

SEC[®]

journal

Software Engineering Center

8

SEC journal 論文賞受賞論文

コードレビューの密度と効率がコード品質に与える影響の分析

中野 裕也, 水野 修, 菊野 亨, 阿南 佳之, 田中 又治

SEC journal 論文賞受賞論文

Java開発者のオンデマンド・ラーニングを支援するソシオテクニカル環境

葉 雲文, 山本 恭裕

組織的プロジェクトマネジメント成熟度モデル(OPM3)の概要と適用上の留意点

平石 謙治

高度化プロセスにおける定量的プロジェクト管理の実践と効果

岩切 博, 大曾根 一将, 藤原 良一, 中前 雅之

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.8目次

- 1 巻頭言
細川 泰秀(社団法人 日本情報システム・ユーザー協会 専務理事)
- 2 所長対談 : Kyo-Chul Kang 韓国・浦項工科大学校 教授
プロダクトラインエンジニアリングの普及に向けて
- 6 「SEC journal」論文賞
受賞論文発表
- 8 受賞者プロフィール
- 10 優秀賞受賞論文
コードレビューの密度と効率がコード品質に与える影響の分析
中野 裕也 水野 修 菊野 亨 阿南 佳之 田中 又治
- 18 Java開発者のオンデマンド・ラーニングを支援する
ソシオテクニカル環境
葉 雲文 山本 恭裕
- 26 「SEC journal」論文賞 審査委員会審査報告
相磯 秀夫(論文賞審査委員会委員長 東京工科大学学長)
- 27 論文講評とSECへの期待
井上 克郎
冨永 章
片山 卓也
重松 崇
有賀 貞一
柳木 好明
- 30 論文の歩き方
ソフトウェア・エンジニアリング実証論文をよむ
平山 雅之
- 34 論文
組織的プロジェクトマネジメント成熟度モデル(OPM3)の概要と適用上の留意点
平石 謙治
- 42 高度化プロセスにおける
定量的プロジェクト管理の実践と効果
岩切 博 大曾根 一将 藤原 良一 中前 雅之
- 50 BOOK REVIEW
- 51 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
- 52 あとがき
- 53 お知らせ(論文募集 / SEC journal バックナンバー)

ISユーザ企業とITベンダ企業の差



社団法人 日本情報システム・ユーザー協会
専務理事

細川 泰秀

ユーザの見方、ベンダの見方

社団法人日本情報システム・ユーザー協会（JUAS）は、業種を超えた情報システムユーザ企業が集う産業情報化推進のための団体であり、正会員143社、賛助会員123社からなる。

物には裏と表がある。ユーザからの見方とベンダからの見方にも差がある。

商品の3要素QCDについて見ると、ユーザ側は商品の値段は安く、品質がよく、工期は短いほどよい。ベンダ側から見れば、値段は高く、品質はよいに越したことはないが、実際には適度なところがよく、工期も極端に短いのは困るといった微妙な差がある。とくにソフトウェア商品はこの差が微妙である。

ユーザとベンダの間には、なんらかの評価値があってこそ優劣が判断できるのであるが、なぜか漠然としていて評価基準値がない。評価できないものは、改善もできない。

日本のソフトウェア受託産業は、世界的な水準から見ると高品質であるが、企業の利益率は低く、長時間残業等労働条件は悪いので最近若者には人気がなくなってきていると聞く。

現時点でのソフトウェア産業界においては、「よいものは値段が高くて当然」の常識が通じにくいことがJUASの調査でわかってきた。何故だろうか？

プロダクト志向とプロセス志向

消費者が車を買う場合に「この車はどのようなプロセスで作られてきたのか？」に関心はない。

「燃費が低くて、乗り心地がよく、安全性に優れていて、安ければよい」のが消費者の選択基準である。これはプロダクト評価（志向）を中心にした考え方である。ほとんどのユーザ企業はこのような評価基準を基にして商品の競争力を測っている。しかしソフトウェア産業はかなり異なっている。

各開発フェーズで「このようなことを守って開発しております」なるプロセス志向が古くから使われている。「何をしなさい」なる基準はISO等で公表されているが、「どのようにしなさい」、「商品品質の目標はこれです」なる基準は存在していない。つまり常識がないのである。

ほとんどバグを出さないプログラムと、いつもバグだらけで「ひんしゆく」をかつているプログラムの評価差は明確にされていない。

商品には開発直後の創始期、成長期、成熟期がある。ソフトウェア商品もそろそろ成長期から成熟期に入りつつある。成熟期に評価されるものは安定した品質である。

SECの成果に期待

そこに日本のソフトウェア産業の期待を背負って登場してきたのが、IPAのソフトウェア・エンジニアリング・センター（SEC）である。SECの旗振りのもと、JUASも共同し、ユーザの目から見たソフトウェア商品の評価基準値を作成している。まだ誕生して2年であるが数々の知見が判明してきた。『プロジェクトの工期は $2.4 \times (\text{投入人月の立方根})$ 』、『ベンダの納入時に、欠陥数を発注額500万円当たり1個に抑えたプロジェクトは約半分』、『優秀なプロジェクトマネージャは未経験プロマネの1/5しか欠陥を残さない』等、明日から活用してもらえばシステムトラブルが減少する基準ばかりである。

これからも、システムリスク、usability、テスト方式等追求すべきテーマはたくさん残されている。

SECを中心とした活動成果によって、やがて日本からシステムトラブルが消え、世界から日本の情報産業の優秀さが評価される日が来ることを熱望している。

プロダクトラインエンジニアリングの普及に向けて

ソフトウェアをリユースすることによってソフトウェア開発の生産性を向上させる手法であるプロダクトラインエンジニアリングに対する関心が世界的に高まっている。その第一人者であるKyo-Chul Kang氏（韓国・浦項工科大学校教授）とIPA/SECの鶴保征城所長がプロダクトラインエンジニアリングの普及に向けた課題やアジアにおけるソフトウェア開発をめぐる課題の解決策について縦横に語り合った。



鶴保 征城（つるほ せいしる）
1966年大阪大学大学院工学研究科電子工学専攻修士課程修了後、日本電信電話公社（現NTT）入社。NTTソフトウェア研究所長、NTTデータ通信株式会社取締役開発本部長、同社常務取締役技術開発本部長、NTTソフトウェア株式会社代表取締役社長を歴任し、2004年10月独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター所長に就任。

- ・ 社団法人 情報処理学会 会長（2001年～2002年）
- ・ XMLコンソーシアム 会長（2001年～）
- ・ 高知工科大学工学部情報システム工学科 教授（2003年～）
- ・ 日本BPM協会 副会長（2006年～）
- ・ 実践的ソフトウェア教育コンソーシアム 会長（2006年～）
- ・ 社団法人 電気情報通信学会 フェロー
- ・ 社団法人 情報処理学会 フェロー

鶴保 日本におけるソフトウェアの問題は2つあります。1つは商品の問題です。OSやミドルウェア、アプリケーションパッケージ等の商品に、日本発のものが少ないことです。2つ目は、企業向けシステムを開発するシステム・インテグレーション及びハードウェアに組み込まれて動作する組込みソフトの開発において、ソフトウェアエンジニアリングの手法や技術が十分に使われていないという問題です。そうした課題を解決するために、ソフトウェアプロダクトラインがキーとなるのではないかと考えています。Kang先生がソフトウェアプロダクトラインに取り組むことになったきっかけはどのようなものだったのですか。

Kang 私は1970年代に米国ミシガン大学でソフトウェアのCASEツールの開発を行い、ツールをどのように利用するかという方法論に関する知識を得ました。ミシガン大学で博士号を修得して2年ほど教員として勤めたあと、ベル研究所に移り、そこでソフトウェアエンジニアリングにかかわる改善活動に取り組みました。

ベル研究所では、商品開発がどういう流れで進んでいるかを

自分の目で見ることができました。新しい商品を開発する際に、開発担当者はまず類似するシステムを探します。そして、類似するシステムが見つかったら、そのソースコードをコピーしたり修正して新しい商品を開発するというのが商品開発の流れです。この手法では、コードの再利用が行われているのですが、その手法はシステム化されていませんでした。私は、コードを再利用する際に、システム化された手法が必要なのではないかと考えたのです。

鶴保 ベル研究所ではどのような商品が開発されていたのですか？

Kang 私の専門のCASEツールにかかわるもので、例えばエレベータの制御システムや製造プロセスの管理システムといったものです。

鶴保 私は30年くらい前に交換機のソフトウェアを開発していたのですが、そのときにソフトウェアのリユースの研究をしていました。そのころは、プロダクトファミリという言葉はなく、母体と呼んでいました。コードの母体を定義し、そこから市内交換用ソフトウェアや市外交換用ソフトウェアを切り出して現場に適用したのです。その手法も、一種のプロダクトライン開発に当たると思います。しかし、日本のソフトウェアメカは各社各様でリユースを推進しているのが現状です。

Kang ヨーロッパでは、欧州委員会の主導で各国の政府が技術開発に積極的に取り組んでいます。また、ヨーロッパの企業は自分たちが開発した技術をオープンな形で共有しています。しかし、アジアを見ると鶴保所長がおっしゃったように企業が個々に活動しているので、知識が共有されていません。そのため、それぞれの企業が同じような過ちをいろいろなところで繰り返すという状況になっています。そこを改善することが必要だと考えています。

プロダクトラインのソフトウェアエンジニアリングのカンファレンスに出席すると、ほとんどの参加者がヨーロッパないし

はアメリカからの人で、アジアからの参加者はきわめて少ないですね。

鶴保 プロダクトラインの適用対象は主に組み込みソフトウェアなのでしょうか。

Kang 多くの企業が組み込みソフトウェアの分野でプロダクトラインエンジニアリングのアプローチを行っています。

鶴保 エンタープライズ分野のアプリケーション開発に適用するということはないのでしょうか。

Kang SAPやPeopleSoft等がエンタープライズ向けのパッケージソフトウェアにプロダクトライン開発のコンセプトを適用しています。彼らはソフトウェアコンポーネントをパッケージ化してエンタープライズ製品を提供しています。製品を開発する際に、過去の実験やベストプラクティスの情報を入れて製品の開発を行っているのですが、最初にコアのコンポーネントを定義し、そこから個々のアプリケーションを作るという手法をとっています。それは正にプロダクトラインエンジニアリングの手法だと考えています。

エンタープライズソフトウェアと組み込みソフトウェアの違いは非常におもしろいと思います。エンタープライズソフトウェアはビジネスのマネジメントを行うためのもので、世界的に標準化が進んでいます。標準化の一環としてB to Bの電子商取引がグローバルに行えるという動きが進んでいます。また、SAPのエンタープライズ向けソリューション等も世界中で展開することが可能となっています。一方、組み込みソフトウェアの分野では、個々にアプリケーションシステムが存在し、標準化ができていない状況です。

鶴保 プロダクトラインエンジニアリングは、ソフトウェア開発の生産性を向上させることを目的としていますが、ソフトウェアの品質や信頼性確保という点でも寄与するのでしょうか。

Kang はい、寄与します。プロダクトラインエンジニアリングを行うに当たっては、ソフトウェアコンポーネントをプロダクトラインのフレームワークの中で用いてソフトウェアを開発するのですが、そこでは既に信頼性がチェックされたコンポーネントを再利用します。そのため、品質上の属性も検証されています。もちろん、テストや検証作業がゼロになるわけではありません。しかし、検証作業を削減することが可能となります。

ソフトウェアエンジニアの意識と行動を変えることが大切

鶴保 プロダクトライン自体、日本人になじみがないのですが、それはコンセプトの問題でしょうか。それともテクノロジーの問題



Kyo-Chul Kang
Kyo-Chul Kangは、1982年ミシガン大学で博士号を取得した後、Software Engineering Instituteにシニアメンバとして参加するまでは、同大学の客員教授、またBell Communications ResearchおよびAT&Tの技術スタッフとして勤務していた。現在は韓国 Pohang University of Science and Technology(浦項工科大学校)で教授を務める。2004年スペインのマドリッドで開催された International Conference on Software ReuseではGeneral Chairを務め、2007年に日本の京都で開催される International Software Product Lines Conference (SPLC2007)でもChairを務める。今まで多くの国際会議またはワークショップで様々な役割を担当し、2001年から2003年までは韓国ソフトウェア研究所(KSI)傘下の韓国ソフトウェア振興院(KIPA)の初代会長を務めた。研究分野としては要求工学、コンピュータ支援ソフトウェア工学、ソフトウェア再利用、生産ライン工学が含まれる。

題でしょうか。

Kang アジアの人にとっては、おそらくプロダクトラインというよりもプロダクトファミリ、すなわち製品群と言ったほうが素直に耳に入るのではないかと考えています。実際、ヨーロッパではプロダクトファミリと呼んでいます。プロダクトラインというと、どうしても製造ラインをイメージしがちですが、ソフトウェアエンジニアリングの観点からいうと、プロダクトラインは関連する商品群を指します。

そこでソフトウェアエンジニアリングが機能するのは、関連する商品群の共通点を共通のベースとして使って、新しい商品群を開発することです。結果として、その流れの中でソフトウェアの再利用を行う。それが、プロダクトラインのコンセプトです。

鶴保 商品进行分类するという事は、商品を横断的に把握して上位概念を抽出することですから、技術者にとって苦手の分野ではないでしょうか。とくに、ソフトウェアエンジニアは、ソースコードの問題やデザインには興味を持つけれど、商品やドメインの分類・分析は自分の仕事でないと考えがちです。その点の難しさについてどう思われますか。

Kang プロダクトラインのコンセプトを組織に導入して展開していくためには、まず、ソフトウェアエンジニアの意識・行動を変えることを考えなければならないでしょう。通常、ソフトウェアエンジニアは、「私たちが作らなければならないアプリケーションのスペックや要件を教えてほしい」と言ってきます。そして、いったん、コードを渡したら「それで終わり」と考えています。そういうソフトウェアエンジニアの行動や姿勢を変えていかなければならないと考えています。

というのも、実際の製品のライフサイクルを考えると、ソフトウェアの開発が占めるウェイトは20%~30%しかなく、残りの70%~80%を占めるのはメンテナンスだからです。です

から、長期的に考えると、ソフトウェアエンジニアは、製品はメンテナンスされるということを意識して作っていかねばなりません。ハードウェアのメンテナンスは部品を交換すれば済みます。しかし、ソフトウェアの場合は、フィーチャを加えたり、逆に削除したり、フィーチャの内容を変更するということがソフトウェアエンジニアが行わなければなりません。そこで、製品を開発する段階で、どのような形でメンテナンスを行うのか、メンテナンスを考えた活動が必要となります。そこで、出てくるのがドメイン分析です。

ソフトウェアエンジニアは、特定のアプリケーションだけを考えてフィーチャを作るのではなく、類似するシステムを見ながらフィーチャを考え、そのフィーチャが将来、他のシステムにどのように展開できるのかをということを視野に入れた開発姿勢が必要だと思えます。

鶴保 それは非常に難しい問題ですね。日本では、理系と文系という言葉があります。ソフトウェアエンジニアは主として理系ということになります。韓国やヨーロッパに理系・文系という言葉があるかどうかわかりませんが、ドメイン分析は理系の人間でもできるのでしょうか。それとも、文系の人間が行うのでしょうか。

Kang 私たちの大学でも、コンピュータサイエンスを学ぶ人たちはアプリケーションの知識を持っていないというジレンマがあります。しかし、そもそもすべての人がすべてのドメインに関する知識を持つことは不可能です。そのため、ソフトウェアエンジニアにはドメイン分析をするために、そのドメインの専門家と一緒に活動することによって、ドメインの基本理念を理解することを推奨しています。ドメインの専門家はいくつかの特定のアプリケーション、特定のドメインに関する知識を持っています。彼らが持っている知識を他の人も活用できるよう、知識をコード化することも行われています。

鶴保 ドメインとは、断片的な知識の集合ではなく、体系化された知識なのですね。

Kang おっしゃるとおりです。私たちは大学で、システム化された方法で情報を収集し、その情報を分類し、コード化する方法を学生に教えています。

鶴保 その対象は大学生ですか、それとも大学院生ですか。

Kang 大学生に対しては、ソフトウェアの開発コースを提供しています。また、ソフトウェアエンジニアリングに関する初級コースも提供しています。一方、プロダクトラインのエンジニアリングに関する講義は、大学院生向けのコースとなっています。大学院生は、実際の経験をしながらプロダクトを視野に入れてソフトウェアエンジニアリングについて学んでいきま

す。

鶴保 コビキタスコンピューティングのアプリケーションでは、製品フィーチャのダイナミックなリコンフィグレーションが必要となると指摘されていますね。非常に興味深い指摘です。

Kang これからのコビキタスコンピューティング環境でさまざまなシステムの開発が増えた場合、フィーチャを動的にバインディングすることが必要になってきます。コビキタスコンピューティング環境では、様々なユーザがいろいろな場所に移動します。そこで、移動する場所によってコンテキストを変更することが必要になります。それを実現するために、私たちは、アプリケーションの中に自己調整機構（セルフアジャストメント）という機能を盛り込んでいます。コビキタス環境の中でプロダクトラインのアプローチを取り入れるためには、ソフトウェアの再利用に加えて、フィーチャバインディングを動的に行うことも考えていく必要があるということです。

そうすると、別の次元の話になるので難しいテーマとなります。私たちは、製品自体の中にプロダクトラインエンジニアリングのノウハウを入れることによって、コンテキストに応じたフィーチャの動的なバインディングを実現するという流れを考えています。

鶴保 それは、非常に難しい技術だと思います。何年ごろに実用化できるとお考えですか。

Kang 実際にコビキタス環境で利用されるアプリケーションにどのような要件が求められるのか、私たちはその要件を検討している段階です。それに加えて、製品の再構成、リコンフィグレーションをする際にどのようなパラメータを使っていかなければならないかということも検討しているところです。こうしたことをすべて考慮したうえで製品作りを行うことになるので、現在のところ、フィーチャの動的なバインディング技術がいつ実現できるかははっきりと答えられる段階にはきていません



ん。

鶴保 従来から、非常に高度な技術は防衛産業や航空産業からスタートすることが多いのですが、動的なバイディング技術の考え方も防衛産業や航空産業の応用からきているのですか。

Kang 動的なバイディング技術に対する大きなニーズはまずコミュニケーション産業にあると思います。コミュニケーションシステムの中で、バイディング技術が必要とされると考えられます。また、軍事産業にもニーズがあると思います。軍事産業はミサイルを作っているだけではありません。軍隊には基本技術があり、コミュニケーションシステムでつながっています。そういう意味では軍用のシステムにもバイディング技術のニーズが存在しています。

産学連携プロジェクトは 大学教育に大きな役割を果たす

鶴保 日本では、学生の間でソフトウェア産業の人気の下がってきています。韓国ではどうなのでしょう。

Kang ソフトウェア産業の勢いがなくなったという点は、韓国もまったく同じですね。それは、世界中で共通している現象なのではないかと思います。韓国でもITベンチャーのバブルブームが起きたのですが、それがはじけたのが現実ですから。若い人たちは、ソフトウェアエンジニアの仕事を隅に座ってコードを書いているような退屈な仕事だと思っているようです。そういう状況なので、優秀なソフトウェアエンジニアを探すのは難しいことです。そのため、需要に対して供給が追いついていないというのが韓国の現状です。

鶴保 一方で、日本の産業界からは大学教育に関して大きなクレームが出ています。それは、IT産業に入ってきた大学卒業生は、極端に言うとも使いものにならないというクレームです。昨年の日本経団連の調査によると、使いものになる新卒者は全体の1割という結果が報告されています。IT産業界からは、大学に対して「産業界に役立つ人材を育成してほしい」という要求が出されています。韓国ではどうなのでしょう。

Kang 韓国でもまったく同じ問題を抱えています。韓国IT産業界でも、韓国の大学は役に立たない分野の学生ばかりを育てていると言われているのは事実です。特に韓国の大学はそうなのですが、アジアの大学では、実務者を養成するというより研究者を養成することを目的にカリキュラムが組まれています。結果として、産業界のニーズがカリキュラムに反映されていません。もう1つの問題点は、教授陣の中で実際に職務経験を持った人がきわめて少ないことです。職務経験を持っていても研

究職がほとんどで、産業界から学術分野に移行するケースは極めて珍しいことです。

理想的な解決策は、コンピュータ科学に関する大学のプログラムを2つに分けることです。1つは、情報科学的な分野、もう1つは、実践に則した方法やプロセス、マネジメントを教育するソフトウェアエンジニアリングの分野です。ただ、現実的にはそれができていないので、理想を実現するのは難しい点もあるでしょう。

鶴保 Kang教授はミシガン大学におられました。米国では産学の連携は進んでいるのですか。

Kang 米国では、ソフトウェアエンジニアリングコースを設ける大学が増え続けています。また、実際のプロジェクトを通して産学の連携がうまく取れています。産学連携プロジェクトに大学生が入ることによって、大学生は産業界のニーズや課題を的確に把握することができるので、産学連携プロジェクトは価値があると思います。

鶴保 日本では、産業界でソフトウェア開発の実務を経験した人が大学で継続的（年間に20回、30回程度）に教える仕組みを設けることが考えられています。ソフトウェアの現場の人が学生を教育することは有効だと思われませんか。

Kang 非常に価値があることだと思います。実務経験をもった人が直接、大学生に対して実務について話すことは、大学生にとって貴重な機会となるのです。

鶴保 いま、日本の組込みソフトウェア産業は非常に規模が大きくなっています。それに伴って、ソフトウェアの開発期間が短くなり、トラブルも増加しています。組込みソフトウェア産業の状況に危機感を持っているのですが、その状況を関係するためにサジェスションをいただけませんか。

Kang 私は、好むと好まざるとにかかわらず、組込みソフトウェアを開発している企業はプロダクトライン開発のアプローチを取るしか、選択肢が残されていないのではないかと考えています。プロダクトラインのアプローチは、唯一、効果的であると証明されているからで、プロダクトラインの手法を採用することが問題解決の道筋になると考えています。

鶴保 日本では、携帯電話会社等がプロダクトファミリという名前は使っていないものの、同様の考え方に基づいたソフトウェア開発のプラットフォームの整備に着手しています。それが、家電や自動車、その他のデジタル機器へと広がっていくと考えられます。SECとしても、業界横断的なアプローチによってソフトウェアプラットフォームの考え方の優れた部分を広めていこうと考えています。

文：小林秀夫 写真：越昭三朗

「SEC journal」論文賞 受賞論文発表

SECは、我が国のソフトウェア産業発展のための様々な取り組みを実施しておりますが、その取り組みの一環としてソフトウェアエンジニアリングに関する論文に賞を設け、表彰を行っております。

今年の「SEC journal」論文賞は、「SEC journal」No.5～No.7に掲載された論文3件及び論文賞向けに投稿された論文9件、合計12件の論文を候補とし、査読委員による厳正な審査の結果4件を優秀賞とし、また審査委員により1件を最優秀賞に選定いたしました。論文をご投稿いただいた皆様の熱意と努力に敬意を表します。

最優秀賞の選定、各賞の発表と表彰は、2006年10月24日に実施したIPA Forum 2006「SECコンファレンス～「SEC journal」論文発表会～」において行いました。審査報告と論文の講評は、本号26頁からご覧ください。

また、SECコンファレンスでは、東京海洋大学教授 佐藤吉信氏による講演「機能安全の標準化について」を行いました。当日の論文発表及び講演の資料は、SEC-Webサイトよりダウンロードが可能ですので、こちらもご覧ください。

(最優秀賞受賞論文『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』及び優秀賞受賞論文『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』は「SEC journal」No.5に掲載、優秀賞受賞論文『コードレビューの密度と効率がコード品質に与える影響の分析』及び『Java開発者のオンデマンド・ラーニングを支援するソシオテクニカル環境』は本号に掲載)

「SEC journal」論文賞審査委員会

委員長	相磯 秀夫	東京工科大学 学長
委員 (50音順)	有賀 貞一	株式会社CSKホールディングス 取締役
	井上 克郎	大阪大学大学院情報科学研究科 コンピュータサイエンス専攻 ソフトウェア工学講座 教授
	片山 卓也	北陸先端科学技術大学院大学 情報科学研究科 教授
	櫛木 好明	パナソニックモバイルコミュニケーションズ株式会社 取締役社長
	重松 崇	トヨタ自動車株式会社 常務役員
	富永 章	日本IBM株式会社 技術顧問 兼 東京大学大学院 工学系研究科 特任教授
	藤本 隆宏	東京大学COEものづくり経営研究センター センター長
	鶴保 征城	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

「SEC journal」論文賞査読委員会

委員長	井上 克郎	大阪大学大学院情報科学研究科 コンピュータサイエンス専攻 ソフトウェア工学講座 教授
委員 (50音順)	大場 充	広島市立大学 情報科学科 情報メディア工学科 教授
	大原 茂之	東海大学 電子情報学部 情報メディア学科主任 教授
	片岡 欣夫	株式会社東芝 研究開発センター システム技術ラボラトリー 主任研究員
	古賀 恵子	株式会社日立製作所 ソフトウェア事業部 プラットフォームソフトウェア本部 サポートサービス部 技師
	津田淳一郎	東芝LSIシステムサポート株式会社 マイコンソフトウェア技術統括部 シニアエキスパート
	平山 雅之	株式会社東芝 ソフトウェア技術センター 企画担当 参事
	松本 健一	奈良先端科学技術大学院大学 情報科学研究科 ソフトウェア工学講座 教授

IPAフォーラム2006

SECコンファレンス～「SEC journal」論文発表会～

2006年10月24日(火) 主催：独立行政法人 情報処理推進機構 後援：経済産業省

最優秀賞

賞品：賞状、記念品 副賞：金1,000,000円

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*1

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

優秀賞

賞品：賞状、記念品 副賞：金500,000円

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*2

菊地 奈穂美, 安藤 津芳, 水野 修, 菊野 亨

コードレビューの密度と効率がコード品質に与える影響の分析

中野 裕也, 水野 修, 菊野 亨, 阿南 佳之, 田中 又治

Java開発者のオンデマンド・ラーニングを支援する ソシオテクニカル環境

葉 雲文, 山本 恭裕



上段左より 有賀 貞一・井上 克郎・富永 章・重松 崇
平山 雅之・松本 健一・菊野 亨・田中 又治・阿南 佳之・中野 裕也
鶴保 征城・葉 雲文・菊地 奈穂美・大杉 直樹・水野 修・相磯 秀夫・佐藤 吉信

