編集兼発行人 〒113-6591 東京都文京区本駒込2-28-8 文京グリーンコート センターオフィス16階
独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター 所長 鶴保 征城
Tel.03-5978-7543 Fax.03-5978-7517
http://sec.ipa.go.jp/

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

お知らせ

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。

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

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

応募要項

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-journal@ipa.go.jp

その他

論文の著作権は著者に帰属しますが、採択された論文については 「SEC journal」への採録、ホームページへの格納と再配布、論文審査会での資料配布における実施権を許諾いただきます。
提出いただいた論文は返却いたしません。
応募時の個人情報の取扱いは下記のとおりです。
SEC内の審査事務局にて管理し、論文審査に係わる査読委員、審査委員とSECが行う広報活動（論文公募、各種イベントの案内、実態調査など依頼）で使用することを許諾いただきます。

応募様式

応募様式は、下記のURLをご覧ください。

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

SEC journal バックナンバーのご案内

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

SEC journal No.5
第2巻第1号（通巻5号）
2006年1月31日発行　

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
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

ISSN 1349-8622