* 1 SEC journal No.5 PP.16-25,2006, * 2 SEC journal No.5 PP.26-35,2006にそれぞれ掲載

(敬称略)

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 (奈良先端科学技術大学院大学情報科学研究科, 特任助手, 博士(工学))
 角田 雅照 (奈良先端科学技術大学院大学情報科学研究科, 博士後期課程, 修士(工学))
 門田 暁人 (奈良先端科学技術大学院大学情報科学研究科, 助教授, 博士(工学))
 松村 知子 (奈良先端科学技術大学院大学情報科学研究科, 特任助手, 博士(工学))
 松本 健一 (奈良先端科学技術大学院大学情報科学研究科, 教授, 博士(工学))
 菊地 奈穂美 (IPA SEC, 研究員, 博士(工学))



大杉 直樹 角田 雅照 門田 暁人 松村 知子

この論文では、ソフトウェア開発プロジェクト管理のための定量的見積手法について述べました。特に、複数組織で収集されたデータ値が欠損している「穴あき」が沢山あります！)に対して有効に機能する手法を定量的に分析しました。

プロジェクト毎に収集した工数、工期、ファンクションポイントなどのデータから、数学的に工数を算出する手法は数多く提案されています。この論文では、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という4つの手法をとりあげ、その手順を紹介し、各々の精度(見積り)の正確さを評価しました。

奈良先端大と大阪大学が共同で進めているEASEプロジェクトでは、定量的データを用いた開発支援について研究しています。データ自動収集、データマイニングに加え、この論文でとりあげた見積手法も重要な研究テーマになっています。

この論文でとりあげた協調フィルタリングは、プロジェクトの初期から研究を続けている定量的見積手法です。

特に「穴あき」が沢山あるデータに対して強く、ソフトウェア開発プロジェクトのデータに適していると言われております。応用研究も進んでおり、まもなく、ワンクリック見積ツールとして自動化ツールを公開の予定です。

EASEプロジェクトでは、多くの企業様と共同で研究を進めさせていただいております。すでにお持ちのデータに対する分析をはじめとして、開発プロジェクトの中に入り込みデータ収集と分析をお手伝いさせていただくケースもあります。この論文をきっかけとし、ぜひお声掛けをいただければと期待しております。



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 (沖電気工業株式会社ビジネスサポート本部, スペシャリスト, 博士(工学))
 安藤 津芳 (沖電気工業株式会社ビジネスサポート本部, 担当部長)
 水野 修 (大阪大学大学院情報科学研究科, 助手, 博士(工学))
 菊野 亨 (大阪大学大学院情報科学研究科, 教授, 工学博士)



菊地 奈穂美 安藤 津芳 水野 修 菊野 亨

ソフトウェア開発を行う企業では、プロダクトの品質・納期・生産性を達成し向上することが常に課題になっています。私たちは、この論文では、特にソフトウェアプロダクトで最終的に良い品質(フィールド品質)の達成を目的として、どのような開発活動を行うと、効果的に良い品質の達成ができるだろうか、という課題にチャレンジしました。背景の1つとして、通信分野において納期までに高い品質が求められていること、そのため効果的な改善策を求めていたことが、大学との共同研究の動機となりました。

品質向上のための施策として、幾つか改善点を考えついてもどれが効果的かの判断は容易ではありません。また、この研究に取り組む前に検討・経験したCMM等のモデルではアセスメントに要する時間の多さと、モデル文書を超える改善案には至りにくいと感じました。

このような問題意識から、この論文では、実際の企業で開発活動の改善エリアを抽出するための効果的な分析アプローチを考案し、実プロジェクトで適用を行いました。分析では、従来から組織でとっていたメトリクスデータに加え、開発活動状況を示すものとして追加で必要と考えた新しいメトリクスも導入しました。論文で対象とした組織では、この方法での分析結果をもとに具体的な改善施策へ展開と効果に結びついています。

このアプローチでは、対象組織を知る人がインタビュー者となり、対象組織のPMの短時間の協力で、比較的短時間で要因候補がつかめます。メトリクス設計を注意して行いデータ収集をしますと要因分析ができます。このような目的指向でのメトリクス選定と分析で改善すべき活動の抽出ができることは、改善検討に有用と考えます。

コードレビューの密度と効率がコード品質に与える影響の分析

中野 裕也 (大阪大学大学院情報科学研究科, 博士前期課程2年)

水野 修 (大阪大学大学院情報科学研究科, 助手, 博士(工学))

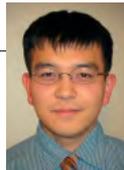
菊野 亨 (大阪大学大学院情報科学研究科, 教授, 工学博士)

阿南 佳之 (オムロンソフトウェア株式会社品質保証部, 品質保証グループマネージャ)

田中 又治 (オムロンソフトウェア株式会社物流ソリューション事業部, 事業部長)



中野 裕也



水野 修



菊野 亨



阿南 佳之



田中 又治

本研究は大阪大学大学院情報科学研究科情報システム工学専攻の菊野研究室とオムロンソフトウェア株式会社との共同研究として実施したものです。

オムロンソフトウェア株式会社では機器組込み型ソフトウェアの開発を中心に行っており、これまでに多くのデータを収集し、データの分析結果に基づくプロセス改善活動を行ってきました。そうした活動の中で、過去に設定した管理基準の適切さを評価することが必要となり、「レビューの量」に関する基準についての分析を行いました。その結果、レビューの量は確かにここ数年で増え続けているものの、ソフトウェアの品質には有意な改善が見られないということがわかったのです。そのため、従来の基準とは別に新たにレビューに関する基準を

設定する必要に迫られました。本研究はこうした背景の下に実施されました。

菊野研究室では、これまでも多くの企業との共同研究を実施しましたが、オムロンソフトウェア株式会社との共同研究にあたっては、まず、企業の現場において、具体的ではないにしても意識されている仮説の検証を行うことから始めました。取り上げた仮説は、『レビューの効率とレビューにおける不具合指摘の密度の間には何らかの関係がある』です。これを検証するために、私たちは700件ほどもあるデータの提供を受け、その統計的分析を行ったところ、現場で信じられていた本仮説について、データより裏付けがとれることが明らかになったのです。本論文ではその結果について詳しく述べます。

Java開発者のオンデマンド・ラーニングを支援するソシオテクニカル環境

葉 雲文 (株式会社SRA先端技術研究所, シニア研究員, Ph.D.)

山本 恭裕 (東京大学先端科学技術研究センター, 特任助教授, 博士(工学))



葉 雲文



山本 恭裕

ソフトウェア工学分野の研究は、従来、技術的な側面とマネジメントの側面に注目したものが多く、ソフトウェア開発におけるヒューマンファクターは重要視されていません。我々は、ソフトウェア開発を知識集約型活動であると考えています。本研究は、実務に携わる開発者が開発プロセスにおいて知識の獲得と活用を行う際に付きまとう難点を分析し、社会的な側面を考慮して支援しようとするものです。

本論文では、Javaクラスライブラリの習得問題を取り上げました。オブジェクト指向開発において再利用ライブラリが不可欠なものであり開発効率と品質向上に大きく貢献することはよく知られていますが、それを学習する際の負担についての検討は極めて少ないのが現状です。個々の開発者が自ら学習を重ねていくべきものだと認識されているのかもしれませんが、ソフトウェア開発者

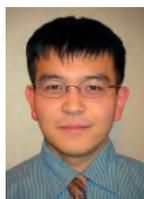
は、新しいライブラリ部品をいかにして学習するでしょうか。まずは、ドキュメントを調べることから始まります。しかしドキュメントは、部品利用者の立場からは抽象的に思われ理解し難いことがあります。二番目の方法は、使用例を見習うことです。具体的な使用コンテキストから使用方法を学ぶことができます。三番目の有効な方法は、当該部品に詳しいエキスパートに訊ねることです。

本論文で提案した手法と支援システムは、これらの活動を統合的に支援することを目的とし、現場のソフトウェア開発者に広く利用してもらうことを目指しています。実装したシステムは、オープンソースソフトウェアとして公開しています (URL: http://stepin.cs.colorado.edu/STeP_IN/)。興味を持たれた方はご連絡いただければと思います。

コードレビューの密度と効率が コード品質に与える影響の分析



中野 裕也†



水野 修†



菊野 亨†



阿南 佳之††



田中 又治††

ソフトウェアプロジェクトにおいてレビューを行うことは品質確保の面から重要な作業とされている。一方でレビューには相応の工数がかかるため、コストの面からはあまり長い期間レビューをすることは好ましくない。しかし、レビューを行う効率を高くしすぎるとレビューが乱雑になり、最終的な品質の向上があまり期待できない。そのため、レビューの効率とレビューによる指摘密度の関係について分析し、これらが後工程での品質にどう影響を及ぼしているかについて調べることで、適切なレビュー効率でレビューが行われているかを判定する条件を発見できると考えられる。

まずSEPGが関連性があると感じていた2つのメトリクス（レビュー効率とレビュー指摘密度）に注目した。そして、これらのメトリクスに関連した仮説「レビュー指摘密度がある条件を満たさないプロジェクトは、後工程での品質が悪くなる傾向がある」を示すことを本研究の目的とする。

この仮説を検証するために、実際の企業の開発現場から収集された500件のプロジェクトデータを使用して分析を行った。その結果は、レビュー効率に対してレビュー指摘密度が大きいプロジェクトでは後工程での残存不具合数は多くなる、というものとなった。よって、この仮説の正しさを確認できた。また、実際の開発現場への適用法も提案した。

Analysis on Impact of Defect Density and Efficiency of Coding Review to Software Quality

Yuya Nakano, Osamu Mizuno, Tohru Kikuno, Yoshiyuki Anan, and Mataharu Tanaka

This paper shows a result of analysis on impact of defect density and efficiency of coding review activity to software quality. In a certain company, a criterion for the amount of review effort has been used so far. However, since it was determined years ago, the criterion is becoming a mere name, but not in reality. Therefore the SEPG in the company tries to determine new metrics and criteria for review activity.

As new metrics to focus on, we selected two metrics, efficiency of review activity and density of detected defects by review, since the SEPG has felt that there is a relationship between these two metrics. In more detail, we try to show the following hypothesis: if the density of detected defects on review becomes too large or too small from criteria (say, outliered) considering the efficiency of review activity, the quality of the product has some kind of problem and more defect will be found in the successive debug and test activities.

In order to show validity of the hypothesis, we performed statistical analysis on 500 project data. The result of analysis showed that the number of remained defects in the outliered projects is large. Therefore we confirmed that the hypothesis have validity at a certain degree.

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

†† オムロンソフトウェア株式会社, OMRON Software Corporation

1 はじめに

ソフトウェアのプロセス改善手法[HUMPHREY1989]は高い品質と生産性をもたらすための有効で実用的な方法の1つとして知られている。プロセス改善に関しては、実際の組織への適用[HUMPHREY1991], [TANAKA1995]だけでなく形式的なフレームワークを用いた理論的な調査が行われてきた[CIMITILE1994], [PODOROZHNY1997]。特に、ソフトウェアレビューのプロセス改善はソフトウェアの品質を改善する費用対効果に優れた方法であると考えられてきた[EBENAU1993]。実際に、ソフトウェアレビュープロセスのいくつかの定量的評価はシミュレーション結果を用いて妥当性が証明されている[BRIAND1998], [KUSUMOTO1993]。

これまでもレビュープロセスに関する多くの研究が行われてきた[BRIAND1997], [FAGAN1986], [RUSSELL1991], [WELLER1993]。文献[BRIAND1997]では、レビュープロセスにおける再捕獲法に関して述べられている。モデルの目的は、レビュー作業で収集された発見不具合数のデータを使用してソフトウェア製品に残存している不具合の数を見積もることである。しかし、残存不具合数の推定は高い精度で行えることを確認したものの、レビューが製品の品質にどのくらい影響を与えるかということについての議論はされなかった。

文献[FAGAN1986]では、不具合のないソフトウェアを時間通りにより低いコストで開発するためのインスペクションの貢献度について研究されている。さらに文献[RUSSELL1991], [WELLER1993]ではインスペクションに関するケーススタディが提示されていて、多くの実証的経験がまとめられている。しかしそれらのほとんどはインスペクションに引き続いて実施されるテストプロセスをどのくらい改善するかについて議論している。すなわち、これらの研究は主に生産性改善の効果について扱っている。

我々は共同研究を行っている企業との議論の中で、きちんとしたレビューの実施が重要であると考えてきた。そして、ソフトウェアの品質の保証や開発コストの削減のための方法の1つとしてレビューに関するメトリクスデータの分析を行ってきた。この論文はそのような試みの1つを示すものである。

この論文では、ある企業から得られた実証的なデータ

を使用して、レビュー効率とレビュー指摘密度の関係を分析する。その上で、後工程での不具合が大きくなると予想したものとそうでないものとを分類する基準の決定方法を提案する。提案する方法のキーアイデアは、基準に関する2つのメトリクス「レビュー効率」と「レビュー指摘密度」を組み合わせて考えることである。そしてこの基準を用いることで、プロジェクトを3つのグループに分類する。

提案方法の有用性を示すために、実際のプロジェクトデータを使用して実証的な分析を行った。分析結果から、「レビュー効率に対してレビュー指摘密度が大きいプロジェクトでは、後工程での不具合発見率が大きくなる」、ということがわかった。また、この実験結果を実際の開発現場でどのように利用することができるかについても提案した。

2 背景と目的

今回対象とした企業では数年前にソフトウェア工学プロセスグループ(SEPG)が社内で設立された。SEPGはそれ以来、ソフトウェアプロセスの改善に取り組んできた。一般に、開発プロセスの後工程(テスト・デバッグ工程)で不具合を除去するコストは前工程(設計レビュー・コードレビュー工程)でのコストよりも高くなる。そのため、不具合があればできるだけ早い段階で除去することが強く推奨される。また、レビューによって発見された不具合の数はレビューの実施量が増えるほど大きくなる、と信じられていた。

この結果として、SEPGはレビューの実施量に関する基準を決定した。しかし、そのような基準は開発においてしばしば誤った決定をもたらすことがある。例えば、「レビュー工数は全設計工数の15%以上を割り当てるべきである」という基準が数年前に制定された。それ以来、平均のレビュー工数は年々増加してきて、現在はほとんどすべてのプロジェクトが全設計工数の15%を超えるレビュー工数を費やすようになっている。

しかし、レビュー工数は増えているにもかかわらず、レビューにより発見される不具合数はそれほど増えているとは言えないことがデータの分析から明らかになった。これは、レビュー工数単独による基準の難しさを示している。この問題を解決するためには、単独の指標による基準ではなく、複合的な指標が必要と考えられた。

そのためレビュー実施のための別の効果的な改善が必要となった。本研究で注目する新しいメトリクスとして、次の2つのメトリクスが挙げられる。

- <1> レビュー効率
- <2> レビュー指摘密度

これはSEPGがこの2つのメトリクスには何らかの関連性があると感じていたためである。しかし、同時にこれらのメトリクスは一概に基準値を決定すべきではないことも分かってきた。そして、これらのメトリクスに関して以下の仮説を考える。

- ・レビュー指摘密度がある条件を満たさないプロジェクトは、後工程での品質が悪くなる傾向がある。

この仮説を示すことを本研究の目的とするが、レビュー効率とレビュー指摘密度の関係を考慮することが重要であると考えたため、まずこれらの2つのメトリクスの関係性を調べる。

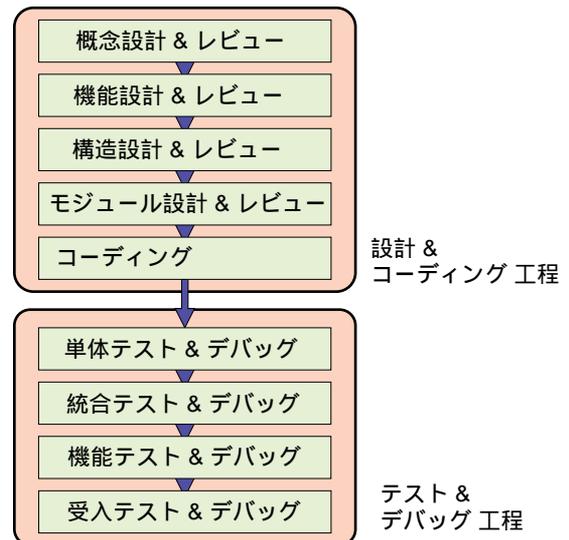


図1 ソフトウェアプロセス

- <2-3> 機能テスト & デバッグ
- <2-4> 受入テスト & デバッグ

3 メトリクスの詳細

3.1 ソフトウェアプロセス

図1にこの開発現場で採用されているソフトウェア開発プロセスモデルを示す。このプロセスモデルは標準的なウォーターフォールモデルである。厳密には、前工程への逆戻りや2つの工程の同時進行などの例外があるが、これらは図1には示していない。

本研究では開発プロセスを2つの連続した工程に分類する。

- <1> 設計・コーディング工程
- <2> テスト・デバッグ工程

設計・コーディング工程では、それぞれの設計・コーディング工程の直後にレビューが実行されるという点が大きな特徴である。設計工程はさらに5つの段階に細分化される。

- <1-1> 概念設計 & レビュー
- <1-2> 機能設計 & レビュー
- <1-3> 構造設計 & レビュー
- <1-4> モジュール設計 & レビュー
- <1-5> コーディング & レビュー

一方、テスト・デバッグ工程は4つの段階に細分化される。

- <2-1> 単体テスト & デバッグ
- <2-2> 統合テスト & デバッグ

3.2 メトリクスの定義

メトリクスは開発プロセスの状況を反映する指標である。この研究では、レビューの実施効率に関係のあるものを選択する。また、レビューについては設計・コーディング工程のうちコーディング段階のレビューを対象とする。

対象企業でのソフトウェア開発で直接使用されたメトリクスとして、以下のものがある。

- ・製品のサイズ (S) ... S は新規開発と修正されたコードの行数の合計と定義する。単位はKLOCである。
- ・レビュー工数 (E_R) ... E_R はレビュー実施にかかった工数と定義する。単位は人月である。
- ・レビュー工程で発見された不具合数 (D_R) ... D_R はコーディングのレビュー工程で発見された不具合数と定義する。なお、レビュー不具合には「誤字脱字」のような軽微なものは対象としていない。
- ・テスト・デバッグ工程で発見された不具合数 (D_T) ... D_T は4つのテスト・デバッグ工程で発見された不具合数の合計と定義する。ここでは、機能的な不具合・非機能的な不具合の両方が対象となっている。

これらのメトリクスを用いて、本研究で使用するためのメトリクスを定義する。

(1) レビュー効率 (ED_R)

ED_R は単位工数に対するレビュー実施量である。これ

は以下のように定義される。

$$ED_R = \frac{S}{E_R}$$

単位は「KLOC/人月」で表される。直感的に、このメトリクスは「レビュー実施の効率」を示している。 ED_R が小さくなるほど、レビューの効率がよいということである。これは同じ量のコードの場合、 ED_R が小さいほど多くの工数をかけているので、レビューが丁寧に行われていると考えられる。

(2) レビュー指摘密度 (DD_R)

DD_R は単位コード量あたりのレビュー実施によって発見された不具合の指摘密度である。これは以下のように定義される。

$$DD_R = \frac{D_R}{S}$$

単位は「個/KLOC」で表される。直観的に、 DD_R は「レビュー工程で発見された不具合がどれぐらいの密度で存在したか」を示している。

(3) デバッグ指摘密度 (DD_T)

DD_T は単位コード量あたりのテスト・デバッグ工程で発見された不具合の指摘密度である。これは以下のように定義される。

$$DD_T = \frac{D_T}{S}$$

単位は「個/KLOC」で表される。 DD_T が大きい場合必ずしも低品質というわけではないが、一般には大きな DD_T はテスト・デバッグ工程でより多くのコストが必要になると考えられる。この DD_T をレビュー以降の後工程での品質を表す指標として用いることにする。

4 レビュー効率とレビュー指摘密度の関係の分析

4.1 利用したプロジェクト

我々の研究で使用したプロジェクトは、この企業での組み込みソフトウェアシステムの開発に関連したものである。この開発現場では、数種類の類似したシステムが並行して開発されている。このため、類似の製品の典型的な開発では、共通して再利用されるコードがあり、開発プロセス自体も共通に近いものが使用されている。ま

た、開発者や開発プロセスは比較的長期に渡って変化の少ないものとなっている。

本研究のために準備されたのは、この開発現場において1999年より2003年の間に実施された772件のプロジェクトより収集されたメトリクスである。データには3.2節で説明したメトリクスの値が記録されている。分析の精度を保つため、このデータのうち有効なもののみを利用する。

まず、これらのデータに対して、レビュー指摘密度とデバッグ指摘密度が両方とも0であるデータは対象から除外した。この処理を行うとデータ数は666件となった。

次に、これらのデータには規模にばらつきがあったためその影響を取り除くために次の処理を行った。すなわち、666件のデータを、サイズ(S)によりソートし、その中央値から ± 250 個(84番目から583番目)のものを500件選択した。この処理によって規模が極端に小さすぎる、または大きすぎるプロジェクトを除去している。このようにして選択された500件のプロジェクトデータを以降の分析の対象とする。

4.2 相関性の分析

仮説を証明する前に、レビュー指摘密度とレビュー効率との間に相関があるかについて検証する。これは、3.2節で定義したメトリクスを用いると、「 DD_R と ED_R との間に相関がある」ことを述べている。選択した500件のプロジェクトに関して、レビュー指摘密度とレビュー効率との間にどのような関係があるかを調べるために、3.2節で定義したメトリクスを用いての関係図を作成した。この企業での ED_R と DD_R の関係を図2に示す。

この図における、 ED_R と DD_R の相関係数は-0.327であった。一般的に相関係数が-0.2~-0.4の間であれば、統計的には「弱い負の相関がある」といえるが、実践的な観点からはほとんど影響は無いと解釈できる。このことから、線形的な相関関係は認められないものの、これら2値の間に何らかの関連があることは確認できる。このような場合は、単に DD_R の基準を決定することができない。

5 レビュー指摘密度の分類基準による後工程での品質の分析

4.2節の検証結果を利用して、プロジェクトごとのレビ

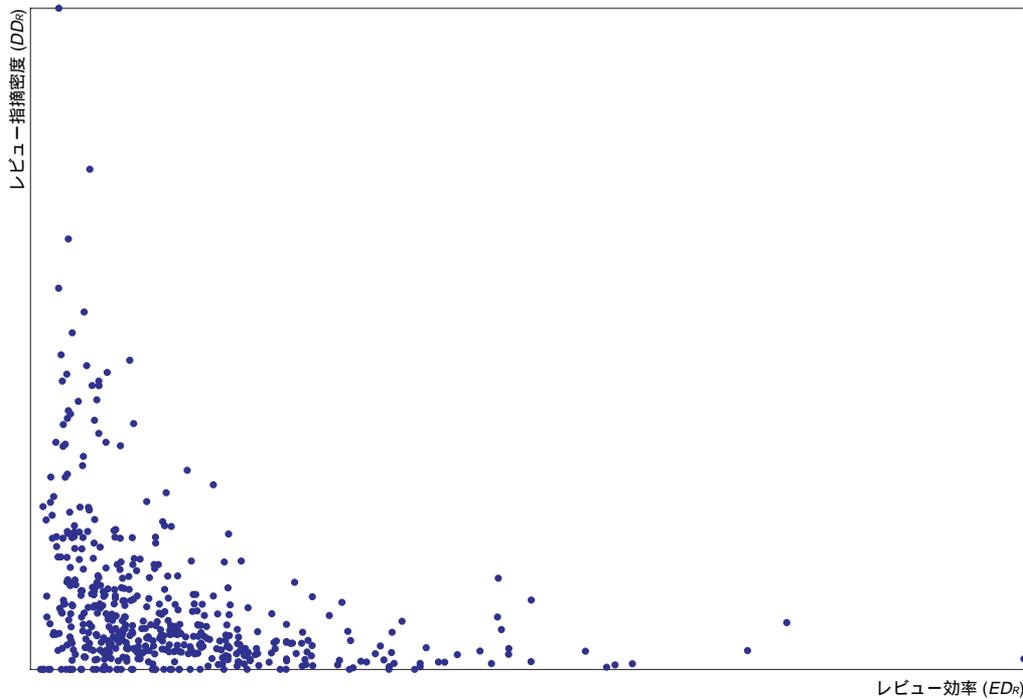


図2 レビュー効率(ED_R)とレビュー指摘密度(DD_R)の分布

レビュー効率に応じてレビュー指摘密度とデバッグ指摘密度に関する分析を行う。そのために、レビュー効率を考慮したレビュー指摘密度による分類基準を作る。そしてこれらの基準により分類されたグループ間で後工程での品質(デバッグ指摘密度)にどのような差が表れるかを検証する。分析のためのデータを提供した企業側では図2に対して、それぞれのレビュー効率に対して極端に高い、もしくは低いレビュー指摘密度の値を持つものは品質が良くないと感じていたため、この予想が正しいかどうかを検証する。

5.1 分析の手順

分析は以下の手順で行う。

最初にすべてのプロジェクトをED_Rの値によってソートし、それらをED_Rの値の小さい方からn個の分割に等分し、小さい方からG₁, G₂, ... という分割にし、最も大きな分割をG_nとする。

G_i (1 ≤ i ≤ n)のそれぞれの分割に対して、DD_Rの分布を得る。そしてDD_Rの値がその分割の上位10%¹にあたるものをG_i^{ex2}、下位10%にあたるものをG_i^{ex1}の値を持つものをG_i^{ex1}、どちらにも入らないものをG_i^{no}として、3つのグループに分類する。このとき同じDD_Rの値を持つ

ものが下位10%である数よりも多く存在する場合は、これらはすべてG_i^{ex1}に分類するものとする。そのため、実際には10%を超える数が分類されることがある。

例として、図3のプロジェクトを分類する例(データ数40, n = 4の場合)を図4に示す。それぞれのグループG_i(各グループ10個ずつ)に対し、DD_Rの値がそのグループの上位10%にあたる数(1個)をG_i^{ex2}とし、同様に下位10%である1個をG_i^{ex1}となるように分類する。そしてどちらにも入らない残りのものをG_i^{no}とする。ただしG₃, G₄のように、10%を超える数が分類されることがある。

全プロジェクトを3つのグループに分類するため、G^{no}, G^{ex1}, G^{ex2}を以下のように定義する。

$$G^{no} = \bigcup_{i=1}^n G_i^{no}$$

$$G^{ex1} = \bigcup_{i=1}^n G_i^{ex1}$$

$$G^{ex2} = \bigcup_{i=1}^n G_i^{ex2}$$

最後に、これらのグループの分類方法が有効なものであるかを検証する。具体的には、G^{no}とG^{ex1}の場合とG^{no}とG^{ex2}の場合のそれぞれについて、DD_Rの値がG^{no}の方が統計的に小さいといえるかどうかを、マンホイットニーのU検定によって示す。

1 正規分布を用いた外れ値の検出手法では、平均値(m) + 2(σ)の範囲に入らないものを外れ値と定義することが多くある。この場合、外れ値となる部分は4.6%程度となるが、本研究では正規分布を仮定しないデータで分析を行ったため、順位による外れ値を考えた。その際、正規分布での外れ値範囲を参考にしつつ、もう少し多くのプロジェクトが含まれるように範囲を広げ、10%とした。

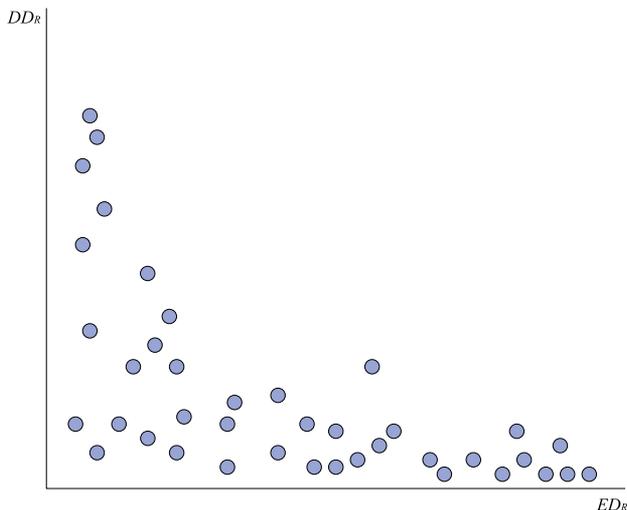


図3 ED_RとDD_Rの例

5.2 仮説の検証

ここでは、第2節で述べた以下の仮説について検証する。

仮説 レビュー指摘密度がある条件を満たさないプロジェクトは、後工程での品質が悪くなる傾向がある。

この仮説は、3.2節で定義したメトリクスを用いると、「DD_Rの値がある基準を満たさなければ、DD_Rの値は大きくなる傾向がある」ことを述べている。

図5に500件のプロジェクトのレビュー効率とレビュー指摘密度をプロットしたものを示す。なお、図5ではそれぞれのメトリクスの最大値を1として正規化したものを示している。図5において、それぞれのプロットはプロジェクトを表す。下側の線はG^{ex1}とG^{no}の境界を表し、上側の線はG^{no}とG^{ex2}の境界を表す。

5.1項で示した手順によって、500件のプロジェクトをG^{no}、G^{ex1}、G^{ex2}の3つのグループに分類する。この研究ではnの値として10を使用した。

表1に各グループに分類されたプロジェクトの個数と、各グループのデバッグ指摘密度(DD_R)の中央値を示す。ここで、DD_Rの中央値はG^{no}の値が1となるように正規化したものを示している。これは例えば、G^{ex2}のグループに分類されたプロジェクトは全部で50件であり、DD_Rの中央値はG^{no}の1.326倍であることを表している。

また、G^{no}とG^{ex1}、G^{ex2}のそれぞれの間でのDD_Rの中央値の差が統計的に有意であるかをマンホイットニーのU検定により調べた。以下にその結果のp値を示す。

・ G^{no}とG^{ex1} : 0.459

・ G^{no}とG^{ex2} : 0.0368

この結果から、G^{no}とG^{ex1}の間にはDD_Rの値に有意な差

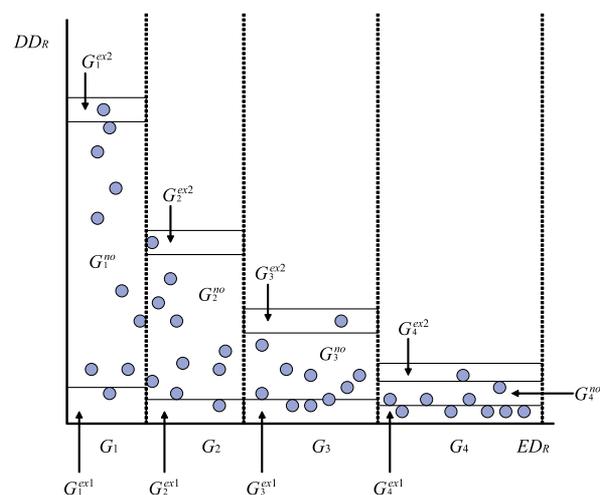


図4 ED_Rの順によるプロジェクトの分類

が見られなかったが、G^{no}とG^{ex2}の間には有意水準5%で差が認められることがわかった。よって、仮説は成り立つことが確認された。

5.3 考察

以上の結果から、次のことがわかる。

- ・ レビュー効率に対してレビュー指摘密度が大きい場合は、そうでない場合と比べてデバッグ指摘密度が大きくなる傾向がある。すなわち、後工程での品質が悪くなる傾向がある。
- ・ レビュー効率に対してレビュー指摘密度が小さい場合は、そうでない場合と比べてデバッグ指摘密度が小さくなる傾向があるが、統計的に有意な差ではない。

このことから、グラフの上位10%の線よりも上にあるプロジェクトは、後工程での不具合が他よりも多くなる傾向があった。これは統計的に有意な差であるので、このような基準の決定方法は有効であるということがわかった。上位10%の線よりも上に分類されたプロジェクトは、レビュー効率を下げることによって、より丁寧なレビューを行うことができ、後工程での不具合数を減らすことにつながると考えられる。

一方、グラフの下部10%の線よりも下にあるプロジェクトは、有意な差ではないが後工程での不具合数が少なくなる傾向があった。これは、元の品質が良かったためではないかと考えられる。

6 現場への適用法の提案

これまでの実験結果から、実際の開発現場においてど

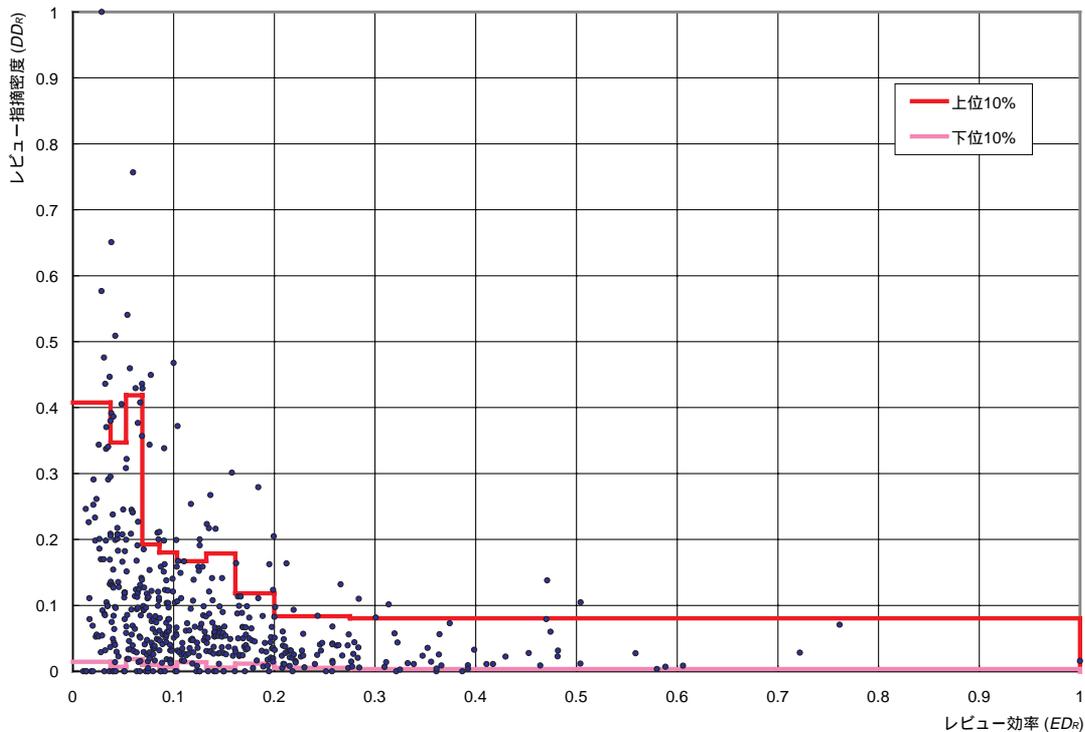


図5 分類基準を表す境界線を示した関係図

表1 各グループの個数とデバッグ指摘密度(DD_T)の正規化した中央値

	個数	中央値
G^{no}	397	1
G^{ex1}	53	0.937
G^{ex2}	50	1.326

のように活用することができるかについて提案する。

このために、5.1節で作成したグループに分類するための基準の境界線を、最小二乗法の累乗近似によって、より一般的な曲線で表す。この累乗近似曲線を追加した関係図を図6に示す。図6において、上位10%線の累乗近似の曲線よりも上に位置するプロジェクト(例えば、図中の×印の点)がレビュー指摘密度の条件を満たさないプロジェクトであると考えられる。

仮に、×印のレビュー効率 ED_r が0.12でレビュー指摘密度 DD_r が0.2であるとする。このような場合、レビュー効率を0.06まで下げると(図中○印)、上位10%線の累乗近似の曲線よりも下に位置することになり、これはレビュー指摘密度の条件を満たすプロジェクトとなる。適切なレビュー効率を検出する基準としてこのような方法を使用すると効果があるのではないかと考えられる。

ここまでは、今回研究対象とした企業のデータを用いた場合について説明したが、これ以外の場合でも3.2節のようなメトリクスを定義し、5.1節のような方法で基準を

作り、統計的に有意な差が表れるのであれば、このような提案法は有効ではないかと考えられる。具体的な適用方法の提案として次のようなものを考える。

まず、これまでのプロジェクトのデータを使用して、それぞれの現場におけるレビュー指摘密度の条件を作成する。そして新規に開発するプロジェクトにおいては、レビュー完了時にこの条件を適用して、レビューの現状を判断する。

あるプロジェクトがレビュー指摘密度の条件を満たさないと判断された時の現場での対応について、より詳細に考察してみる。ここでは、 ED_r の値と DD_r の値によって4つの場合を考える。

- ED_r が小さく、 DD_r の値も小さい場合：おそらくコードの品質は良いものと考えられるため、ほぼ現状のまま開発を進めればよいと判断できる。そもそも、この場合にレビュー指摘密度の条件を満たさない可能性は極めて低い。
- ED_r が小さく、 DD_r の値が大きい場合：作成されたコードの品質に何らかの問題があると考えられる。そこで、レビューの結果を有効に活用し、コードを適切に改善するような方針を立てるべきである。
- ED_r が大きく、 DD_r の値も大きい場合：少ししかレビューをしていないのに多くの不具合が発見されたとい

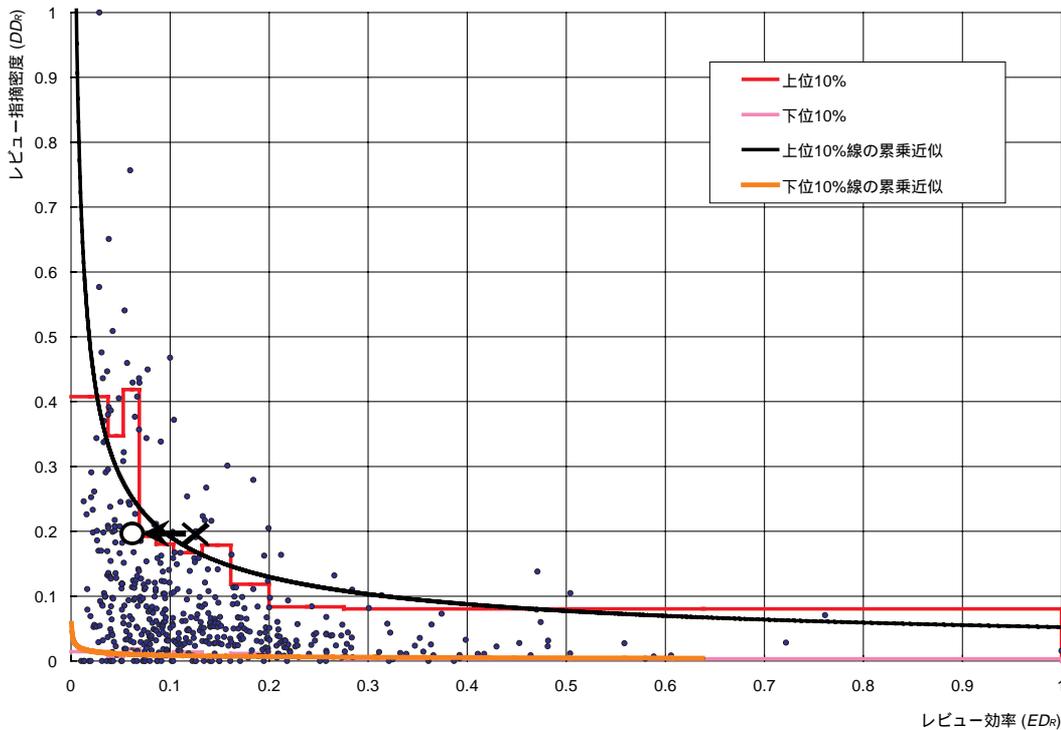


図6 累乗近似による境界線を示した関係図

うことは多くの不具合が残存していることを暗示している。コーディング工程を見直し、再度コーディングを行った後にレビューを実施することが望ましい。

- ・ ED_R が大きく、 DD_R の値が小さい場合：判断が難しいケースであるが、 ED_R を下げることによって DD_R がどのように変化するかを調べつつ、レビュー終了の判断を行うべきだと考える。

7 まとめ

この論文では、レビュー効率を考慮したレビュー指摘密度と後工程での品質に関する分析を行った。ここで注目したメトリクスとして、SEPG が関連性があると感じていた2つのメトリクス（レビュー効率とレビュー指摘密度）を利用した。

そしてこれらのメトリクスに関連した仮説「レビュー指摘密度がある条件を満たさないプロジェクトは、後工程での品質が悪くなる傾向がある」を提示した。

この仮説の妥当性を示すために、実際に企業の開発現場で使用された500プロジェクトのデータを使用して統計的分析を行った。その結果は、レビュー効率に対してレビュー指摘密度が大きいプロジェクトは、後工程での不具合発見密度が大きくなった、というものとなった。よって、この仮説の正しさを確認できた。そのようなプ

ロジェクトをテスト・デバッグ工程の前に検出し、適切なレビュー効率に下げることにより、残存不具合数を減らすことが期待できるのではないかと考えられる。実際のプロジェクトへの適用法の例も示したが、現場への適用法については今後の課題である。

参考文献

- [BRIAND1997] L. C. Briand, K. E. Emam, B. Freimut, and O. Laitenberger : Quantitative evaluation of capture-recapture models to control software inspections. In Proc. of 8th International Symposium on Software Reliability Engineering, pp. 234-244, 1997
- [BRIAND1998] L. C. Briand, K. E. Emam, O. Laitenberger, and T. Fussbroich : Using simulation to build inspection efficiency benchmarks for development projects. In Proc. of 20th International Conference on Software Engineering, pp. 340-349, 1998
- [CIMITILE1994] A. Cimitile and G. Visaggio: A formalism for structured planning of a software project. Journal of Software Engineering and Knowledge Engineering, 4(2):277-300, 1994
- [EBENAU1993] R. G. Ebenau and S. H. Strauss: Software Inspection Process. McGraw-Hill, 1993
- [FAGAN1986] M. E. Fagan: Advances in software inspections. IEEE Trans. on Software Engineering, 12(7):744-751, 1986
- [HUMPHREY1989] W.S. Humphrey: Managing the Software Process. Addison-Wesley, MA, 1989
- [HUMPHREY1991] W. S. Humphrey, T. Snyder, and R. Willis: Software process improvement at hughes aircraft. IEEE Software, 8(4):11-23, 1991
- [KUSUMOTO1993] S. Kusumoto: Quantitative evaluation of software reviews and testing process. PhD thesis, Osaka University, 1993
- [PODOROZHNY1997] R. M. Podorozhny and L. J. Osterweil: The criticality of modeling formalisms in software design method comparison. In Proc. of 19th International Conference on Software Engineering (ICSE'97), pp. 303-313, 1997
- [RUSSELL1991] G. W. Russell: Experience with inspection in ultralarge-scale developments. IEEE Software, 8(1):25-31, 1991
- [TANAKA1995] T. Tanaka, K. Sakamoto, S. Kusumoto, K. Matsumoto, and T. Kikuno: Improvement of software process by process description and benet estimation. In Proc. of 17th International Conference on Software Engineering, pp. 123-132, 1995
- [WELLER1993] E.F. Weller: Lessons from three year of inspection data. IEEE Software, 10(5):38-45, 1993

Java 開発者の オンデマンド・ラーニングを支援する ソシオテクニカル環境



葉 雲文†



山本 恭裕††

Java 開発において大規模なクラスライブラリの利用は不可欠である。開発者がそれらのライブラリをいかにして習得するかは大きな課題である。本稿では、Java 開発者がクラスライブラリをオンデマンド的に習得するためのソシオテクニカルな支援環境を提供することを目的として、開発者同士による知識交換の場としてのダイナミックコミュニティの形成を、エキスパート同定および選定のプロセスを通して行う仕組みを提案し、その仕組みを実装した STeP_IN システムを報告する。

A Socio-Technical Environment in Support of On-Demand Learning for Java Developers

Yunwen Ye †, Yasuhiro Yamamoto ††

The quality and efficiency of Java development rests on the effective use of its large API class libraries. The huge number of available class libraries presents a great learning challenge for Java developers. This paper describes a socio-technical approach to supporting Java developers to learn class libraries on demand. This approach provides by providing a collaborative learning platform for Java developers to exchange knowledge by identifying and selecting knowledgeable peer experts.

1 はじめに

Java 開発における大規模クラスライブラリの利用は、生産性の向上と品質の改善に大きく貢献するとされている [LINDEN1998]。これらのクラスライブラリを再利用することにより開発に必要な労力が軽減される一方で、開発者がそのようなクラスライブラリをいかにして習得するかは大きな課題である。

クラスやメソッドといったライブラリ部品の新たな習得は、ドキュメントを調べることから始まる。ドキュメントの多くは部品開発者により書かれたもので、部品利用者の立場から見ると抽象的で理解し難いことが多く見

られる [BASSET1996]。ドキュメントを読んでわからない場合、当該部品に詳しいエキスパートに聞くことは早道とされている [BERLIN1993]。つまり他の開発者との知識交換を通して部品の利用方法を習得する。

しかし、エキスパートに尋ねる方法では、誰がエキスパートであるかを確定するのは困難であり、さらに、エキスパートを確定してもそのエキスパートがタイムリーに応じてくれるかは保証されない。

そこで我々は、Java 開発者がクラスライブラリをオンデマンド的に学習する際の基盤環境を提供することを目的として、STeP_IN (Socio-Technical Platform for In-situ Networking) 環境を開発した。STeP_IN は、技術的及び社会的な側面の双方の要素を統合的に考慮したソシオテク

† 株式会社 S R A 先端技術研究所, SRA Key Technology Laboratory, Inc

†† 東京大学先端科学技術研究センター, RCAST, University of Tokyo

ニカル環境[MUMFORD1987]であり、開発者同士の活発な知識交換によってクラスライブラリの習得を支援するものである。

STeP_IN環境では、Java開発者が個人適応型ライブラリ検索機能を利用して必要となるライブラリ部品を検索し、システムに格納されているその部品の使用例から利用方法を習得することができる[YE2006]。ドキュメントと使用例からだけでは習得できない場合、その部品のエキスパートに質問することができる。

開発者Aが部品に関して質問すると、STeP_INシステムは、各開発者の知識モデルと開発者間のソーシャルネットワークの分析に基づいて、に関しての専門知識を提供することができ、かつ、Aと良好な社会的な関係を有し積極的に情報提供する可能性が高い、と考えられるエキスパートを選択し、知識交換の場としてのダイナミックコミュニティ(DynC(A,))[YE2004]を形成する。選ばれたDynC(A,)のメンバに対しAからの質問を送信し、開発者同士の相互支援によりライブラリの習得を支援する。

以下では、第2節で、STeP_INシステムの概要と利用方法について述べる。第3節では、STeP_INでダイナミックコミュニティを形成するプロセスを説明する。第4節では、持続的な知識交換を支援するためのソーシャルウェアネスコミュニケーションの提案を行う。第5節では、STeP_INシステムの有効性を評価するシミュレーション

実験について報告する。第6節では、関連研究との比較を行い、最後に第7節で本稿をまとめる。

2 STeP_INシステム

2.1 システム構成

STeP_INシステムは、図1に示すように、リポジトリ、プロファイルマネジメント、サーチエンジン、知識交換支援、という4つのサブシステムから構成される。

リポジトリには、クラスライブラリ部品検索のためのインデックス、部品に関するドキュメント、及びこれまでに開発者同士の間で行われたディスカッションのアーカイブが格納されている。また、各開発者にとっての既知の部品をモデリングしたテクニカルプロファイル、及び、開発者同士の社会的な関係をモデリングしたソーシャルプロファイルも格納されている。

システム利用者となる開発者は、プロファイルマネジメントを通して、自分のテクニカルプロファイルとソーシャルプロファイルの初期化及び編集ができる。サーチエンジンは、部品、ドキュメント、使用例、及びディスカッションアーカイブを検索する機能を提供する。知識交換支援では、あるユーザがある部品に関して質問した際に、各ユーザのテクニカルプロファイルとソーシャルプロファイルに基づき、当該ユーザのために有益な情報

提供を行う可能性の高いメンバを選び、知識交換の場としてのダイナミックコミュニティを形成する。

2.2 リポジトリの初期化

STeP_INのサーチエンジンはCodeBroker[YE2005]をベースにしている。検索インデックスはライブラリのソースコードから作成される。ドキュメントは拡張したJavadocを利用してソースコードから生成される。

テクニカルプロファイルとソーシャルプロファイルの初期化は各ユーザにより行われる。サーバでのユーザ登録の後、ユー

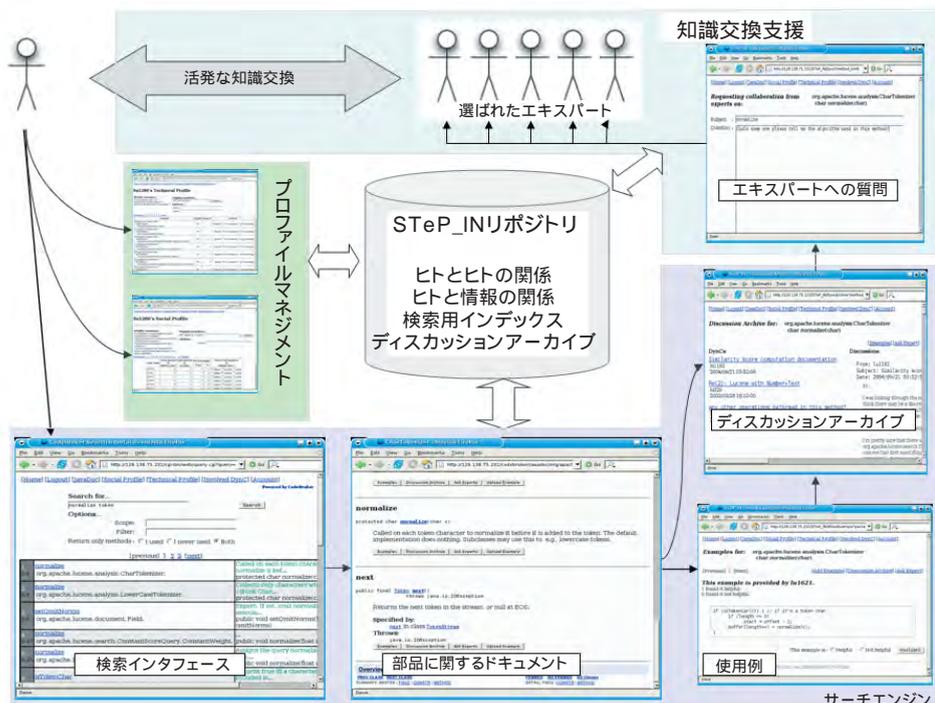


図1 STeP_INシステムの構成図

ザはサーバからプロファイルをダウンロードしてテクニカルプロファイルとソーシャルプロファイルの初期化を行う。プロファイルは、ユーザがこれまでに開発したJavaプログラムを解析し、利用したことのある部品（メソッド）とその利用回数を抽出する。抽出結果は、当該ユーザの初期テクニカルファイルの元情報としてSTeP_INシステムにアップロードされる。

初期ソーシャルプロファイルの元となる情報は、ソーシャルプロファイルが、ユーザの指定したメールボックスのヘッダ部分を解析して作成する。具体的には、当該ユーザへのメールの送信元メールアドレス及び送信回数を抽出し、STeP_INサーバにアップロードし、当該ユーザの初期ソーシャルプロファイルとする。

2.3 システムの機能と利用

ユーザは、下記のような流れでSTeP_INシステムを利用し、新しい部品についての知識を習得する。

検索インタフェースにおいて検索要求を自然言語で入力すると、部品の検索が行われる(図1)。検索結果リストにおいて部品名をクリックすると、その部品のドキュメントが表示される(図2)。このドキュメントは、既存のJavadocを拡張したもので、Examples, Discussion Archive, Ask Experts, Upload Examples という4つのボタンが追加されている。

Examplesボタンからは当該部品を使用したプログラムの例を、Discussion Archiveからは当該部品に関してそれまでに行われた知識交換のメッセージを、閲覧することができる(図1)。

Ask Expertsからは、質問を入力するための画面へと遷移する(図1)。ここで質問の内容を入力してSubmitすると、当該ユーザ(DynC 開始者と呼ぶ)のための、当該部品に関するDynC が形成され、選ばれたエキスパート(DynC 参加者と呼ぶ)に質問テキストが電子メールで送られる。



図2 機能拡張されたJavadocドキュメント

メールを受信したDynC参加者が送る返信メールは、STeP_INサーバを介して当該DynC の参加者全員に送られる。返信メールを出したDynC 参加者をDynC貢献者と呼ぶ。DynC 開始者は、自らの開始した当該DynC をHelpfulあるいはNot Helpful と評価することができ、評価した時点で当該DynC は解散される。なお、開始から解散までのDynC で行われたメール交換は、ディスカッションアーカイブに1つのスレッドとして蓄積される。

3 DynC形成プロセス

特定のユーザAのために部品 に関する知識交換の場としてのDynC は、エキスパート同定及びエキスパート選定というプロセスを経てその参加者が決定される。エキスパート同定では、部品 に関する専門知識を有する人が抽出され。エキスパート選定では、同定されたエキスパートの中からAと良好な社会的な関係を有する人が選択される。

3.1 エキスパート同定

STeP_INは各ユーザにとっての既知の部品をモデリングしたテクニカルプロファイルに基づいて、部品 に関するエキスパートを同定する。テクニカルプロファイルでは、部品 に関する知識についての度合いが、確定エキスパート、申告エキスパート、推定エキスパート、未定エキスパート、という4つのレベルに分けられている。

部品 に関する確定エキスパートとなるのは、過去に部品 に関して形成され、開始者によってHelpfulと評価されたDynC において返信メールを送信した人(DynC 貢献者)である。

申告エキスパートは、部品 を知っているとして自己申告した人である。STeP_INでは、各ユーザがプロファイル管理機能を通して、図3で示すように、個々の部品に対して、Declare(申告)の欄においてExpert(専門)、Not Declared(未申告)、No Knowledge(非専門)から1つを選択できる。初期値はNot Declaredであるが、ここでExpert を選択すると当該部品の申告エキスパートとなる。

推定エキスパートは、部品 を使ったことがあり(上記テクニカルプロファイル(図3)でのUsage # (部品利用回数)が1以上の値)、かつ、テクニカルプロファイル

のDeclareの欄でNo Knowledgeを選ばなかった人である。

未来エキスパートとは、部品 に関して、過去に質問し1つのDynCの開始者となったものの、当該DynCから有益な知識を得られずNot Helpfulと評価したDynC開始者である。このような人を未来エキスパートとしてエキスパート同定の対象とするのには2つの理由がある。第一に、 に関して質問したという事実から、 を習得する意欲があると推測できる。 に関して新たに形成されるDynCに参加することは、新たな学習機会であり、有益な知識交換ができる可能性があるといえる。第二の理由は、その後の学習の可能性である。過去に に関して質問した時点では に関して知識が無かったものの、 を利用する需要があったという事実から、当該時点までに他の何らかの手段で習得した可能性が十分あり得ると考えた。

部品 に関するエキスパートの同定プロセスでは、リポジトリのテクニカルプロファイルにおいて部品 とのリレーションがあるエキスパートを、知識レベルの高い順、すなわち、確定エキスパート、申告エキスパート、推定エキスパート、未来エキスパートの順で並べ、エキスパート候補リストを作成する。

3.2 エキスパート選定

エキスパート選定は、同定プロセスで作成したエキスパート候補リストから、当該DynCの開始者Aとの社会的な関係に基づき、DynC参加者を選び、DynC(A,)を形成する。

(1) ソーシャルプロファイルのモデリング

ソーシャルプロファイルはユーザ間の社会的な関係を

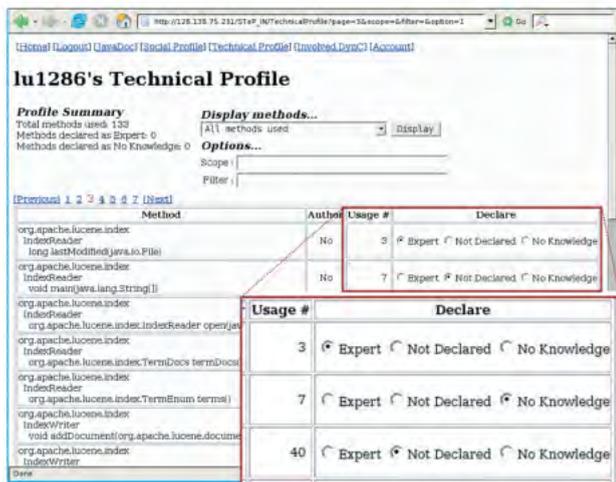


図3 テクニカルプロファイル編集画面

モデル化している。任意の2人のユーザAとBの間に、exclude<A, B>, friend<A, B>, help<A, B>, email<A, B>という4種類のリレーションが定義される。

Aが自分のソーシャルプロファイル編集で、Bに対してのFuture Participation in His/Her DynC(当該ユーザが開始したDynCへの今後の参加)欄においてneverを選択すると(図4)、exclude<A, B>が成立する。これは、Bの開始したDynCにAが参加したくない旨となる。逆に、Aが自分のソーシャルプロファイルに、Bに対してのFuture Participation in His/Her DynC欄においてalwaysを選ぶと、friend<A, B>が成立する。これは、AがBの開始したDynCに参加したい旨となる。exclude<A, B>とfriend<A, B>とは同時には成立しない。

help<A, B>リレーションはSTeP_INシステム内での活動においてAがBを助けた回数を表すもので、excludeやfriendと違って、ユーザが変更するものではない。Bの開始したDynCにAが返信メールを送ると、AがBを一回助けたと計上される。これは、AがBを助けようとした努力を考慮するもので、当該DynCの最終評価結果がHelpfulかNot Helpfulかに関わらずに計上される。なお、Helpfulと評価された場合であれば、Aはその部品に関する確定エキスパートに認定され、Not Helpfulと評価された場合は、当該部品に関してのAのエキスパートレベルは変化しない。

email<A, B>は、AがSTeP_INシステムの外でBに送った電子メールの個数を示す。これは、Bが初期ソーシャルプロファイルを作成する時点で、メールボックスの解析により得られるデータである。

(2) エキスパート選定の手順

エキスパート選定のプロセスは下記の6つのパスからなる。各パスにおいては、エキスパート候補リストに含まれるエキスパート(Xと表記する)全員について、前述の4つの知識レベルの高い方から順にAとの社会的な関係を調べ、エキスパート候補リストから除外するか、DynC(A,)の参加者とするかが決定されていく。参加者が一定の人数に達した時点で、選定のプロセスは終了し、残るパスは無視される。この参加者人数は、実際の運用環境において決定されるべきものであるが、現状の実装でのデフォルト値は5人としている。

パス1では、不参加者が除外される。exclude<X, A>が成立した場合に、当該Xをエキスパート候補リストから

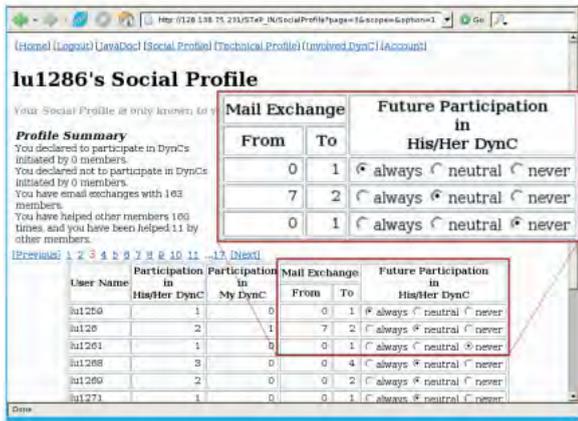


図4 ソーシャルプロフィール編集画面

除外する。これは、Aの開始したDynCにXは参加したくないとの申告を意味するので、XをDynC(A,)に選んだとしても、知識交換に参加する可能性は非常に低いと考えられるからである。

パス2では、友好関係を基準に選ぶ。friend<X, A>が成立した場合、XをDynC(A,)に入れる。この時点でfriend<X, A>のリレーションを持つ人数が5人を超える場合には、ランダムに選び、エキスパートの選定を終える。5人に満たない場合は、friend<X, A>のリレーションを持つX全員を選んで次のパス3に進む。

パス3では、個人間の相互支援の履歴で選ぶ。ソーシャルキャピタルの理論によれば、AがXを助けることは、Aに将来の期待を、Xには将来の義務を生じさせる[COLEMAN1988]。STeP_INでは、help(A,X)をAがXに対して作った貸し(credit)、help(X,A)をXがAに対して作った貸しとみなす。help(A,X)がhelp(X,A)より大きい場合、XはAを助けた回数よりAに助けられた回数が多い。これを、XがAを助ける義務として解釈し、Xをこのパスで選ぶ。

パス4では、個人間インタラクションの発生時点を考慮して選ぶ。このパスの段階では、選定されずにエキスパート候補リストに残ったエキスパートは、Aを助けた回数がAに助けられた回数より多い。ここでは、記憶が新しいものであるほど印象がより強いものとし、より近い過去にAに助けられたXをDynC(A,)に選ぶ。エキスパート候補リスト内に残っている個々のエキスパートXについて、Aが最後にXを助けた時間が、Xが最後にAを助けた時間より後であれば、当該XをDynC(A,)に選ぶ。このパスで選んだ人数と上のパスまでで選んだ人数との合計が5人を超えた場合、時間の新しい順で5人

になるまで選んで、エキスパートの選定を終える。そうでなければ、全員を選んで次のパス5に進む。

パス5では、XとSTeP_INにおける他のすべてのユーザとの相互支援の履歴から選ぶ。XがSTeP_IN全体で他者を助けた回数が他者に助けられた回数より少ない場合に当該Xを選ぶ。このような人の数が上のパスで選んだ人数と合わせて5人を超えた場合、助けられた回数順で5人まで選んで、エキスパートの選定を終える。そうでなければ、全員を選んで次のパス6に進む。

パス6では、STeP_INシステムの外でのインタラクションを考慮して選ぶ。email<X, A>の高い順でエキスパートを5人まで選び、DynC(A,)に入れる。XがAにメールを送った回数が多いほど、XがAをより知っていると推定でき、Aを助ける可能性が高いと考えられるからである。

上記の6つのパスでエキスパートを選定するプロセスでは、既存の良好な社会的関係が更なる協力的行動への動機づけに繋がるというソーシャルキャピタルの理論[COLEMAN1988]に基づき、Aのために知識交換に参加する動機をより強く持っているエキスパートを当該DynCの参加者とする。まず個人間の友好関係をベースとし、次にSTeP_INにおけるインタラクション履歴から生じた期待と義務の関係を考慮する。最終的にはSTeP_INシステムの外で生じた一般的なソーシャルインタラクションによって形成された既存のソーシャルネットワークの分析結果に基づいて選定する。

4 ソーシャルアウェアネス コミュニケーション

これまでの知識交換に関する研究は、知識を求める側の立場から、適切な知識提供者を発見し知識交換を可能にすることに注目しているものが多い[MOCKUS2002]。知識提供者側にかかる負担や知識提供者にとってのメリットなどについての考慮が十分ではない[NAKANOKOJI2006]。また、それらのアプローチの多くは一回の知識交換を成功させることのみを重視している。本研究では、長期間にわたって持続的な知識交換を可能にすることを最優先して、1つ1つの知識交換の成功のみではなく、知識提供者の長期的な参加動機を損なわないことを目指して、社会的な行動規範に則したソーシャルアウェアネスコミュニケーションのためのメカニズムを提案する。

4.1 不参加の自由を与えるメカニズム

STeP_INでは、状況と理由を想定し、DynC への参加を他のユーザに知られることなく自由に拒否することができるようにしている。

特定の相手を助けたくない場合であれば、自分のソーシャルプロフィール(図4)において、その相手に対してのFuture Participation in His/Her DynC 欄でneverを選択することにより、3.2.2項でのパス1の段階で自分が相手のDynCに選ばれないことが保証される。

また、同じ部品に関する質問に頻繁に答えることが嫌になったなどの理由で、特定のトピック(部品)に関するDynCに参加したくないのであれば、自分のテクニカルプロフィール(図3)の当該部品に関するDeclare 欄でNo Knowledge を選択することにより、エキスパート候補リストに入らないことになり、当該部品に関するDynCに選ばれることを避けることができる。

さらに、DynCに参加していたものの時間的な余裕がなくなったなどの理由で、参加中のDynCから抜け出ることもできる。これは、STeP_INにログイン後の、Involved DynC(関わっているDynC)に示されるOthers' active DynC you are participating(他のユーザが開始したDynC)のリスト(図5)においてLeave this DynC(当該DynCを離脱)をチェックすることによって行われる。それ以降の当該DynCにおけるメール送信を遮断することができる。

これらのメカニズムは一見、活発な知識交換の支援という目的に反するものにも思われるが、長い目で見ると、参加意欲の高くない知識交換を避けることができ、より活発で効果的な知識交換の促進につながると考えられる。また、知識提供者に不参加の自由があるので、知識を求める側にとっても、エキスパートを邪魔するかもしれないという心配を軽減した上で気軽に質問することができる。



DynC ID	Subject	Method	Start Date	Initiated by	Leave this DynC
2540	SUBJECT	Document void setBoost(float)	2006/09/28 11:55:58	lu1777	<input type="checkbox"/> Leave
2535	SUBJECT	Hits int length()	2006/09/27 03:07:30	lu1856	<input type="checkbox"/> Leave

図5 参加中のDynCにおけるメールの遮断

ると考えている。

4.2 開示情報の非対称性

日常の社会生活では、人に頼まれると、不本意でも断り切れないケースが多かれ少なかれ存在する。断ることによってその人との関係を損ねたり、自分の社会的な評判も落ちかねない恐れがあったりするからである。

他者のための知識交換への参加を拒否する際にそのような事態が生じることは望ましくないと我々は考えている。一方で、他者を助けた人にとっては、自らの社会的な貢献に応じて、それに相応しい社会的な評価を得られなければ、更なる貢献する動機が損なわれかねない。STeP_INでは、参加拒否を秘匿すると同時に、貢献を周知するための、非対称的な情報開示[NAKANOKOJI2006]の仕組みを実装している。

DynCが形成され、選ばれた参加者に質問メールが送られる際、個々の参加者は、自分が参加していることと質問者が誰であるかだけを知ることができ、他に誰が選ばれたかはわからない。開始者自身も誰が自分のDynCに参加しているかを知ることができない。これによって、参加者が貢献しない(返信しない)でいることも途中で抜け出てしまうことも他のメンバに知られることがなくなる。参加者が質問に対して返信した際には、その参加者のUser Nameの入った返事が当該DynC全員に届くので、当該参加者が貢献していることを当該DynCメンバ全員が知ることができる。

5 システム評価

STeP_INシステムにおけるエキスパート同定および選定の手法の有効性を検証するためにシミュレーションにより評価実験を行った。

実験に使用したデータは、情報検索システムluceneのJavaライブラリ、および、Java開発者のメーリングリストjava-user@lucene.apache.orgのアーカイブである。ここではlucenceライブラリの使用法に関して知識交換が行われている。このメーリングリストの2001年から2006年8月までのアーカイブを利用して、STeP_INのシミュレーション運用実験を行った。

以下にシミュレーション手順を示す。

luceneのJavaライブラリのソースコードをインデク

シングし、拡張Javadoc を生成する。

メーリングリストにメールを送信したメンバの全員をSTeP_INにユーザとして登録する。

各メンバのテクニカルプロフィールを初期化する：これは、各メンバが実際に開発したJava プログラムを入手することが不可能であるので、本実験では、2001 年から2006 年8 月までの全メールを解析して、メール本文中にlucene ライブラリのメソッドが含まれている場合に、そのメール送信者がそのメソッドを一回使ったとみなし、これを計上してテクニカルプロフィールとする。

各メンバのソーシャルプロフィールを初期化する：2001 年から2005 年までのメーリングリストのアーカイブのスレッドごとに分け、メンバAが開始したスレッドにBが返信すると、email(B, A)とシミュレートする。

過去に形成されたDynC をシミュレートする：2001 年から2005年までのメーリングリストにあったスレッドの内容を分析して、中にlucene ライブラリのメソッドについて議論がある場合、そのスレッドを一個のDynCとみなす。スレッド中に出現するメソッドの中から出現回数が一番多いメソッドを当該DynC のトピックとし、スレッドを始めたメンバをDynC 開始者とする。そのスレッドに返信した他のメンバをDynC 貢献者とする。これにより、メンバ間の5 年間分のhelp 関係として計算する。

2006 年のメーリングリストのスレッドに基づいて、を形成し、これを実際のメーリングリストでのスレッドと比較する：2006年にメーリングリストに投稿された質問のメールを読み、その質問が特定のライブラリメソッドに関するものであった場合、送信したメンバとしてSTeP_INシステムにログインし当該メンバが開始者となったDynC を形成する。DynC に選ばれたメンバと、実際にメーリングリストで当該質問に返信したメンバとを比較する。

表1は、2006年に投稿されたメールからライブラリに関する質問となっているものを時間順に20個抽出して、STeP_INがDynC 参加者としたユーザと実際の返信者との比較である。このうち、各質問者(コラム2)が質問しているライブラリ部品(コラム3)について、実際のメーリングリスト上で答えたメンバ(コラム4)が、STeP_INで当該質問者のために当該部品に関して形成し

たDynC に選定したメンバに完全に包含されたケースは5 個となった。部分一致しているケースは9個、完全に不一致となったものは6個であった。STeP_INは、70%のケースで、答えてくれるエキスパートを部分的には選定したことになる。各メンバのテクニカルプロフィールおよびソーシャルプロフィールのモデリングが不完全であることを考慮すれば、今回の評価実験によってSTeP_INにおけるエキスパートの同定と選定の手法について一定の有効性を示すことができたと考えられる。

6 関連研究と議論

知識交換を支援する研究およびツール開発はこれまで数多くなされている。現状では、メーリングリストやニュースグループ、ディスカッションフォーラムに代表されるオンラインコミュニティによる知識交換が活発である。これらのアプローチの利点は、質問者が特定のユーザに対して質問するわけではないため、回答者が自分の意志と都合で答えることができ、誰もが気軽に参加できることである。しかし、これらのアプローチによる知識交換の効果は高いとはいえない。

まず、誰でも答えられるため、回答内容の質と信頼性の低い回答が多くなり、知識を求める人にとってノイズ情報から有益な情報を見分けるのが困難となる。さらに、回答を得られる比率が低い。[LAKHANI2003]がApache Usenet help forumを分析した結果によれば、回答を得られた質問は全体の39%にとどまることがわかった。しかも得られた回答が問題解決に繋がったかを考慮していない結果である。本研究では、これらのオンラインコミュニティに見られる、質問を投げかける相手に関する不特定性という利点を、ソーシャルアウエネスコミュニケーションメカニズムで生かし、その上で、回答の質と回答率の向上を計っている。知識レベルの高い参加者とするためのエキスパート同定、および、参加動機を損ないにくくするためのエキスパート選定である。

タスクごとにエキスパートを同定する考え方は、[MOCKUS2002]にも用いられている。開発ログを解析することにより、ファイルに手を加えた開発者を抽出し、保守者が尋ねるべき人間についての情報を提供する。このような、誰がエキスパートであるかを完全に開示するアプローチは、エキスパートに直接的に質問を送った場合

表1 実験結果

No	質問者	メソッド名(org.apache.luceneは省略)	メーリングリストでの回答者のID番号	STep_INで選んだDynCでの参加者のID番号	比較
1	1293	document.Document.add	1156	126, 256, 65, 782, 1269	-
2	1671	index.IndexReader.deleteAll	579	568, 1102, 1162, 1978, 1086	-
3	1605	index.IndexReader.isLocked	65, 579	65, 1268, 1255, 1337, 918	部分一致
4	390	search.spans.SpanFirstQuery.getSpans	579, 953, 549, 1286	292, 1558, 953, 1787,304	部分一致
5	1858	search.Hits.length	1286	953, 292, 126, 1799, 1286	包含
6	1856	queryParser.QueryParser.parse	126	1034, 1293, 1329, 1307, 70	-
7	1871	queryParser.MultiFieldQueryParser.getRangeQuery	126,1291,1286 1754	1117,321,1070,1162,126	部分一致
8	1214	FieldSortedHitQueue.getMaxScore	1276, 1162	1823, 1276	部分一致
9	1757	search.Hits.score	499, 1286	126, 1286, 1034, 1184, 776	部分一致
10	1409	index.FilterIndexReader.isCurrent	1162, 1246, 1772	1246, 1949, 2161, 1845, 1622	部分一致
11	1837	index.TermFreqVector.getTermFrequencies	1286, 126, 65	1817, 720, 1286, 1162, 1795	部分一致
12	1606	search.BooleanQuery.setMinimumNumberShouldMatch	1162	2061, 1162, 1286, 1777	包含
13	1928	index.FilterIndexReader.getTermFreqVector	1772, 849	1198, 1144, 1356, 628, 553	-
14	1926	index.FilterIndexReader.delete	1162	1952, 1162, 1074, 1456, 1978	包含
15	1777	index.IndexWriter.setSimilarity	1286	1499, 789, 873, 1231, 292	-
16	1777	search.Weight.sumOfSquaredWeights	1286	126, 1286, 1192, 1149, 1230	部分一致
17	1792	search.HitCollector.collect	390, 1162, 953	531, 100, 1192, 1684, 1149	-
18	1778	analysis.WordlistLoader.getWordSet	1978	1978	包含
19	1884	document.Field.setOmitNorms	65, 1582, 1286	2055, 1286, 1963, 1162, 2108	部分一致
20	2009	store.IndexOutput.writeChars	1162	1162	包含

に半ば強制的に知識交換を要求することになると考えられ、我々とは立場が異なる。

参加者の動機を高めるためのメカニズムとしては、experts-exchange.comにおけるexpert pointがある。質問に答えたエキスパートはexpert pointを獲得し、それを用いて他のエキスパートに質問することができる。我々の提案したアプローチと類似している点もあるが、experts-exchange.comでは、ユーザ間の既存のソーシャルネットワークを利用していない点が異なる。

7 まとめ

本研究は、ソフトウェア開発者がクラスライブラリをオンデマンド的に習得することを支援することを目的とし、知識交換の場としてのダイナミックコミュニティを形成する環境を提供するものである。

本稿では、Javaクラスライブラリ部品を利用し習得する過程において生じる、個々のJava開発者の時々刻々と変化する知識交換の需要に応じて、当該部品に関する専門知識を有するエキスパートを同定し、その開発者と良好な社会的な関係を持つエキスパートを選定することによってダイナミックコミュニティを形成する仕組みについて述べた。また、持続的な知識交換を支援するものとして、ソーシャルウェアネスコミュニケーションメカ

ニズムの提案を行った。これらの仕組みを実装したSTep_INシステムについて解説した。

今後の課題としては、Java以外のソフトウェアライブラリ部品の習得のための環境を実装すること、及びシステムを実運用し、提案したメカニズムの有効性を検証することが挙げられる。

参考文献

- [BASSET1996] Basset, P.G. : Framing Reuse: Lessons from the Real World. Upper Saddle River, NJ: Prentice Hall,1996
- [BERLIN1993] Berlin, L.M. : Beyond Program Understanding: A Look at Programming Expertise in Industry, in Empirical Studies of Programmers: Fifth Workshop, Palo Alto, CA, pp.6-25, 1993
- [COLEMAN1988] Coleman, J.C. : Social Capital in the Creation of Human Capital. American Journal of Sociology. 94: pp.95-120, 1988
- [LAKHANI2003] Lakhani, K.R., and E. von Hippel : How Open Source Software Works: Free User to User Assistance. Research Policy. 32(6): pp.923-943, 2003
- [LINDEN1998] van der Linden, P. : Just Java 2, 4th ed.Upper Saddle River, NJ: Prentice Hall, 1998
- [MOCKUS2002] Mockus, A., and J. Herbsleb : Expertise Browser: A Quantitative Approach to Identifying Expertise., in Proc. of 2002 International Conference on Software Engineering, pp.503-512, 2002
- [MUMFORD1987] Mumford, E. : Socio-Technical System Design: Evolving Theory and Practice, in Computers and Democracy, P.G. Bjerknæs, P. Ehn, and M. Kyng (eds.) Aldershot, UK: Avebury, pp.59-76, 1987
- [NAKANOKOJI2006] 中小路久美代, 山田和明 : 持続的知識 共創コミュニティのための要件と支援手法, in DynC Symposium 2006 (東京, 2006)
- [NEAL1996] Neal, L. : Support for Software Design, Development and Reuse through an Example-Based Environment, in Structure-Based Editors and Environments, Szwillus, G., and L. Neal. (eds.), San Diego, CA: Academic Press, pp.185-192, 1996
- [YE2004] Ye, Y., Y. Yamamoto, and K. Kishida : Dynamic Community: A New Conceptual Framework for Supporting Knowledge Collaboration in Software Development, in Proceedings of 11th Asia-Pacific Software Engineering Conference, pp.472-481, 2004.
- [YE2005] Ye, Y., and G. Fischer : Reuse-Conductive Development Environments. Automated Software Engineering. 12(2): pp.199-235, 2005
- [YE2006] 葉雲文, 山本恭裕, 西中芳幸, 浅田充弘 : 再利用ライブラリの利用と習得を支援するダイナミックコミュニティ環境. ソフトウェアシンポジウム2006論文集, pp.73-82

「SEC journal」論文賞 審査委員会審査報告



「SEC journal」論文賞審査委員会委員長
東京工科大学 学長
相磯 秀夫

第2回SEC journal 論文発表会において、最優秀論文賞ならびに優秀論文賞を受賞された皆様に心からお祝い申し上げます。

今回の論文賞審査は、「SEC journal」に掲載された3件の論文と新たに応募された極めて質の高い9件の論文の合計12件の論文を対象に行いました。審査にあたっては、まず審査委員会の中に設けた、大阪大学大学院情報科学研究科教授井上克郎委員長ら8名による論文査読会が厳正な査読を行い、4件の論文を優秀論文候補として選出しました。その後、私が委員長を務める論文審査会がその査読を認め、その中から最優秀論文の選考を行い、本日その結果を発表いたしました。

今回の論文審査にあたって特に配慮した点は、昨年のIPA フォーラムの講演で鶴保証城SEC所長が述べられたSECの活動理念・活動方針です。その1つは、ソフトウェアの開発を可能な限り組織的かつ数量管理することです。2つ目は、ソフトウェア開発の実態を可視化することです。3つ目は、開発の現場に学び、知識・技術（スキル）・経験を集大成し、開発の知恵として活用することです。4つ目は、それらをソフトウェア開発のプロセス改善につなげ、製品の高信頼性を保証することです。このような理念を踏まえて、論文を審査しました。

今回の受賞論文は、昨年の受賞論文と少し違った性格もっていることに気がきます。それは、いずれの論文も大学の研究者が重要な役割を果たしていることです。また、SECを含めた産学官連携の顕著な傾向が見られることも特色で、ソフトウェアエンジニアリング特有の特徴と考えられます。問題の提起は、産業あるいは政府側から行われ、それに対して大学側が、問題の分析を行い、問題解決の手

法の提案とその結果の評価を担当しています。実際の問題解決にあたっては、三者が協調して行っていることが見受けられます。今回の受賞論文は、ソフトウェアエンジニアリングの問題は産学官がそれぞれの役割を分担しながら、問題解決にあたることを望ましいことを示唆しています。

4件の受賞論文はいずれも“プロセス改善”に関連した問題を対象にしていますが、信頼性の高いソフトウェアの開発のために、検討しなければならない大変重要な課題を捉えていることは評価できます。また、企業側がかかえるプロセス改善の悩みを通して、大学側により研究課題を提示しているともいえます。実際の問題解決には、経験的なアプローチをとっていますが、これもソフトウェアエンジニアリングの現状を示しています。通常、大学では問題の解決には、理論的なアプローチをとることが多いのですが、ソフトウェアエンジニアリングの場合は、まだまだ経験的なアプローチをとることを物語っています。特に、最優秀論文は、企業横断的に現場のデータを積極的に収集し、そのデータを上手に活用している点が優れています。ソフトウェアエンジニアリングの分野は、縦割りの組織ではなく、組織横断的な問題解決手法が必要なことを示したという意味で高く評価される論文です。

審査論文を通して一般的にいえることは、ソフトウェアエンジニアリングの分野は、まだまだ解決すべき問題が山積しており、それらの解決を積み重ねてはじめて、体系的なソフトウェア技術が確立するということです。そのような意味でも、受賞された皆さんはもとより、ソフトウェア技術に関係する研究者・技術者・専門家の一層の研鑽と問題解決への挑戦を期待しています。

論文講評とSECへの期待

優秀賞受賞論文について、評価できる点、不足している点を審査委員の皆様にご説明していただきました。また、同時に、SECに期待することをお伺いしました。

「SEC journal」編集部



大阪大学大学院情報科学研究科
コンピュータサイエンス専攻 教授

井上 克郎

今回の「SEC journal」論文賞の対象となった論文は全部で12件で、その中から、7件の「SEC journal」への掲載論文が選ばれ、さらに論文賞として4件が選ばれた。それぞれ産学連携の枠組みで研究を推進したもので、特色のある実践的な技術を発表している。選ばれた論文の著者を祝し、査読にあられた査読者へ感謝する。

さて、対象となった論文の数が、去年より減少したのは、「SEC journal」の発行が進み、ある程度、論文の対象となる領域や、そのレベルがはっきりしてきたからではないか、と思われる。余りに外れた領域や、レベル外の論文は見受けられなくなってきた。そういう意味では、「SEC journal」の1つのスタンダードが確立されつつある、ということになる。逆に、選ばれた4件の論文が、比較的似た領域を対象としており、限られた著者で占められているという印象があるのが少し気がかりである。もう少し幅広い論文やケーススタディがあってもよいだろう。

今回は、大学関係者の論文が多くを占めたが、企業の人々だけで書く論文がもっとあってもよいと思われる。一般に、そのような論文は、自分たちの行ったことのみを、一生懸命書くだけで、その仕事の位置付け（他でやっていることとの違い）や抽象化、一般化が弱い場合が多い。従って、読み手にうまく伝わらず、不採録や、大幅な書き直しを要求される場合が多い。

まず、論文を書く前に、同類の論文に目を通しておき、位置付けをはっきりさせる。そして、何が売りなのかを明確化し、読み手を中心に考えた論文構成を事前に十分練る。こうすることで、論文としての価値がずいぶん違ってくると思われる。今後、企業からの投稿が増加することを期待する。



日本アイ・ビー・エム株式会社 技術顧問 兼
東京大学大学院工学系研究科 特任教授

富永 章

SEC収集データを用いた工数見積りでは、不揃いとなりがちな複数企業の多様な数値に対し、工夫を凝らして近似式の評価をしている。苦心して纏められた大杉氏らの努力には頭が下がる。今後さらに、実践向けの具体的な見積り法が提示されることを期待する。

菊地氏は、PMO組織の活動において品質に影響を与える要素を分析し、仮説を検証した。PMOという見地では、プロジェクトマネジメントの視点での要素がもっと多く入ると、さらに有益なものになると感じる。

水野氏はレビューの質と成果物の品質の関係を検討し、レビュー速度に対する指摘密度の高さが、最終的な品質に強く関係する点を示した。この結果を、問題予防に具体的にどう生かすかがさらに示されるとよい。

葉氏・山本氏の論文は、ソフトウェア開発者のダイナミック・コミュニティによる知の共創として、これまでに他へも種々の発表がされた研究活動の成果である。実運用による検証が待たれるところである。

さて、組込みソフトウェアの規模は世界的に急拡大し、例えば、搭載ソースが既に1億行を超えた航空機すらある。大規模化に伴うソフトウェア・グリッチは、今後の脅威ともなる。企業の情報システムも、さほど速くはなくとも規模は増大しがちで、問題も生じ易くなっている。

そのような中、ソフトウェア技術の追究・進歩は、将来へ向け不可欠で、急を要するものである。実践に役立つ成果がとくに望まれ、学術の一領域に閉じてはならない。例えば、障害原因の背後には、原因を産むマネジメント上の要因があるのが普通であり、その除去は予防の鍵である。SECの活動が、引続き広い見地で行われ、一層実践向けの成果が出るよう、大いに期待している。



北陸先端科学技術大学院大学
情報科学研究科 教授

片山 卓也

「SEC journal」2年目の本年も優秀な論文4編が選ばれた。このうち3編は、プロジェクト工数見積もり、プロセス改善、コードレビューなどに関する、いわゆるプロジェクト管理に関するものであり、他の1編は、ソフトウェア開発環境に関するものである。いずれも企業と大学の共同研究から生まれた論文であり、また、ソフトウェア開発の現実問題を最新の学術的観点から考察したものである。実際の問題を真正面から扱うという「SEC journal」の基本姿勢に沿ったものである。昨年度は、企業からの単独論文が2編選出されたのに対して、本年はすべて大学との共同研究に関するものである。これは、ソフトウェア開発現場の多忙さのために、技術者が論文作成の時間的余裕が取りにくいということであろうか。あるいは、企業と学界の連携の場所としてのSECの認識が高まったことかもしれない。

SECには、産学の連携の場としての大きな役割が課せられている。ソフトウェア工学においては、産業界からの開発者や大学や研究機関の研究者が共通の課題について研究開発をすることが特に重要であり、その場所としてSECが機能することが大いに期待されている。この2年間の論文を見ると、企業の開発データをもとにして、大学が実証解析を行うものが主なものであり、連携の実績は大いに上がっていると考えられる。一方、学界からの仮説なり、理論なりを開発現場で確認するタイプの協調は、今後の課題であろう。形式手法などが高信頼ソフトウェアの開発手法として注目されているが、今後はこのような形での協調が進むことを期待したい。



トヨタ自動車株式会社
常務役員

重松 崇

今回、発表された論文は日本のソフトウェア開発の現場から発信されたきわめて産業界にとって有意義な実証論文である。机上で発想された学術論文と異なり、ソフトウェア開発現場を改善しようとする熱意が伝わってくる。それに加え、ソフトウェア工学の実証という観点でまとめられており、非常に興味深い提案となっている。

最優秀賞に選ばれた「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」に関する論文は、急激にソフト化が進む自動車、デジタル家電などでの効果的な人員配置やスケジュール立案に大きな効力を発揮すると期待している。

「コードレビューの密度と効率がコード品質に与える影響の分析」と「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」の2つの論文はいずれも、ソフトウェア開発におけるプロセス改善活動の指針となり、日本のソフトウェアの高い品質維持に大きく貢献すると期待を寄せている。

「JAVA開発者のオンデマンドラーニングを支援するソシオテクニカル環境」は学習者、エキスパートと知識伝達方法のあり方を解いた興味深い内容である。これは、JAVA学習者以外についても適用できる可能性が高い。

今回のいずれの論文も大学や研究機関が協力して研究を進めている産学共同の発表であり、産業界からの開発者や大学や研究機関の研究者が共通の課題について研究開発をする場所としてSECが機能しはじめていることを意味している。

今後もSECが中心になって、論理的な枠組みやデータの裏付けの技術を一般化し、現場参加型の知見の共有化を加速していくことで、広く日本のソフトウェア産業の発展に寄与していくことを期待している。



株式会社CSKホールディングス
取締役

有賀 貞一

今年は4編の論文が審査に付された。昨年に比較して、実務者により著述されたものが見られないのはさびしい。もちろん各論文のテーマは、実務の中から提起された問題意識、課題等に基づいて、大学からのメンバーを中心にまとめられているので、内容的には一定のレベルを維持している。査読の課程で実務者のものがなかったかどうかも確認したが、昨年に比べいずれも内容的に合格ラインに達せず、また論文の形式上も問題があったため最終審査に残らなかった、ということであった。ただし、前回も書いたが、SEC本来の目的に合致したものであれば、必ずしも純粋な学術論文形式を追求する必要はないと考える。

最終的に残った4編であるが、いくつかの共通的な印象があるのでまとめてみた。まず、テーマの切り取り方がうまい、といえる。限られた誌面で、目的を明確にし、説得力ある論旨を展開するには、絞込みが必要である。この点いずれもうまくまとまっている。ただ同時に、もう少し「大きさ」を感じさせるものが欲しいと思うのは、欲張りすぎか？これが一部の論文で、表題が内容を代表していないような印象をもたらしている原因だ。また、学側の論文を書きなれたメンバによるにもかかわらず、未定義語の使用や独自用語の使用等が見られ、やや残念である。

SECコンファレンスにおけるプレゼンテーションは概ね良好であったが、一部時間オーバーや、まとまりの悪さを見せたものがあり、運営上今後の工夫が必要である。来年度は、実務者が学側からの支援によって自らまとめ上げ、形式的にも整ったものが最終審査に残ることを期待したい。実務をこなしながらの論文書きはそれほど楽な仕事ではない。ネタがよければ、事前にうまくまとめるための支援策なども考えられるべきであろう。



パナソニック モバイルコミュニケーションズ株式会社
取締役社長

櫛木 好明

携帯電話は音声電話やメールといったコミュニケーションツールの時代を超えて、インターネット・ゲーム・音楽・お財布代わり等、日常生活に必須のツールになっている。これに伴って、電話帳の個人情報と、お金につながるID情報など情報セキュリティ機能が必須になっている。これらの機能拡大にとどまる気配はない。

これを実現する組み込みソフトウェアは、かつての銀行オンラインシステムを上回る規模と言われており、かつてない大規模開発のマネジメント能力が要求されている。したがって、解決のためには、ソフトウェアエンジニアリングで築きあげられてきた各種の技法やマネジメント手法を駆使し、よく統制されたプロジェクトチームが必要である。さらに、開発コストの爆発を解決するには、国内外の外部ソフトウェア会社とプロジェクトチームを組んで共同開発が必須である。

大規模開発での課題は、複雑性の中で、追加や修正が発生したときに混乱を早く収束させ、正しいアーキテクチャを構築することである。安易なジャンパー線のような解決や、当該モジュール間だけの調停は、スパゲティの最大原因になる。その解決のためには、システム全体の「見える化」が必須であり、調停のルールが必須である。いずれも「科学的に分析」されたソフトモジュールの性能や機能の分析が基盤に必要である。勘と経験の解決策はその後必ず「もつれ」の元になる。

「SECコンファレンス2006」で発表されているような業績が、試用・実用を経て幅広く課題解決に実用されてくることを期待する次第である。

進化し続ける機能の爆発を、常に監視し、適切に拡大に適応するにはどうすればよいのか。CMMIやISO9000によるマネジメントが重要であることは間違いない。しかし、サービスやアプリの無限繁殖を、過去に市場に投入した端末で受け止めなければならない。そのためには常にシステム全体の適正に判断し「アーキテクチャ」を正しく制定するチームが必須である。ただ現実にはスーパーマンがいて「えいや」と決めるのが事実という場合が多い。

天才に頼るのではなく、「工業」として誰でも実現できるようにすることこそ、業界に課せられたテーマであり、SECの解決策推進活動に大いに期待するものである。

(順不同、敬称略)

ソフトウェア・エンジニアリング 実証論文をよむ

株式会社東芝 ソフトウェア技術センター 企画担当 参事

平山 雅之

「SEC journal」論文賞の前提にあるソフトウェアエンジニアリング領域における産学連携による実践のあり方や今回、論文賞として選ばれた4編の論文に関する背景などについて簡単に紹介をしてみたい。

1 はじめに

昨年に引き続き「SEC journal」論文賞の4件が発表された。「SEC journal」論文賞の選定の基本的な考え方は、ソフトウェア工学や開発技術の実践経験を知識として整理し、それを多くの方々に参考にしていただくことに重点が置かれている。ソフトウェア開発の現場では、様々な技術革新や導入にも関わらず、相変わらず様々な課題や問題点が発生している。こうした課題や問題の解決へのヒントとして今回の論文4編をみると、どの論文を取り上げてみてもいずれ劣らず実際の開発現場での参考となる優れた論文が選ばれたように思う。本稿ではこの4件の論文が取り上げている主題に関して、その位置付けや意味などを紹介していきたい。

2 ソフトウェアエンジニアリングの実践

2.1 ソフトウェアエンジニアリングの立ち位置

ソフトウェアエンジニアリングはソフトウェアを作るための道具と考えることができる。通常、こうした「道具」を適切に使えば、より良いソフトウェアを作ることができる。ソフトウェアエンジニアリングで用いられる開発技術や手法の多くは、コンピュータサイエンスやソフトウェア工学の分野で研究され議論されてきた成果から生まれてきている。しかし、一方でこれらの「道具」は現場で利用され、ソフトウェア開発の助けとなってはじめて「道具」としての価値が生まれるものでもある。このためソフトウェアエンジニアリングは実際にその技術を利用するソフトウェア開発現場での実践の視点とソ

フトウェア工学を研究している大学などでの研究の視点が密に連携することでより洗練された技法へと進化していくと考えることができる。

現在、国際的に見た場合、ソフトウェアエンジニアリングの拠点として名が挙がる組織として、米国カーネギーメロン大学にあるSEI (Software Engineering Institute) やドイツランフォーファ財団による実験的ソフトウェア工学研究所 (IESE) などがある。CMU / SEIはCMMI (Capability Maturity Model Integration) などソフトウェアプロセスに関する成熟度に関する研究で世界をリードしており、IESEは製品開発のシリーズ化を推進するプロダクトラインなどの技術研究を進めている。この2つの組織に共通しているのは、大学と企業が協力し合っソフトウェア開発の現場での課題をくみ上げて、それに対する解決のための技術研究を推進している点にあり、企業と大学の距離が比較的近いという特徴がある。これに対し、我が国のソフトウェアエンジニアリングでは、企業などの実際のソフトウェア開発の現場と大学などの研究の場の間が比較的開いており、双方で課題共有や解決策検討などについて有効なスクラムが組まれていないことが問題といわれている。本来、ソフトウェアエンジニアリングがソフトウェア工学分野で培われた技術をベースに現場の課題解決のためのソリューションを提供していくものであると考えた場合、この企業と大学との距離の隔たりは、ソフトウェアエンジニアリングの進化において致命的となってしまう。

2.2 産学連携における双方の役割

このように考えていくと、ソフトウェアエンジニアリングを実践しその効果を実証していくための最適な枠組みは産学の連携を如何に活用していくかという点に帰着

する。その枠組みの中で産学がそれぞれにふさわしい役割りを担っていく必要がある。

(1) 企業側の役割り

企業が実際にソフトウェアを開発し提供していく中では、開発過程で様々な課題やその解決のための工夫などが蓄積されている。一般に、こうした課題やその解決のための工夫がソフトウェアエンジニアリングのベースになるが、その多くは個々の企業の現場に埋もれてしまっているケースが多い。このため大学など研究機関でこれらの経験を整理し手法として一般化するという形をとることは極めて有効なアプローチである。

(2) 大学など研究機関の役割り

一方、こうした企業の開発現場からのボトムアップ的な課題抽出や解決手法の一般化といった方向とは逆に、現状の数歩先を行く革新的な開発手法の基礎となる技術要素を検討し、それを実際の現場への適用という形に摺り合わせていく方向性も必要である。このようにソフトウェアエンジニアリングをより洗練されたものとするためには産学の連携の中で双方が経験や知恵を出し合っていくことが極めて重要になってくる。しかし、前節で指摘したとおり我が国のソフトウェアエンジニアリングに関しては、この産学の距離があまり密接でないため、現場の問題意識や工夫、あるいは大学での先端研究の情報などが有効に連携した動きができていない。SECはこうした背景のもとに、その間を繋げる役割りも担う組織として設立された経緯がある。

今回の「SEC journal」論文賞に選ばれた4件の著者を見ていただくとわかるように、いずれの論文も企業側の執筆者と大学側の執筆者が連名になっており、産学協同行われた研究やプロジェクトでの知見を整理したものと位置づけることができる。このような産学連携という視点で今回の4件を見た場合、

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

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
コードレビューの密度と効率がコード品質に与える影響の分析

Java開発者のオンデマンド・ラーニングを支援するソシオテクニカル環境

の中で、 ~ の3件はいずれも企業内で収集された様々なデータについて、適切な分析を大学サイドで行い、その結果について企業・大学の両方で精査していくといった形の産学連携の成果を整理したものであることが読み取れる。一方、については企業が抱える課題点の解決のために、大学サイドで研究されていた要素技術を応用適用した例を評価したものであることがわかる。このように産学連携という視点で、それぞれがどのような役割りを担っているかという点でこの4件を見ても参考になるのではないかと思う。

以下、この4件について、論文の背景となる技術などについて簡単に紹介してみたい。

3 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

ソフトウェア開発において重視される視点の1つが見積りである。ソフトウェアを開発する際に、どれくらいの作業が発生し、それにどれくらいのコストや期間・工数を要するかを正しく見積もることは、ビジネスとしてソフトウェア開発を考える上で極めて重要な視点である。ソフトウェアの見積りについては、古くから開発するソースコードの行数に基づくCOCOMO (Constructive Cost Model) やソフトウェアが提供する機能の規模に着目したFP法 (Function Point Method) など様々な方法が提案されてきている。その中でも、実際の開発現場でもっとも広く使われている方法が、過去の類似プロジェクトなどのデータを元にこれから手がけるであろう開発を見積もる類推法と呼ばれる方式である。この論文中で引用されている回帰分析を用いた方式もこの類推法の延長線上にあると考えることができる。しかし、こうした類推法を採用する場合には「適切な類似プロジェクト」を見つけ出すことが極めて重要かつ難しい問題となる。SECでは発足当初より国内の約1,000プロジェクト以上の開発データを収集し、見積りの際の参考情報として活用していただくことを目指してきた。本論文ではこのSECで収集したデータの中から、「類似プロジェクト」を効率的に見つけ出し、その情報を活用して新規プロジェクトの「類推見積り」を行う方式を検討したものである。

一般的にこうしたプロジェクトの開発データを集めた場合、往々にして一部のデータが収集されず欠損値となってしまうことが多く、プロジェクトが当該プロジェクトと類似しているかという判断が難しくなってしまう。SEC収集のデータについても、収集データの一部に欠損が含まれており、この論文ではこの問題を解決するために、「協調フィルタリング」という考え方に着目し、ソフトウェア開発における見積りに応用した点が極めてユニークであり、かつ、その適用結果を見る限りにおいて、極めて実用的な研究となっている。ソフトウェアの見積りに関して、より実践的な方式の検討を考えられている方々には是非、参考にしていただきたい論文の1つである。

4 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

「SEC journal」で論文を募集すると必ずといっていいほどプロセス評価や改善、あるいはCMMといったキーワードを含んだ論文がいくつか投稿される。これは我が国のソフトウェア開発の現場が如何に「プロセス」というキーワードに魅されているかを示している。さて、この論文評、結論から言うところの「プロセス評価・改善」に魅されている方々には是非、一読していただきたい論文の1つである。現在、ソフトウェア開発の現場の興味をひいているソフトウェアプロセス評価/改善論の中心は、CMMIに代表されるように、「ソフトウェア開発過程で実施すべき事項ができていないかどうかを確認し、できていなければその部分を是正していくことで全体として開発が円滑に進むようにする」という考え方である。しかし、こうした考え方一方で、「それではCMMIなどで求められるどの活動（プロセス、アクティビティ）をどのように、どの程度実施すると、結果として開発がどのようになるか（例えば製品の品質がどのように向上するか、あるいは、開発コストがどのようになるか）」といった点、すなわち実施した活動とその結果の因果関係を十分に説明しきれていないという疑問が常に付きまわっている。こうした疑問に対し、この論文では「フィールド品質」という明確なゴールを設定し、これについて実際のプロ

ジェクトで行われている活動がどのように影響しているかを客観的に評価し分析しようとしたものである。これはある意味で、実際のプロジェクトで実施された活動（プロセス、アクティビティ）とその結果（フィールドでの品質）の因果関係を分析した研究と読むこともできる。実際の開発現場にプロセス改善を根付かせるためには、こうした個々の活動とその結果や効果をきちんと分析評価して説得していくアプローチが基本となることは言うまでもない。その一方で、単に「プロセス評価・改善」というキーワードに踊らされて、その枠組みのみを表面的に実施している組織も少なくなく、こうした活動の推進者側と実際の開発現場の間で乖離が見られるケースもあるように聞いている。本論で導き出された「フィールド品質に関わる活動要因」そのものは、ある意味、この論文の筆者らが所属する企業という背景に起因するものも含まれるため、直接参考にするのは適切でないかも知れない。しかしこの論文で述べられている、ソフトウェアプロセスとその結果に関する因果関係の分析の考え方については、多くの方々に是非、目を通していただき、それぞれの組織で採用されているプロセス評価や改善の進め方の参考にしていただきたい。

5 コードレビューの密度と効率とコード品質に与える影響の分析

ソフトウェア開発を効率的なものとし、生まれてくるソフトウェアの品質を向上させる上では、「適切な作業を適切なやり方や道具立てで実施すること」は最初の出発点となる。このうち、「適切な作業」という部分については「ソフトウェアプロセス」という考え方が整理され、開発の過程で何をすべきか(what)は明確にしていこうという風潮になりつつある。この論文で取り上げている「コードレビュー」も、より良い品質のソフトウェアを開発することを考えると、実施が求められる必須のアクティビティ（作業）と考えて差し支えない。しかし、ここで次に問題となるのは、「それでは具体的にどのようにコードレビューをしたらよいか」といった疑問にすぐに行き着いてしまうことである。近年のソフトウェアは規模や複雑さが増大する一方で、開発に割く時間は短くなる傾向が強くなってきている。こうした中で、限られた時

間内でどのようにコードレビューを進めていくかは難しい問題である。即ち、単位時間当たりのレビュー量を増やせば目を通せる部分は多くなるが、その一方でレビューの質という点において雑なレビューになってしまう可能性がある。逆に、単位時間当たりのレビュー量を少なくし丁寧にレビューするとレビューの質は向上するものの、レビューに時間がかかったり、レビューできる部分が限られてしまうといった問題が生ずる。こうした中で、この論文は単位工数当たりのレビュー実施量（レビュー効率）とレビューにおける不具合指摘量（レビュー指摘密度）の関係を分析し、レビューの進め方やコード品質改善のための指針に結び付けている。コードレビューについては、どのような特性をもったコードを、どのようなメンバでレビューするか、あるいは、どのような観点でレビューするかといったことによっても、そのレビュー効率やレビュー指摘密度は大きく異なってくる。

コードレビューでは、これらの要因を考慮した上で、最適なレビュー方法やレビュー実施量、実施範囲を決めることでコードの品質を大幅に向上させることが可能となる。是非、読者の皆様も、本論で紹介されているようなレビュー効率やレビュー指摘密度といったデータを収集し、それぞれの組織やプロジェクトに適したレビュー方法につなげていただけるとよいのではないかと思う。

6 Java開発者のオンデマンド・ラーニングを支援するソシオテクニカル環境

近年のソフトウェア開発では、JavaやC++などの言語が用いられることが少なくなく、とくにこうした言語で用意されているクラスライブラリを使いこなすことが開発効率や品質といった面で極めて重要な要素になってきている。しかし、こうしたクラスライブラリを利用する場合に、利用しようとしているクラスライブラリがどのような機能を持っていて、それをどのように開発中のソフトウェアに組み込んでいくかといった情報、あるいは、それらの設計情報などが必ずしも十分に公開されたり整備されていないことが多い。結果として、クラスライブラリを利用する際には、既にそれらを利用した第三者を含めた技術者に聞くといったことが必要になる。この論

文ではこうしたクラスライブラリを利用する際の知識共有に着目し、上記のような問題を解決するための環境を提案したものである。具体的には開発者同士でダイナミックコミュニティを形成し、ライブラリ活用などに関する知識交換を持続的に実現する方式を採用している。本論文は「Javaのクラスライブラリに関する知識共有」の仕組みとしてみた場合、現実の開発現場での課題解決の一施策として大変に興味深いものとして読むことができる。また、一方で、近年様々な領域で問題視されている「知識共有」のための技術としてみても、知識を提供する側と知識を享受する側という2者間の関係やあり方、あるいは、コミュニティへの参加不参加の意思や情報開示の非対称性といった極めて興味深い話題も触れられており、参考にするところが多い。今回の論文では提案方式の考え方や実現方法とともに、それを実際にシミュレーションの形で検証しその有効性を評価しているが、その範囲では従来手法との比較においても十分な効果が期待できそうであり、実際のシステムとしての登場が待ち遠しい技術である。

7 おわりに

今回の「SEC journal」論文賞4件とその背景にあるソフトウェアエンジニアリング領域での産学連携によるソフトウェア工学の実証のあり方について簡単に紹介をした。筆者は今回の論文賞の選考に先立ち、その候補論文の査読を担当したが、今回の4件以外にも大変に興味深い論文がいくつかあったことを付記しておきたい。こうした論文を目にするにつけ、徐々にではあるが、我が国のソフトウェアエンジニアリングの分野でもようやく産学連携によるソフトウェア工学のよい実践例が紹介される文化が芽生えてきたように感じている。是非、こうした流れを途切れさせることなく、継続的に様々な実践例が公開されていくことを期待するとともに、その中から次の時代のソフトウェア開発を支える技術が生まれてくることも期待したい。

組織的プロジェクトマネジメント成熟度モデル(OPM3)の概要と適用上の留意点



平石 謙治†

組織的プロジェクトマネジメント成熟度モデル(OPM3: PMI標準)は、組織的プロジェクトマネジメントプロセスとベストプラクティスのモデル、さらにアセスメントのツールならびに成熟度の詳細や改善の順序を示すディレクトリより構成される。OPM3はアセスメントと改善を継続的に繰り返しながら逐次組織の成熟度を向上することを目指したモデルで、段階的な成熟度レベルの考え方は取り入れていない。OPM3の効果的な適用には、標準を補足するツールならびにアセスメントチームの構成、チーム内での共通の理解ならびに短いサイクルの繰り返しによる取り組みが重要である。

Organizational Project Management Maturity Model (OPM3) and its Application

Kenji Hiraishi †

OPM3(Organizational Project Management Maturity Model) consists of models for organizational project management processes and best practices, a self assessment tool and directories showing resulting maturity details and improvement paths. OPM3 is designed to apply cyclically and continuously with a short duration and intentionally avoided to define any staged maturity level. It is important to utilize additional tools along with OPM3 standard tools, to build an assessment team with common understanding and to apply with a short duration cycle.

1 はじめに

PMI標準の1つであるOPM3 (Organizational Project Management Maturity Model: 組織的プロジェクトマネジメント成熟度モデル) [PMI2003]について解説し、PMI東京支部組織成熟度研究会が実施したパイロットアセスメントについても述べる。この標準は、組織の中でそれぞれ個別に実施されるプロジェクトが組織のビジネス目標に沿った成果を上げ、組織が目指す方向とプロジェクトの成果の間にギャップが生じないようにすることを目的に開発された。

OPM3は、個別のプロジェクトのマネジメントに関する知識体系であるPMBOKガイド [PMI2000]、プロジェクトマネージャ個人の能力に関する標準であるプロジェクト

マネジャーコンピテンシ開発体系 (PMCDF: Project Manager Competency Development Framework) [PMI2002]と併せて、プロジェクトを実施する個々のプロジェクトマネージャのスキルと、個別のプロジェクトをマネジメントするプロセスに加えて組織までを対象とするPMI標準の3本柱を構成するものである。

本稿では、まず、OPM3の基盤をなす組織的プロジェクトマネジメントのプロセスとベストプラクティスについて解説し、次に組織的プロジェクトマネジメントの成熟度の概念について述べる。さらにパイロットアセスメントで実際にOPM3を適用したケースの実施例を紹介し、そこから得られたOPM3適用上の注意事項についても述べる。

† 日本電気航空宇宙システム株式会社, NEC Aerospace Systems, Ltd.
PMI東京支部組織成熟度研究会委員長

PMI, PMBOK, PMCDF, OPM3は米国Project Management Instituteの登録商標です

2 OPM3の概要

OPM3は、組織的プロジェクトマネジメントの成熟度測定の評価尺度を提供し、現状の測定結果を出発点として、さらなる成熟度向上に向けた改善計画立案へのヒントを示し、個別に実施されるプロジェクトの成功を組織のビジネス目標達成につなげることを目指して作成された [PMI2003]。

OPM3には成熟度を診断するためのツールならびに診断結果や成熟度改善計画立案の基盤となる詳細データを提供するツール類が含まれている [HIRAISHI2005] [HIRAISHI2006-1]。

図1にOPM3の全体像を示す。

2.1 組織的プロジェクトマネジメントプロセス

組織のビジネス戦略とプロジェクトの整合を確保するために、プロジェクト、プログラムならびにポートフォリオの3つの領域からなる組織的プロジェクトマネジメントのプロセスモデルを導入している。

(1) プロジェクトマネジメント

個別のプロジェクトのマネジメントであり、PMBOKガイドに示されるプロセスをそのまま採用している。

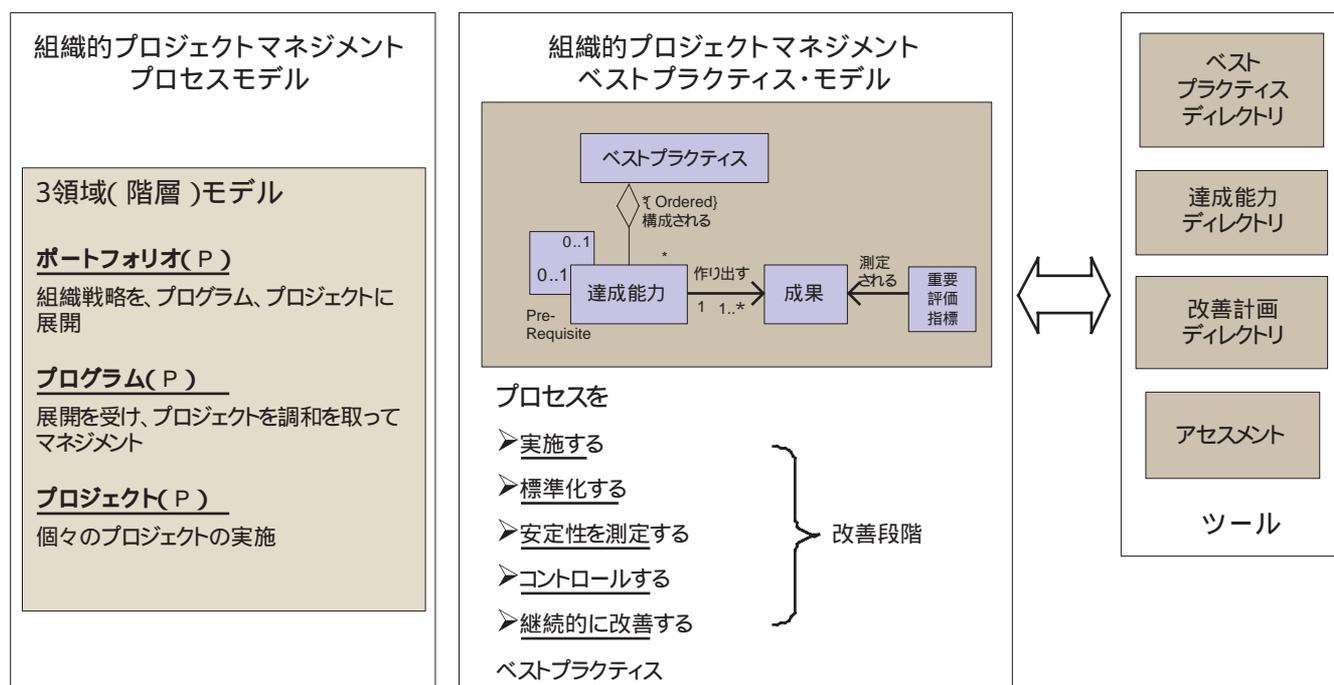
(2) プログラムマネジメント

“プロジェクトを個別にマネジメントするだけでは得られない利益とコントロールを可能にするために、調和が取れるようにマネジメントする一連の互いに関連があるプロジェクトのグループ”と定義され、このプログラムをマネジメントする階層がプログラム・マネジメントである。

(3) ポートフォリオマネジメント

戦略的ビジネス目標の達成のために、作業の効率的なマネジメントを目的にプロジェクトやプログラム等の作業をグループ化したものである。

ポートフォリオマネジメントは、プロジェクト、プログラム、他の関連作業等を識別し、優先順位を付けて、承認、管理、コントロールする。また組織の戦略計画や利用可能な資源を制約条件にしてプロジェクト及びプログラムへの投資を調整する。



成熟度はベストプラクティスの存在の有無で判断する。

図1 OPM3全体像

2.2 改善段階

OPM3は、次に示す4つの改善段階を踏んで組織が成熟することを前提にしてモデルを構築している。

プロセスの標準化

プロセスの安定的遂行状況の計測

計測結果に基づくプロセスのコントロール

継続的プロセスの改善

2.3 ベストプラクティス

組織的プロジェクトマネジメントプロセスが、成熟した組織でどのように実施されているかを調査、整理し、改善段階を踏んで成熟していくために実施される合計586個のベストプラクティスを定義している。

ベストプラクティスは以下の視点で整理された。

- ・適切なガバナンスの仕組みの開発
- ・プロセスの標準化と統合
- ・パフォーマンス重要評価指標の利用
- ・プロセスのコントロールと継続的改善
- ・プロジェクトマネジメントへのコミットメント

- ・プロジェクトの優先順位付けと組織戦略との整合
- ・プロジェクトの継続・打切りを判断する評価基準の適用
- ・要員のプロジェクトマネジメント・コンピテンシー向上
- ・プロジェクトへのリソースの割当
- ・チームワークの改善

ベストプラクティスは、さらに複数の達成能力に分解され、創り出すべき達成成果と、達成成果の有無を判断するための重要評価指標を示している(図2)。

総計2千数百個の達成能力が定義され、達成能力間の相互依存関係を改善計画の指針として示している。図2を例にとると達成能力B-3の実装は達成能力B-4とA-4両者の実装の上に成り立つ。

2.4 組織的プロジェクトマネジメント成熟度

組織的プロジェクトマネジメントの成熟度はベストプラクティスが実施されている度合をもって連続的スケールで判断する。他の成熟度モデルに見られるような成熟

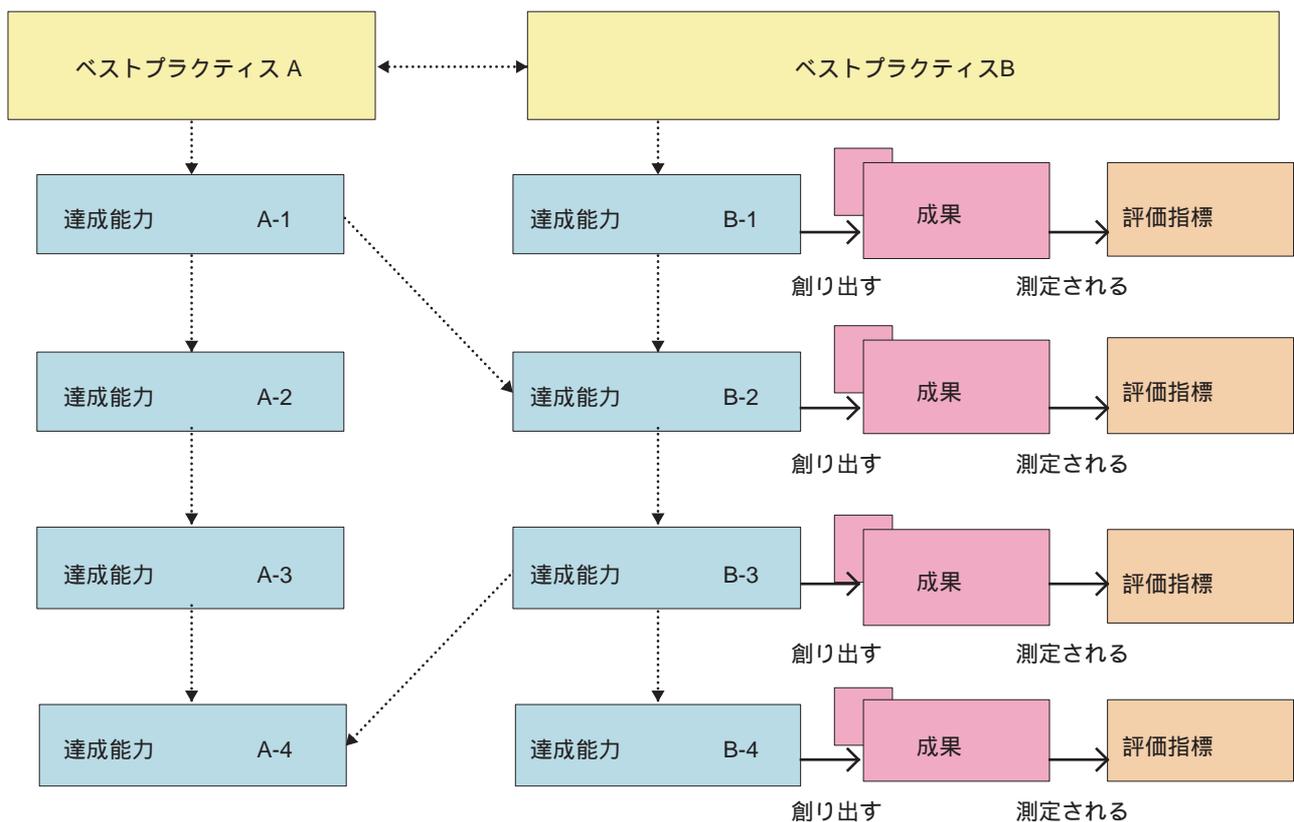


図2 ベストプラクティスの構成

度をいくつかのレベルに分けて段階的な数値を持って評価するものとしてはいない。個々の組織がそのビジネス目標に応じて注目する領域、段階に焦点を当てた使い方ができるように配慮している。

OPM3では関連するすべての達成能力の実装をもってベストプラクティスが実現されたと判断する。重要評価指標を用いることにより達成成果の有無、すなわち達成能力の有無を客観的に判断することができる。達成能力の有無によりベストプラクティス実施の有無、さらには成熟度を客観的に判断することが可能となる。

3 OPM3適用の事例

PMI東京支部組織成熟度研究会では、図3に示すステップによる複数のパイロットアセスメントを実施した。

パイロットアセスメントは、実際に適用することによりOPM3をよりよく、より深く理解し、さらにその効果や有効性、実際に適用する場合の注意点等を把握することを目的として実施した。また受診組織にとっては、自らの組織の組織的プロジェクトマネジメント成熟度をOPM3のモデルと比較して評価することにより測定し、今後の改善に向けての指針を得るといった効果が期待された。

適用に当たっては、最初にOPM3を理解し適用上必要な知識を身に付けることが求められる。次のステップでは、OPM3のモデルで記述された特性と組織の現在の特性を比較することによって組織的プロジェクトマネジメント成熟度の評価を行う。最初にアセスメントツールを用いて成熟度の概要を把握する概要把握アセスメントを実施し、ベストプラクティスのレベルで現状の組織成熟度を評価、認識する。一般にほとんどの組織において、この時点で識別される未実施ベストプラクティスはかなりの数となり、一度に取り組むことは現実的ではない。そこで詳細な分析を行う前に検討の対象とするベストプラクティスを組織の戦略を加味して絞り込む。

続いて、詳細アセスメントでは絞り込まれた未達成ベストプラクティスに対して達成能力のレベルで、より詳細に情報を収集し、重要評価指標により達成能力が創り出すべき達成成果の有無を判断し、達成能力が身に付い

ているかどうかを評価する。

次に現状の成熟状況を出発点として組織のビジネス戦略を加味して決定される改善目標に向けて改善計画を立案し、改善を実施する。その後、再びアセスメントを実施し、改善の成果を確認すると同時に次の改善サイクルに入る。

パイロットアセスメントの代表例では受診組織と研究会メンバーからなるアセスメントチームを構成し、半日あるいは1日の会議を、合計6回開催し改善計画の作成までのサイクルを実施した。最初の会議から報告書完成までに要した期間は3ヶ月、述べ工数は約30人・日であった[MURAYAMA]。

またOPM3を実際に適用するに当たっては、OPM3に備えられたツールに加えていくつかのツールを準備した[MATSUYOSHI2005-1] [MATSUYOSHI2005-2]。代表例で使用したツールを表1に示し、以降その手順について述べる。

3.1 環境の整備

PMI東京支部組織成熟度研究会では、パイロットアセスメントに着手する前に、準備のための環境整備が必要と判断し、(1) 解説書であるOPM3KF (OPM3 Knowledge Foundation) の日本語訳の作成、(2) メンタープログラムによる要員の育成を行った。またパイロットアセスメントとは別の活動としてOPM3を広く普及することを目的にセミナーの開催を計画しセミナー資料を作成したが、この作業自体が要員育成に大きな効果を上げることになり、またこの資料をパイロットアセスメント立ち上げ時の説明資料として活用することにもなった。

日本語版の作成においては、モデルの概念が必ずしもプロジェクトマネジメントの拡張だけではなく、経営レベルからの戦略との整合を強く意識しており、新しい用語も多く解釈に議論と時間を要する場面が多々あった。

メンタープログラムでは、5~6人のチームを計3チーム構成し、3ヶ月の間に1回当たり約2時間合計6回の座学を中心とした発表と討論を行いOPM3全体像の把握と理解に努めた。この時点では発行直後であるOPM3の専門家はいなかったため、メンター自身もOPM3の内容を把握しながらメンティーを指導する形態となった。

OPM3は、従来からのプロジェクトマネジメントの枠

組だけでは理解しにくい概念も多く（プログラム、ポートフォリオ）この部分をしっかり理解することに重点をおいて理解の促進を図った。

3.2 アセスメントの準備

OPM3の適用に当たってはアセスメントの実施に先立ち、(1) アセスメントチームの構築、(2) 受診組織の理解、(3) チーム内での共通の理解基盤の構築を行った。

OPM3を必要なレベルまで理解したPMI東京支部組織成熟度研究会のメンバ3名と受診組織の技術、企画を代表するメンバからなる混成アセスメントチームを構築し、必要に応じて適時、適切なメンバが参加することとした。また受診企業内に事務局を設置した。

最初に、キックオフも兼ねてチームメンバに対する半日間の説明会を開催した。この説明会では、前述のセミナー用資料を用いてOPM3の説明を行い、OPM3の理解の促進を図った。組織が組織的プロジェクトマネジメント成

熟度をOPM3のモデルと比較して評価するには、モデルの内容を理解し、アセスメントツールやベストプラクティス、達成能力等についての詳細データを示す3つのディレクトリの扱いに精通することが求められる。

同時に、受診組織の戦略目標を含む概要についての説明、さらには実施計画の確認を行った。なおこの席で、受診組織の事業内容、事業環境、戦略目標、組織構造、課題、さらにはOPM3に期待することを調査するための“事前調査表”を配布、記入を依頼しその後の知識共有化のベースとした。

3.3 概要把握アセスメント

このフェーズでは、OPM3で示される、どのベストプラクティスが実現されているかを評価し、その組織の全般的な達成度（成熟度）を識別した。

OPM3が提供するアセスメントツールは151項目の質問に答えることにより、どのベストプラクティスが実現さ

表1 パイロットアセスメントで準備したツール

ステップ	ツール、手法
環境整備	OPM3 KF 日本語版
	OPM3 解説教育資料
	メンタープログラムによる要員育成
アセスメント準備	アセスメントチーム
	事前調査表
アセスメント	解説付きアセスメント回答書
	エリア分析票
	成熟度分布図
	未実現ベストプラクティスリスト
	優先順位付け表
	優先順位マトリクス

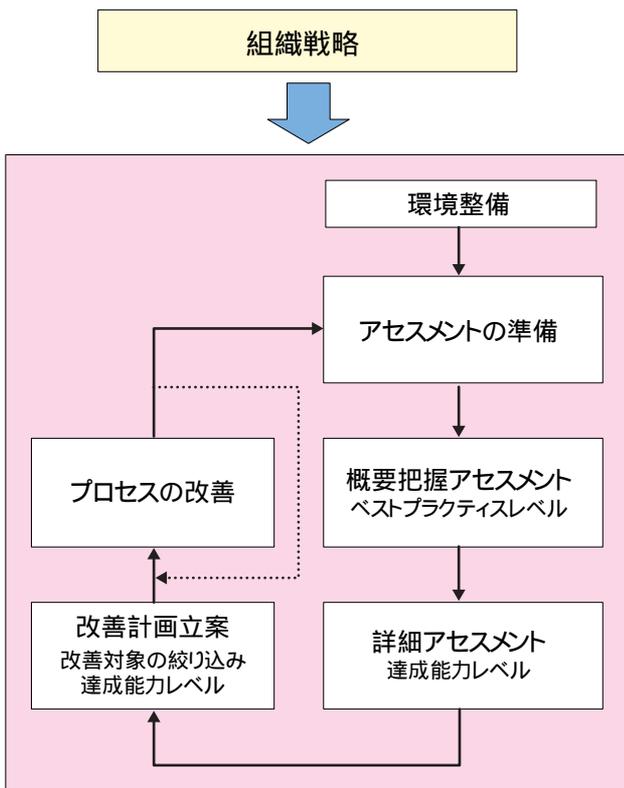


図3 OPM3適用のステップ

ポートフォリオ	90%	50%	0%	30%
プログラム	90%	44%	50%	50%
プロジェクト	90%	90%	100%	90%
	標準化	測定	コントロール	改善

図4 エリア分析表

れているかを識別し、成熟度の概略を素早く評価できるツールである。ただし、質問には抽象化レベルが高く理解が容易でない質問や、PMBOKの理解を前提とした質問も少なくなく、補足説明を加えてよりの確な回答を引き出す工夫をした“解説付き質問回答書”を作成して利用した。また組織を広くカバーしてデータを収集するため、複数の要員から回答を集めたが、要員の所属組織、現在までのバックグラウンド等により質問の理解に差異が生じて、要員間で回答にばらつきが生じることはやむをえないと判断した。この回答のばらつきに対応するためのルール（多数決）を設定した。これらの補足説明を活用し、ルールに従って評価を行い、結果を“エリア分析表”（図4）に表示し、組織の成熟度概要を把握した。

代表例の場合、全体的成熟度は約70%と識別され、220項目強の未達成ベストプラクティスが識別され、一時にすべてに対応することが非現実的であることがあらためて認識された。そのため、改善対象とするプロセス領域と改善段階を、戦略を加味してプログラム領域の測定段階の未達成ベストプラクティス5個に照準を絞ることとした。

さらに、プログラム領域の測定段階エリアに隣接するエリアの成熟度を評価し、関連の深い未実施ベストプラクティスを識別し、これらを加えて合計18個のベストプラクティスを詳細評価の対象とすることにした。

3.4 詳細アセスメント

詳細アセスメントでは、概要把握アセスメントで絞り込んだベストプラクティスについて、より詳細に情報を収集して評価を行った。対象となる各ベストプラクティスを構成するすべての達成能力について、重要評価指標を用いて達成能力が創り出すべき達成成果が存在するか否かを判断し、評価を行った。評価の結果、27個の未実装達成能力が確認された。

3.5 改善計画立案

詳細アセスメントの結果、組織がどの達成能力を身に着けているかが明確に識別された。一般にほとんどの組織において、この時点で識別される未実装達成能力の数も多く、これらを一挙に実装することは難しい。したがって、改善計画の立案に当たっては、OPM3が示す達成

能力間の依存関係、実装順序を考慮しつつ、以下に示す観点から評価を行い取り組みの優先順位を付けた。

達成難易度：早期の成果による改善活動の持続
戦略的優先度：戦略的ビジネスニーズとの整合
費用：他の要素の重要性とのバランス
効果：特に短期的な改善の効果

優先順位付けの結果に従って、改善に取り込む一連の達成能力を絞り込むこととした。

このため、検討対象となる達成能力ごとに上記4項目に対して優先順位を5段階評価で記入する“優先順位付け表”を作成し、この情報をもとにして改善の対象とする達成能力を絞りこんだ。この絞り込み作業においては、優先順位を可視化するため“優先順位付け表”に加えて、上記4つの観点のどれかをそれぞれ縦軸、横軸とする2次元平面上に対象となる達成能力をプロットした“優先順位マトリクス”も有効であった（図5）。

代表例の場合、戦略的重要度と効果の面から評価を加え、16個の達成能力を選んだ。これにより実現されるベストプラクティスは9個であった。

3.6 改善の実施

改善の実施そのものはOPM3の範囲外であり、今回のパイロットでも改善の実施は範囲外としたが、いったん計画が確立されると、リソースを配分してその計画を実行することが求められる。

3.7 プロセスの繰り返し

一定の改善活動が完了したら、アセスメントのフェーズに戻り、組織が現在置かれている組織的プロジェクトマネジメント成熟度の状況を再評価するか（このプロセスが望ましい）、または改善計画の立案に戻り、先のアセスメントで識別されたが今回の改善計画には取り込まなかったベストプラクティスへの取り組みを開始する。

以降このサイクルを繰り返し、継続的に改善を推進する。

4 パイロットアセスメントの教訓

OPM3は、プロジェクトマネジメントの範囲を拡張するだけでなく組織戦略とプロジェクトを結び付けるた

めのプロセスを含んだモデルであり、効果的な適用を行うには、ポートフォリオマネジメント、プログラムマネジメントの理解が必須であることがパイロットアセスメントを通じて再認識された。これによって、OPM3が単なるPMBOKの拡張ではなく、プロジェクトを実施する個々のプロジェクトマネジャのスキルと、個別のプロジェクトをマネジメントするプロセスに加えて組織までを対象するPMI標準の3本柱を構成するものであることも確認された。

このような状況下でOPM3の要員育成を図るには、OPM3を実際に用いることが必須であり、今回のパイロットアセスメントの場合も、文書のみでは習得が難しい知識や理解を実際のアセスメントを通じて習得すること

ができた。

適用ステップの観点では、入り口である“アセスメントの準備”、“概要把握アセスメント”がOPM3を効果的に適用する上での重要な要素であることも確かめられた。準備段階での適切な知識の習得が、効果的アセスメントチームの構築とチーム内での共通の理解実現に必須であり、OPM3を適用するには、最初にOPM3を理解し、適用上必要な知識を身に付けることの重要性があらためて認識された。

パイロットアセスメントを通じて、実効的なアセスメントチームの構築、アセスメントチーム内での共通の理解ならびに短いサイクルの繰り返しによる継続的改善が、OPM3を適用する上での重要な留意点であることが確認

された[HIRAISHI2006-2]。それぞれの留意点の内容を表2に示す。

アセスメントチームの構築では経営層、上級管理層のコミットメントと参加が必須であり、OPM3に精通したリーダーの下に受診組織の全階層レベルからの参加と受診組織の全体をつかんでいる要員による事務局設置、さらに外部のOPM3専門家の参加が重要である。

またOPM3はプロジェクトマネジメントの範囲を超えたモデルであるとはいえ、その基本はPMBOKにあるので、チームメンバーにはPMBOKの知識とプロジェクトマネジメントの経験が求められる。

OPM3に取り組んで効果を上げるには、組織のビジネス目標に基づいたOPM3導入の目的をまず理解した上で組織の形態・構造やOPM3プロセスモデルと実組織の対応、用語等についてアセスメントチーム内での共通の理解が必須である。

OPM3の特徴は、段階モデルではなく、連続的に成熟度の測定が可能であり、重点項目に絞って改善を行い、改善効果を評価しながら持続的改善を行うことができる点にある。

第3章で述べたパイロットアセスメントでのステップごとの対象ベストプラクティス/達成能力の絞込み結果を見ると、概略把握アセスメント時に観察された220個強の未達成ベ

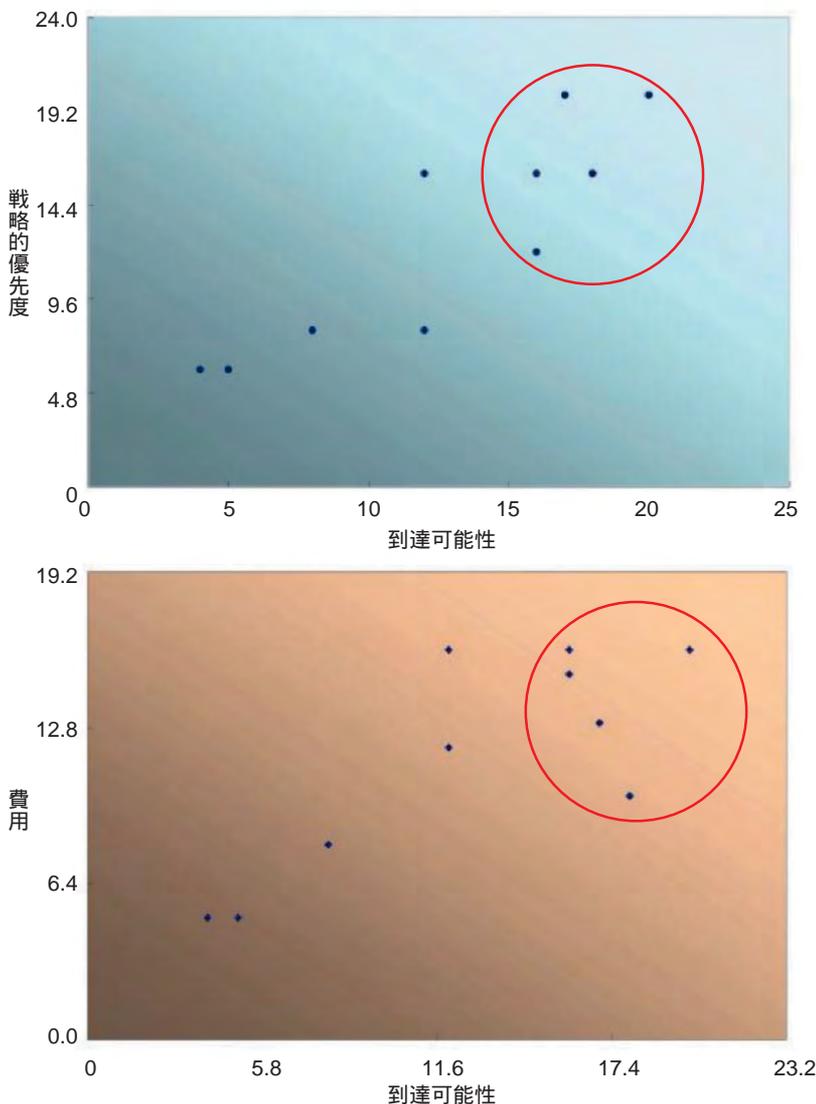


図5 優先順位マトリクス

表2 OPM3適用にあたっての留意点

アセスメント・チーム	<ul style="list-style-type: none"> ・経営層 / 上級管理層の参加 ・OPM3に精通したリーダー ・組織階層の全レベルからの参加 ・外部メンバの参加 ・適切なレベルのOPM3の知識 ・適切なレベルのプロジェクトマネジメント知識と経験
共通の理解	<ul style="list-style-type: none"> ・OPM3導入の目的 ・組織のビジネス目標 ・組織の形態 ・OPM3プロセスモデルと実組織との対応 ・達成能力等主要用語の定義
改善の繰り返し	<ul style="list-style-type: none"> ・OPM3 サイクルの繰り返しによる持続的改善 ・最重点への取り組み ・組織のビジネス目標と整合した段階的焦点の絞込み ・改善効果の評価

トプラクティスを改善計画では、9個のベストプラクティス、(達成能力に分解すると16個)にまで絞り込むことができた。これによって、改善の対象とする達成能力をここまで絞り込むことが可能で、短期間でサイクルを回して、あるいは短い周期(数ヶ月)と長い周期を組み合わせ合わせて成果を上げる目途がついた。

OPM3が推奨する適用ステップでは、アセスメントを概要把握と詳細評価の2つにフェーズに分け、改善計画立案を別のステップとしてとらえている。実際にパイロットアセスメントを行って見ると、詳細アセスメントで実施する達成能力レベルでの評価と改善計画立案ステップで実施する改善に向けての絞込みは一連の作業として実施すべき性格であることが判明した。今後の作業においては一連の作業としてとらえていくことを提案する。

5 まとめ

パイロットアセスメントを通じて、OPM3は、組織の戦略に即して柔軟に適用できる標準で、短期間でサイクルを回して効果を確認できる、組織的プロジェクトマネジメント成熟度の評価、改善をサポートする有効なツールであり、組織の強み、弱みの確実な認識や、戦略に沿った評価と改善計画立案に有効であることを確認した。

PMI東京支部組織成熟度研究会ではOPM3の日本での取り組みをさらに促進するため、日本語版の最終監修を行

っており、また汎用的に作成されたOPM3標準を業種ごとの言葉に置き換えるためのガイドライン作成も行いたいと考えている。

この5月のポートフォリオマネジメント標準[PMI2006-1]とプログラムマネジメント標準[PMI2006-2]のPMIによる相次ぐ発行に見られるようにOPM3をめぐる動きが活発になっている。これらの動きに注目しながら、今後、さらに適用実績を増やし、より使いやすい支援ツールの整備を進める所存である。

謝辞

本稿はPMI東京支部組織成熟度研究会により得られた成果を取り込んで作成した。ここに、組織成熟度研究会の各位、とくにパイロットアセスメントチームのメンバ、ならびにパイロットアセスメントに協力いただいた企業の方々に深謝する。

また本稿執筆の機会をいただいた日本電気株式会社の門田浩氏ならびに本稿をまとめる上での方向性についてコメント頂いた日本電気株式会社の井沢澄雄氏に深謝する。

参考文献

- [PMI2003] Project Management Institute, Organizational Project Management Maturity Model Knowledge Foundation, PMI, Dec. 2003
- [PMI2000] Project Management Institute, Guide to the Project Management Body of Knowledge (PMBOK Guide) 2000 Edition, PMI, Dec. 2000
- [PMI2002] Project Management Institute, Project Manager Competency Development Framework (PMCDF) , PMI, 2002
- [HIRAISHI2005] 平石謙治, OPM3とはどんなもの～OPM3(TM)の使い方とその特徴～, PMI東京フォーラム2005
- [HIRAISHI2006-1] 平石謙治, 組織的プロジェクトマネジメント成熟度モデル(OPM3)とその適用, プロジェクトマネジメント学会2006年度春季研究発表大会PP257-258, 2006年3月
- [MATSUYOSHI2005] Y. Matsuyoshi & K. Hiraishi, Study of Tools to Find the Focusing Points for Improvement Through OPM3(TM) Cycle, 2005 Proceedings of PMI Global Congress- Toronto, Canada, Sep. 2005
- [MATSUYOSHI2005] 松吉靖, OPM3(TM)の効果的活用～改善点を絞り込むためのツール, PMI東京フォーラム2005
- [MURAYAMA]2005 村山秀樹, IT企業におけるOPM3(TM)適用事例～日本アイビーエム・ソリューション・サービス株式会社での事例～, PMI東京フォーラム2005
- [HIRAISHI2006-2] K. Hiraishi, Points to be considered when applying OPM3-lessons learned from the pilot assessments, ProMAC 2006, Sydney Australia, Sep. 2006
- [PMI2006-1] Project Management Institute, The Standard for Portfolio Management, PMI, May 2006
- [PMI2006-2] Project Management Institute, The Standard for Program Management, PMI, May 2006

高度化プロセスにおける 定量的プロジェクト管理の実践と効果



岩切 博† 大曾根 一将†† 藤原 良一†† 中前 雅之††

定量的プロジェクト管理（QPM：Quantitative Project Management）は、プロジェクトの目標を確実に達成できるようにするための有効な方法である。QPMを実施することにより、事実が客観的に把握でき、問題の早期特定と適切な是正処置が実施でき、その結果、プロジェクトを成功へと導くことが可能となる。当社では、QPMのプロセスをCMMIのレベル4、5の高度化プロセスをベースに構築している。本稿では、高度化プロセスにおけるQPMを実践することにより、どのようにしてプロジェクトを成功に導き、事業目標を達成していったか、その事例と効果について紹介する。

Implementation and benefits of quantitative project management in our high-level processes

Hiroshi Iwakiri †, Kazumasa Oosone ††, Ryoichi Fujihara ††, and Masayuki Nakamae ††,

Quantitative project management (QPM) is one of the most effective methods to achieve project objectives surely. With QPM, we can understand facts objectively, identify problems early, take corrective actions appropriately, and as a result, we can lead our projects to their success. We have established a process of QPM based on the high-level process of CMMI Level 4 and 5. In this paper, we introduce an example and its benefits of having attained project objectives and business goals by implementing our high-level process of QPM.

1 はじめに

昨今の技術の高度化・専門化や要求事項の複雑化等により、プロジェクトのリスクは多様化し、プロジェクト・マネジャーの勘と経験だけでは予期せぬスケジュール遅れやコスト超過を防ぎきれず、これまで以上にプロジェクトを成功へ導くことが難しくなっている。

定量的プロジェクト管理（QPM：Quantitative Project Management）は、事実を客観的に把握することにより、問題の早期特定と適切な是正処置が実施できるようになり、プロジェクトを成功へと導くことが可能となる。つ

まり、QPMは、ロスコストを抑止するための有効な手法であり、プロジェクトの計画達成確度を向上させ、その結果、ビジネスゴール達成に寄与することができる [FURUYAMA2004]。また、プロジェクトが実践したプロセスの資産（教訓/リスク/QCDの実績データ/プロセス実施能力を示すプロセス実績ベースライン等）を新たに始めるプロジェクトのインプットにすることで、見積りや計画の精度向上が期待できる（図1）。

QPMのベストプラクティスとしてはCMMIがあり、当社ではCMMI[CMU]のレベル4、5の高度化プロセスをベースにQPMのプロセスを構築し改善を進めている。本稿では、高度化プロセスにおけるQPMを実践することにより、

† 三菱電機株式会社, Mitsubishi Electric Corporation

†† 三菱電機インフォメーションシステムズ株式会社, Mitsubishi Electric Information Systems Corporation

どのようにしてプロジェクトを成功に導き、事業目標を達成していったか、その事例と効果について紹介する。

2 公共事業分野における課題と改善方針

当社では公共事業分野をモデル部門として、SE部門とスタッフ部門を含め約100名の組織を対象に、高度化プロセスの導入を行った。公共事業分野では市場競争の変化が激しく、市場競争力をつけるために、売上向上と生産性向上の2つを改善方針として立てた。生産性向上については、

- ・上流工程での品質確保による手戻り工数削減と出荷後誤りの撲滅
- ・新技術の導入

の2つを具体的な改善方針とし、『上流工程での品質確保による手戻り工数削減』を行うための施策として、客観的かつ将来予測が可能なプロジェクト管理が行える、CMMIの高度化プロセスにおけるQPMを実践することとした。

3 QPMとは

QPMとは、プロジェクトの目標を達成するために、目標達成に効果的なプロセスを選択し、過去の実績データをベースに統計的手法を用いて行う管理のことである。QPMを実践するためには、以下の基盤が確立されている必要がある。

- ・標準プロセスが確立されている。
- ・標準プロセスに基づくプロジェクトプロセスが実行されている。
- ・プロジェクトプロセスを実行した実績データが蓄積されている。

QPMを実施するための組織的な活動（OPP）とQPMの計画と管理の概要を以下に示す（図2）。

(1) OPP：組織的な目標ガイドラインと予測モデルの確立 事業目標を達成するために注力すべきプロセスを選択

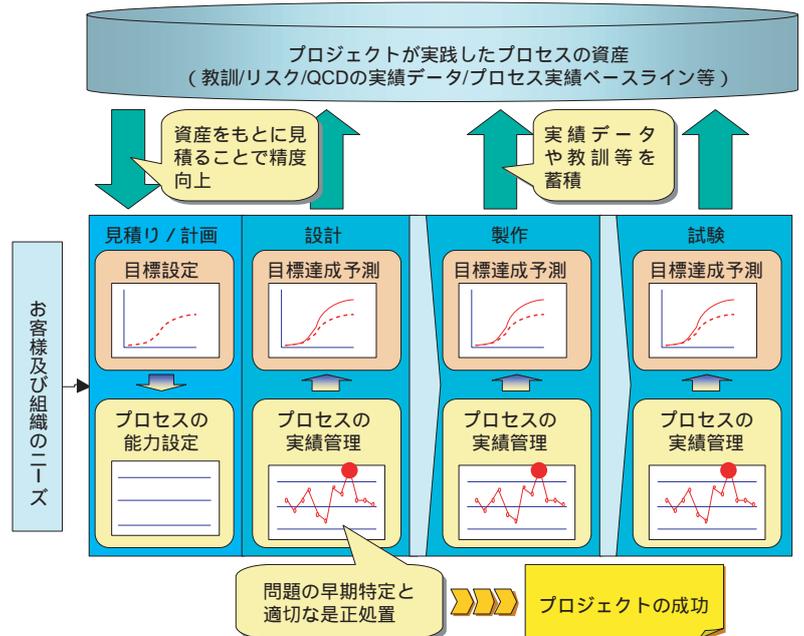


図1 QPMの効果

する。

選択したプロセスを計測するための指標と目標値を設定する。

過去の実績データをもとに、プロセス実績ベースライン（期待値と3σの幅を持った上下限值）を求める。プロジェクトの状況から最終目標の達成度を予測するためのモデルを構築する。

(2) QPM：プロジェクト計画

プロジェクトの目標を設定する。

目標を達成するためのプロジェクトプロセスを定義する。

目標達成に向け、重点的（統計的）に管理を行うプロセスを選択する。

指標と分析技法を選択する。

(3) QPM：統計的管理と目標達成予測

統計的管理において、プロセス実績ベースラインからの逸脱やバラツキの偏りを監視する。異常が検出された場合、原因分析と是正処置を行う（CAR）。

統計的管理プロセスにおいて、目標達成を監視する。統計的管理プロセスにおける実績データを、プロセス資産として収集・蓄積する。

プロジェクトの最終目標が達成できるかを予測し管理

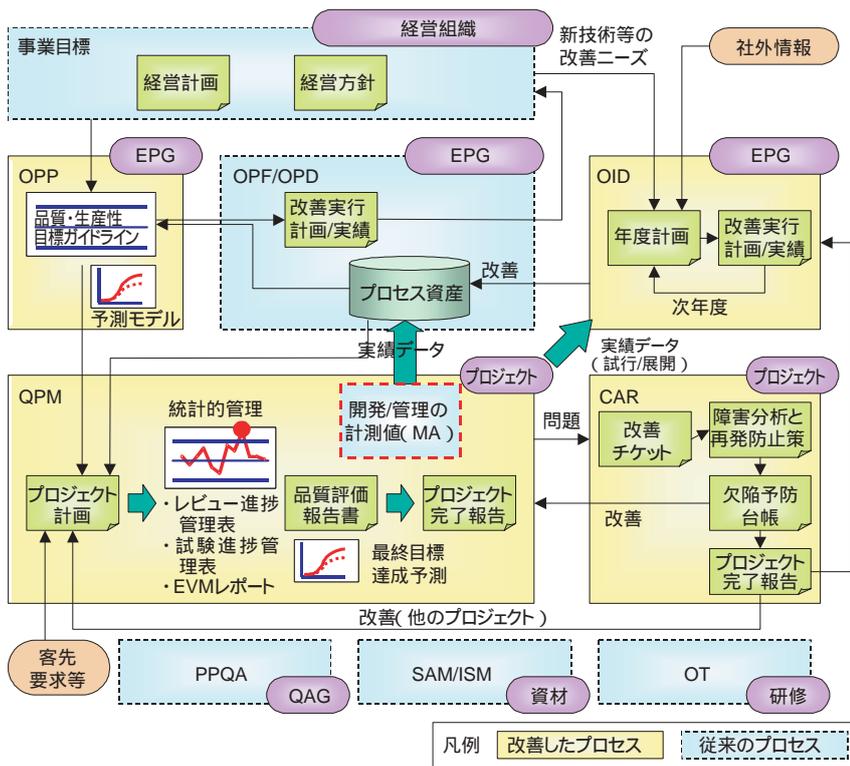


図2 QPMに関連するプロセス

表1 QPMの計測データと予測データ

	計測データ(中間目標)	予測データ(最終目標)	予測モデル
品質	レビュー指摘率(件/頁) 試験誤り検出率(件/KL)	出荷後誤り率(件/KL)	誤り実績モデル
生産性	SPI CPI	CVAC(完了時コスト差異) ライン生産効率(L/Hr)	EVM

する。

4 QPMに関するプロセス改善

上流工程での品質確保による手戻り工数削減と出荷後誤りの撲滅を目的に、QPMを実践するためのプロセスの改善について以下に示す[NAKAMAE2003]。

QPMのプロセスは、プロジェクト開始時点から定量的なモニタリングができるように、システムライフサイクル全般の品質(Quality)と生産性(Cost, Delivery)の定量的な計測を、品質管理のレビューや試験データ、進捗管理のEVMの進捗効率(SPI)やコスト効率(CPI)のデータを活用することとした。また、プロジェクトの誤り実績データのモデル化による出荷後誤り予測やEVMを活用した進捗・コスト予測により、実績最終目標達成予測を行うこととした(表1)。

QPMプロセスは、前述の計測データを統計的に管理し、その結果から限界値を逸脱したデータに対し、原因分析と是正処置を行うプロセスとしている。また、各フェーズからプロジェクトの最終目標の達成を予測可能とし、最終目標達成のためのプロジェクト計画の見直しを行うプロセスとした(図3)。

以下に、統計的管理、原因分析、目標達成予測のプロセス改善のポイントを示す。

4.1 統計的管理プロセスの改善

従来は、定量的な管理を下流の試験フェーズの監視に重点をおいて実施していたが、誤りは上流フェーズで検出する方が品質面からもコスト面からも効果的・効率的であるため、要件定義/設計フェ

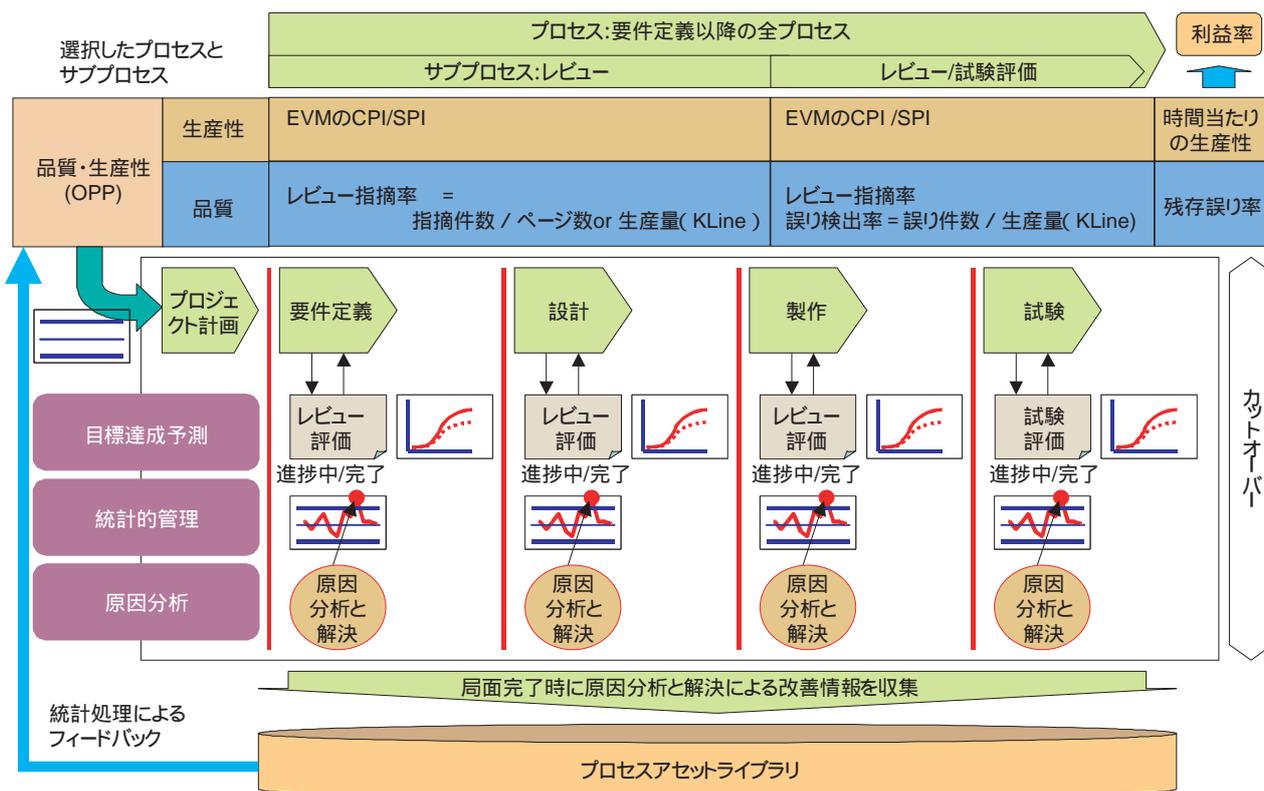


図3 QPMプロセス

ーズから試験フェーズの範囲において誤り検出を監視することとした。

統計的手法としては、プロジェクトでの管理負荷をなるべく増やさないようにするため、実績データをプロットするだけで管理できる運用の簡便なコントロールチャートを採用し、計測指標はレビューフェーズではレビュー指摘率、試験フェーズでは誤り検出率を監視することとした。

4.2 原因分析プロセスの改善

従来、プロジェクト実行中に実施していた原因分析の結果は、プロジェクト完了時点で過去を思い起こし記録していた。よって記録が十分ではなく、せっかくのノウハウが水平展開できず、再発させてしまうことが散見された。

今回はプロジェクト実行中に原因分析と改善提案を行い逐次記録に残し、迅速なプロジェクト内での再発防止の展開と、他プロジェクトへの確実な水平展開が行えるプロセスに見直した。

コントロールチャートでの監視により、プロセス実績

ベースラインより逸脱した場合、発生事象・原因と対処・根本原因・水平展開項目・教訓などを記載した”改善チケット”を発行し、関係者による原因分析と改善提案の検討を行うプロセスとした。

4.3 目標達成予測プロセスの改善

従来は試験フェーズにおいて、試験密度と誤り検出率から残存誤り率を予測する自社製ツールを用いて予測と品質評価を行っていた。

今回は、レビュー指摘率とレビュー密度から残存誤り率を予測できるように自社製ツールを改良することにより、要件定義/設計フェーズから予測と品質評価を行い、早期に問題の有無を把握し、品質を作り込むプロセスを改善した。

5 プロジェクトでのQPM実践事例

約4ヶ月50KLineから約13ヶ月1,600KLineまでの3つのプロジェクトにおいてQPMを実践した結果、有益又は改善

が必要であった幾つかの事例を以下に紹介する。

5.1 プロセス実績ベースラインの是正

(1) 現象

従来のプロセス指標を元に、フェーズ単位での管理限界を設定し、コントロールチャートによる監視を始めた。単体試験フェーズにおいて、72%のプログラムがプロセス実績ベースラインを逸脱してしまうという状況が発生した。しかし、それらのプログラムについて原因分析を行ってみても、それほど大きな問題は見つからなかった。

(2) 現象の背景

誤り検出率の管理は、フェーズごとの目標管理を実施しており、組織のガイドラインもプロジェクト完了時点での誤り検出数と生産規模の合計を基に目標値のガイドラインを作成していた。しかし、プログラム単位での生産量のばらつきが大きく、プログラムごとの誤り検出率の平均値と、フェーズ単位の合計値から算出した誤り検出率の値との差が大きくなり、管理限界値の設定方法に

表2 誤り検出目標算出例

プログラム	生産量 (KL)	誤り件数 (件)	誤り検出率 (件/KL)
A	10	3	0.300
B	5	1	0.200
C	100	15	0.150
合計	115	19	

プログラムごとの誤り検出率の平均	0.217
フェーズ単位(合計)をもとにした誤り検出率	0.165

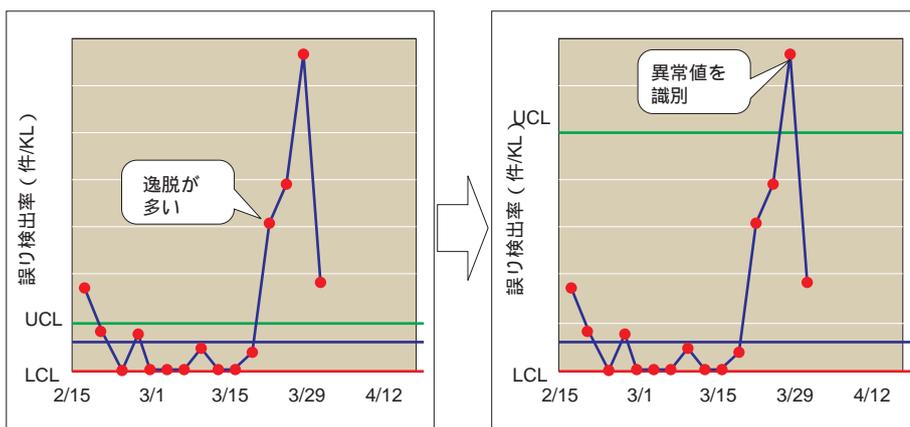


図4 プロセス実績ベースラインの是正

問題があることが判明した(表2)。

個々のプログラムの誤り検出状況を監視するためには、各プログラムの実績データのバラツキから求めたプロセス実績ベースラインを用いる必要があった。しかし、当初は、プロジェクト単位の実績データから管理限界値を求めてしまっていたため、結果的に管理限界値の幅が小さくなってしまい、問題のあるプログラムを適切に識別できなくなっていた。

(3) 改善策と効果

改善策として、プロジェクトで発生した個々のプログラムの実績データを収集し、管理限界値を求め直したところ、プログラムごとのバラツキが考慮されたパフォーマンス(実力値)となり、本当に問題のあった6%のプログラムのみを適切に識別できるように改善することができた(図4)。

5.2 レビュー不足を見つけて是正した事例

(1) 現象

ドキュメントのレビュー指摘率をコントロールチャートにて監視したところ、要件定義、外部設計では特に問題は発生していなかったが、内部設計のドキュメント6冊の内、1冊のドキュメントのレビュー指摘率が±3の管理限界を逸脱した。

(2) 現象の背景

要件定義や外部設計・内部設計では、レビューの対象となるサブシステムやアプリケーションごとに業務の理解度や技術的な難易度に差が出ることもある。コントロールチャートを用いてドキュメントごとのレビューの指摘率を過去のパフォーマンスの範囲と比較し監視することにより、これらを原因とするレビュー不足を発見することができた。

(3) 改善策と効果

お客様への追加ヒアリングの実施や、有識者レビューの追加等の

是正処置を早期に実施することができた。その結果、プロジェクト後半での仕様変更の多発等の問題を防止することができ、内部設計以降のコスト効率も改善した（図5）。

また、“改善チケット”を用いてプロジェクト実行中に原因分析を行ったことにより、問題を見つけたメンバー以外の人にもレビュー不足の原因と改善策について周知することができ、同様の問題の再発を防止することができた。

5.3 中間評価で改善施策を策定した事例

(1) 現象

プログラムごとの誤り検出率の実績をコントロールチ

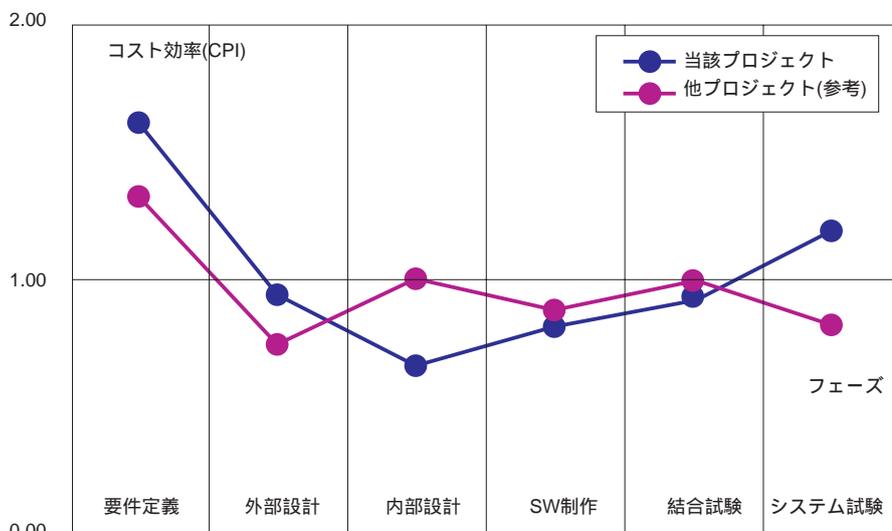


図5 コスト効率の推移

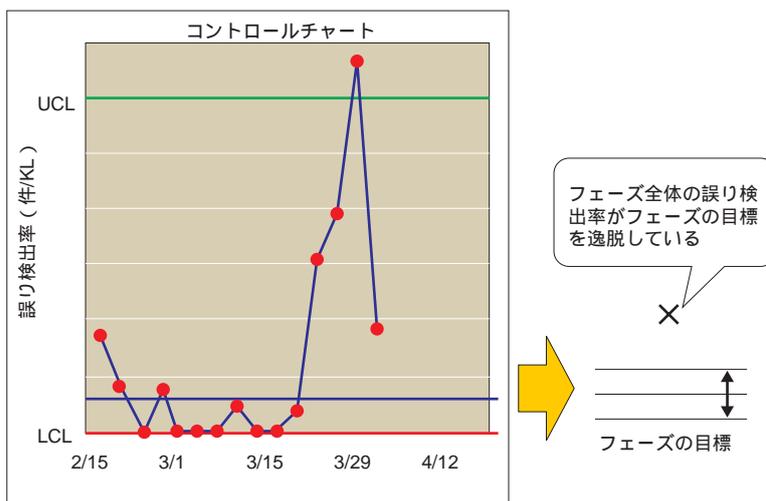


図6 中間評価での目標達成度評価

ャートで監視し、とくに問題は発生していなかった。しかし、前述のプログラムごとの誤り検出率とフェーズ単位の誤り検出率とのギャップの問題があり、単体試験フェーズの誤り検出率が±3の管理限界を逸脱する可能性があった（図6）。

(2) 現象の背景

プログラムの大きさが均一であればプログラムごとのコントロールチャートでの監視のみでよいが、実際にはプログラムの大きさは異なる。コントロールチャートによる監視において、個々のプログラムの実績値が、管理限界内に分布しても、フェーズ全体の誤り検出率は大きなプログラムに引っ張られてしまう。よって、プログラムごとの監視のみを行っていても、フェーズ全体の誤り検出率が目標を達成できるかどうかは判断できない。

(3) 改善策と効果

フェーズの目標達成度を評価するために、個々のプログラムの誤り検出率をコントロールチャートで監視するとともに、フェーズ全体の誤り検出率を求め、フェーズの目標の範囲内に収まっているかの評価を行った。

単体試験において、プログラムごとの誤り検出率は管理限界内に分布していたものの、フェーズ単位での誤り検出率は目標の上限値を大幅に超えていた。このままでは目標未達となる可能性が高いと判断し、フェーズの中間評価時に現状の課題抽出と原因分析を行った。その結果、単体試験で検出された誤りの原因の多くがコードレビューに起因するものであったため、後半に試験を行うプログラムについては、コードレビューをやり直すこととした。その時点では大きな手戻り作業であったが、その後の結合試験/システム試験の

品質が安定し、プロジェクト全体のCPIも1以上を達成することができた。

6 QPM実践による効果

前述の事例を一例に、QPMを実践したことにより上流工程での品質確保、出荷後の品質向上が達成できた。また、プロジェクトメンバーが改善効果を認識することにより改善意識の向上などの副次的効果も得ることができた。以下にQPM実践による効果を示す。

6.1 上流フェーズでの誤りの検出割合向上

QPMの実践により、設計フェーズでのレビュー不足部分の検出と是正、中間評価時点での改善施策適用等により、上流フェーズで検出できる誤りの割合が60% 84%と格段に向上した(図7)。誤りは上流フェーズで検出する方が品質面から見てコスト面から見て効果的・効率的であることは自明であり[YAMADA1993]、計画の遵守率も2倍近く向上した。

6.2 出荷後品質の向上

前述の要件定義から製作までの上流工程での品質確保により、システム品質も向上し、従来のシステム品質に比べ出荷後の不具合件数が40%削減、規模当たりの不具合率で17%の削減実績を得ることができた。また、QPM適用プロジェクトの品質が安定したことにより、結果的にお客様にも十分満足いただくことができた。

6.3 改善意識の向上

完了報告をまとめるプロジェクト・マネジャーが主に改善提案を行っており、ややもするとプロジェクト・チーム・メンバーとの間に改善意識の格差が見られた。

改善したプロセスにより、プロジェクト実行中に改善提案を実施させることで、プロジェクト・チーム・メンバーにも改善について検討する機会が増え、改善提案件数が増加した。また、改善効果が定量的に認識できるようになり、各自が積極的に改善に取り組む姿勢が見られ、改善意識の向上が見られた。

6.4 公共事業分野における事業目標達成

以上のような改善の相乗効果により、公共事業分野においては、複数の大規模プロジェクトを成功裏に完遂することができ、当初の事業目標も無事に達成することができた。また、このような活動をお客様に知っていただくことにより、当社の信頼度が向上するとともに、受注の機会が増加するといった副次的効果も得ることができた。

7 QPM実践における技術的な課題

QPMの実践を行うことにより、様々な効果が得られ、事業目標を達成することができた。しかし従来のプロジェクト管理の延長で、QPMのプロセスを実装したことで、様々な問題も発生した。以下にQPMを実践した経験をもとに今後の課題を示す。

(1) 効率的なプロセス管理指標への見直し

今回は、従来のプロセス管理指標との整合性を優先し、プログラムごとの管理指標とフェーズ単位での管理指標の二重管理となり管理指標を複雑にしてしまった。プロセスとフェーズの局面や、プロセスとプロジェクト完了の指標の定義を統一し、プロセスの管理指標を監視することで、結果的に局面やプロジェクト全体の品質の状況が管理できる管理指標に見直す必要がある。

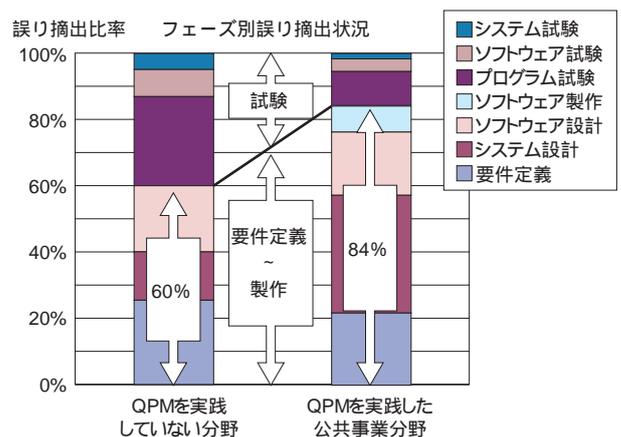


図7 上流フェーズでの誤りの検出割合向上

(2) 作業品質の定量的な把握

サンプル数の少ない、上流工程の品質評価は、誤り件数のみでは品質評価の判断が難しい。設計やレビューの作業品質のばらつきにより、その結果をもとにしたデータ分析・評価の精度は低くなる傾向がある。例えば、レビューは試験と違い、レビュー項目を設定して、その項目をレビューして、ある一定の作業品質を確保し、実施することはまれである。よって、レビューで管理基準をクリアし、発生したレビュー指摘内容の分析を行い問題が見つからなかったとしても、レビューという行為自体が適切であったかが常に問題視される。このように作業要員のスキルや業務知識、モチベーションなどのヒューマンファクターも定量化した、作業品質の定量的な評価手法を検討する余地がある。

(3) 実績データ収集の負荷軽減

QPM実践には正しく実績データを収集できていることが絶対条件であるが、この作業にはかなりの負荷がかかることがあり、小規模プロジェクトではプロジェクトに悪影響を及ぼす恐れもある。小規模プロジェクトでもQPMが行えるように、データの収集・分析作業の負荷軽減と的確な改善サイクルを回すエンピリカル手法[ALBERT2005]の考え方を導入し、ドキュメント作成やコーディング、レビュー/試験等の活動を行うことにより、自動的な実績データの収集と効率的な分析、改善施策の適用が行える生産環境の検討が必要である。

(4) 予測モデルの精度向上

早期に適切な判断を行うためには予測モデルの精度向上が求められる。しかし、上流工程ではドキュメント、下流工程ではプログラムと管理対象の性質が異なるものから最終品質を予測することは難しい。曖昧性が残るドキュメントと、論理性の高いプログラムとの品質に関連する相関を導き出し、上流工程でプロジェクト完了時の品質予測をどのように行うかが課題である。

(5) 統計的手法のトレーニング充実

QPMを活用するプロジェクト・チーム・メンバーの基礎能力として統計的手法のトレーニングも必要である。QPMを実践するには、プロセス実績ベースラインの適切

な活用、状況を正しく判断するための統計に関する高度なスキルが要求される。したがって、プロジェクト・チーム・メンバーが統計的手法を正しく理解し、実践できるようにするためのトレーニングコースの充実や、専門家の支援を受けられる仕組みの構築等も必要である。

8 おわりに

これらのQPMの導入・実践を通じて、公共事業分野では2003年度にSW-CMM レベル4を達成し、2006年にCMMI レベル5の組織を達成した[KANZAKI2005-1][KANZAKI2005-2][KANZAKI2004]。

前述の課題も含め、更なるデータの収集と、それらを組み合わせてのトータルなデータ分析が実践できる環境の実現に向けた検討や、予測を行うフェーズだけでなく、それ以前のフェーズの状況や、定性的な状況も加味して予測を行うことができる予測モデルの試行開発[KUWATA2006]等を進めている。

今後もさらなる改善を続けていくことで、確実にビジネスゴールを達成できる市場競争力の高い組織を実現してゆく所存である。

参考文献

- [FURUYAMA2004] 古山恒夫・富野壽監訳, 実践的ソフトウェア測定, 構造計画研究所, 2004
- [CMU] CMU SEI, Capability Maturity Model Integration (CMMISM) Version 1.1 Staged Representation
- [NAKAMAE2003] 中前雅之, 計測データに基づく見積り精度の向上, 2003年秋季PM学会予稿
- [YAMADA1993] 山田茂・高橋宗雄, ソフトウェアマネジメントモデル入門, 共立出版, 1993
- [ALBERT2005] Albert Endres, エンピリカルアプローチによる法則とその理論, コンピュータエージ社, 2005
- [KUWATA2006] 桑田すみれ, 実績データに基づく品質目標の達成度予測, 2006年春季PM学会予稿
- [KANZAKI2005-1] 神崎光司, 統計分析による類似プロジェクト抽出手法の提案, 2005年秋季PM学会予稿
- [KANZAKI2005-2] 神崎光司, 藤野友也, 平井規郎, 中前雅之, データ分布を考慮したパフォーマンスベースラインの近似算出の有効性, 2005年春季PM学会予稿
- [KANZAKI2004] 神崎光司, CMMI レベル4 実現のためのパフォーマンスベースラインの提供とその活用, 2004年春季PM学会予稿

BOOK REVIEW

れんそう 联想（上）（下）

凌志軍著、漆嶋稔訳

ISBN：（上）4-8222-4496-2 （下）4-8222-4502-0 日経BP社刊
四六判・（上）456頁（下）472頁・上下巻それぞれ定価2,310円（税込） 2006年2月刊



感動と哀切の中国産業報国叙事詩

原題は「联想風雲」、このほうがびったりだ。人民日報編集者の筆による人物を中心に描いた中国联想集団の起業とその前史から今日世界3位のパソコンメーカーになるまでの大河ドラマである。創業の1980年代からあの狂騒の90年代のIT業界、そしてIBMパソコン事業買収まで、中国現代史の実相と合わせて息もつかせずタイムトラベルさせてくれる。読んでいてほんとに涙の出る場面が何度も有る。感動の大河ドラマだからその内容をここで明らかににはできない。是非読んで、同時代の自己の仕事の記憶と重畳させながら、大河のあちこちで涙してほしい。中国の現代史、そして中国人への先入観が一変する。中国にはかくも純朴な人々がいたのかと思う。その

人々が猛烈な政策リスクと市場リスクの狭間で懸命に戦う姿は尊く感じられる。根底にある「産業報国」の理念は拝金主義の対極に輝いて見える。しかし読了後の哀切感は何だろう。彼らが結果として到達した資本主義社会の根源的な虚しさとグローバルリズムにつきまとう悲しさが浮き出てくる。あらためて上海で見た人民英雄記念塔横のひときわ高く耀いている「联想 / lenovo」の大ネオンサインが思い出され胸がつかまる。創業者の父親はここ

（神谷芳樹）

経験からの学習 プロフェッショナルへの成長プロセス

松尾 睦著

ISBN：4-495-37581-4 同文館出版刊
A5判・272頁・定価3,360円（税込） 2006年6月刊



オフショア開発、先人の知恵を有効活用できます

ソフトウェア開発には経験が必須である。知識だけでなく、場数を踏むことで得られる知見が、ソフトウェア開発において欠かせない。この書籍は、ソフトウェア開発のプロフェッショナルへと成長する過程において、経験という捉えどころがない要素をどのように考えるべきかについて、有効な指針となる書籍である。

経験と学習の関係、経験と業績の関係について、人材育成において欠かせない認知心理学における過去の論文や報告、さらには経営組織論やマーケティング論における定番の論点が数多く引用されており論理的な説明がなされている。

一人前になるには10年必要だとよくいわれる。石の上にも3年でないが、経験の長さは必須であるが、いい経験とそうでない経験もあるのも事実。それ以上に関係するのが、その

経験を活かすための学習する個人や組織の信念である。どのようなスタンス、モチベーション

で、その仕事にあたっているのかが非常に重要である。顧客志向と目標志向といった信念の違いによって、成長のスピードがどのように異なるかも検証されている。

業績という結果を導き出すためには、知識とスキルが欠かせない。この知識とスキル獲得のためには経験が必要であり、この経験を有効に働かすために個人や組織の信念が重要になる。これから若手の人材育成を考える人は、この構造をきちんと把握し、適切な指導やフォローをするべきである。しかし、経験においては回り道や失敗も重要であることも忘れてないことが必要であろう。

ソフトウェア・エンジニアリング関連イベントカレンダー

作成：SEC journal編集委員会

開催年月	開催日	イベント名	主催	開催場所	URL
2006年 11月	15日(水)～ 17日(金)	Embedded Technology 2006 / 組込み総合技術展	社団法人 組込みシステム技術 協会(JASA)	神奈川県横浜市・ パシフィコ横浜	http://www.jasa.or.jp/et/
	15日(水)～ 16日(木)	クリティカルソフトウェアワークショップ (WOCS2006)	宇宙航空研究開発機構(JAXA)	秋葉原コンベンションホール5B	http://www.wocs.info/
	16日(木)	連続セミナー2006 第5回 「経営戦略とIT戦略」	社団法人 情報処理学会	東京都千代田区・ 東京電機大学神田キャンパス 7号館1F丹羽ホール	http://www.ipsj.or.jp/
	28日(火)	SEC主催セミナー (テーマ:組込み開発の プロジェクトマネジメント)	IPA/SEC	東京都文京区・ 文京グリーンコート センターオフィス17階会議室	http://sec.ipa.go.jp/
12月	6日(水)	連続セミナー2006 第6回 「情報システム部門の役割と人材育成」	社団法人 情報処理学会	東京都千代田区・ 東京電機大学神田キャンパス 7号館1F丹羽ホール	http://www.ipsj.or.jp/
	22日(金)	SEC主催セミナー (テーマ:組込み開発の プロダクトライン)	IPA/SEC	東京都文京区・ 文京グリーンコート センターオフィス17階会議室	http://sec.ipa.go.jp/
2007年 1月	26日(金)	SEC主催セミナー (テーマ:組込み開発の 利用品質)	IPA/SEC	東京都文京区・ 文京グリーンコート センターオフィス17階会議室	http://sec.ipa.go.jp/
	30(火)～ 31(水)	ソフトウェアテストシンポジウム2007東京	NPO法人 ASTER (ソフトウェアテスト技術振興協会)、 JaSST'07 in Tokyo 実行委員会	目黒雅叙園	http://www.jasst.jp/
3月	6(火)～ 8(木)	第69回全国大会	社団法人 情報処理学会	早稲田大学 大久保キャンパス	http://www.ipsj.or.jp/
6月	6(水)～ 7(木)	Embedded Technology West 2006 / 組込み総合技術展 関西	社団法人 組込みシステム技術協会 (JASA)	マイドームおおさか	http://www.jasa.or.jp/etwest/

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

SEC主催セミナー報告

SECでは、2006年5月より、2005年度の活動成果と2006年度の取り組みテーマについてのセミナーを主催いたしました。参加者は、SEC-Webサイトでの告知と「SECからのお知らせ」と題したダイレクトメールで募集しましたが、毎回反響が大きく、参加者の皆様からご好評をいただいております。以下に、これまで開催した各セミナーの概要をご報告いたします。

なお、各セミナーの資料のうち講師の許諾を得たものについては、SEC-Webサイトの利用登録者専用ページにて公開しています(ログインが必要です)。

SEC主催セミナーは参加無料です。今後も様々なテーマのセミナーを企画いたしますので、是非ご参加ください。

【今後予定しているテーマ】

2006年11月...組込み開発のプロジェクトマネジメント

2006年12月...組込み開発のプロダクトライン

2007年 1月...組込み開発の利用品質

なお、現在、SEC主催セミナーは、主にSECの所在するビル(東京都文京区・文京グリーンコートセンターオフィス)で開催しております。他地域での講演をご希望の方は、SECまでご要望をお寄せください。

また、SEC-Webサイトにて「利用者登録」をしていただいた方に、SEC主催セミナーの情報をご案内させていただいております。ご希望の方は、登録項目「SECからのお知らせ」を受け取る」の欄にチェックをしてください。

詳しくは、SEC-Webサイト(<http://sec.ipa.go.jp/>)をご覧ください。

プロダクトライン開発入門

開催日時：2006年5月26日(金) 13:30～18:30

講師：北陸先端科学技術大学院大学 情報科学研究科 教授 岸知二

参加者数：82名

実装品質向上技術 - コーディング作法の使い方

開催日時：2006年6月23日(金) 13:30～18:30

講師：SEC組込み系プロジェクト 研究員 大野克巳

参加者数：57名

組込みスキル標準(ETSS)の活用【入門編】

開催日時：2006年7月24日(月) 13:30～18:00

講師：SEC組込み系プロジェクト 研究員 関口正

参加者数：55名

定量データの活用と共同研究成果発表

開催日時：2006年10月17日(火) 14:00～18:00

講師：SEC エンタプライズ系プロジェクト 研究員 横山 健次 / 大阪大学大学院情報科学研究科 助手 水野 修 / 奈良先端科学技術大学院大学 助教授 門田 暁人

開催場所：東京都千代田区・三菱総合研究所

参加者数：87名

モデル検査技術入門

開催日時：2006年10月30日(月) 13:30～18:00

講師：北陸先端科学技術大学院大学 助教授 青木 利晃

参加者数：43名

『組込みソフトウェア向け 開発プロセスガイド』解説

開催日時：2006年10月31日(火) 13:30～18:00

講師：SEC組込み系プロジェクト 研究員 山崎 太郎 / SEC組込み系プロジェクト 研究員 室 修治

参加者数：53名

編集後記

2005年1月に『SEC journal』を創刊してから第8号を数え、季刊誌として2年が経過しようとしております。

初心に戻って創刊号を開いてみると、“創刊にあたって”にある、『「高品質のソフトウェアを効率よく開発する手法を確立し、普及させる」ことを旗印に活動を開始します。ソフトウェア・エンジニアリングに的を絞り、学術論文のみならず、プラクティカルでエンピリカルな情報を幅広く発信したいと考えております。』という鶴保所長の言葉が目にとまりました。

『SEC journal』の情報発信内容を振り返ると、産学官連携の助けになることを目指した「組織紹介」、要求度合いが高い技術の「技術解説記事」、「招待論文」、開発現場での有効利用を期待したい「投稿論文」、「SECの成果報告」などといった内容ですが、情報発信が十分だと自信を持って話せる状況にはまだまだです。

SEC-Webサイトでは、利用者登録機能を実装してから既に7,000名を超える登録をして頂き、SECの発信する情報をご活用頂いております。『SEC journal』のPDF版も、人気の号では900件もダウンロードして頂いておりますが、他のダウンロード数(コーディング作法V0.8(20,000件超) \ V1.0(10,000件超) \ ETSS(20,000件超)等)があります。)と比べると2桁も違います。

『SEC journal』をもっとソフトウェア・エンジニアリングに関わっている方々に読んでいただけるようにするために、読者の皆様のご協力(企業や大学での紹介)をお願いしたいと思っております。是非、掲載された論文、技術解説記事などを利用して頂き、また、論文の投稿にも挑戦してください。

近い将来、より多くの論文等の投稿がされ、真に日本のソフトウェア産業の発展の礎となるよう、今以上によりよい誌面作りを心がけていきます。(ヒゲ)

本journalに対してのご意見は、<http://sec.ipa.go.jp/index.php>内の「SECへのお問い合わせ」をご利用ください(ご意見用メールアドレス <sec-journal@ipa.go.jp> でのご意見受付は終了しました)。

SEC journal 編集委員会

編集委員長

猪狩 秀夫

編集委員(50音順)

青木 奈央

赤田 眞弓

奥 保正

菊地奈穂美

新谷 勝利

田丸喜一郎

樋口 登

神谷 芳樹

門田 浩

渡辺 登



SEC journal® 第2巻第4号(通巻8号) 2006年11月30日発行

© 独立行政法人 情報処理推進機構 2006

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

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

本誌は、「著作権法」によって、著作権等の権利が保護されている著作物です。
本誌に掲載されている会社名・製品名は、一般に各社の商標または登録商標です。

お知らせ

SEC journal 論文募集

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

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

応募要項

スケジュール

・10号(2007年4月発行予定)

応募締切 2007年2月19日

投稿締切 2007年2月26日

採録通知 2007年3月13日

・11号(2007年7月発行予定)

応募締切 2007年5月21日

投稿締切 2007年5月28日

採否通知 2007年6月12日

採録決定後、1週間程度のカメラレディ期間があります。

詳細は別途通知されます。

採録の場合には「SEC journal」への掲載およびIPA SECのWebやイベント等での発表を行います。

提出先

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

eメール:sec-ronbun@ipa.go.jp

その他

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

論文賞

「SEC journal」では、毎年「SEC journal」論文賞を発表しております(前回は2006年10月24日SECコンファレンス)。受賞対象は、「SEC journal」掲載論文他投稿をいただいた論文です(論文賞は最優秀賞、優秀賞、SEC所長賞からなり、それぞれ副賞賞金100万円、50万円、20万円)。

応募様式

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

詳しくは

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



SEC Journal No.8
第2巻第4号(通巻8号)
2006年11月30日発行 ©独立行政法人 情報処理推進機構

編集兼発行人

〒113-6591 東京都文京区本駒込2-28-8 文京グリーンコート センターオフィス16階
独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター
所長 鶴保 征城

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

